parallel tools platform
http://eclipse.org/ptp

A New and Improved

Eclipse Parallel Tools Platform

Advancing the Development of Scientific Applications

Jay Alameda, NCSA
alameda@illinois.edu

Jeff Overbey, NCSA
overbey2@illinois.edu

UCAR University Corporation for Atmospheric Research Software Engineering Assembly
February 24, 2012

Portions of this material are supported by or based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under its Agreement No. HR0011-07-9-0002, the United States Department of Energy under Contract No. DE-FG02-06ER25752 and the SI2-SSI Productive and Accessible Development Workbench for HPC Applications, which is supported by the National Science Foundation under award number OCI 1047956

Based on slides by
Greg Watson, Beth Tibbitts, and others

Tutorial Outline

Time (Tentative)	Module	Topics
8:30-9:00	1. Eclipse & PTP Installation	✦ Installation of Eclipse and PTP (can start early as people arrive)
9:00-9:30	2. Introduction & Overview	✦ Eclipse architecture & organization overview
9:30-10:30	3. Developing with Eclipse	✦ Eclipse basics; Creating a new project from CVS; Local, remote, and synchronized projects ✦ Editing C files; MPI Features; Building w/ Makefile
10:30-10:45	BREAK	
10:45-11:45	3. Developing with Eclipse (continued)	Continue from before the break… ✦ Resource Managers and launching a parallel app ✦ Fortran, Refactoring, other Advanced Features
11:45-12:00	4. Wrap-up	✦ NCSA HPC Workbench, Other Tools, website, mailing lists, future features

Module 1: Installation

- ✦ Objective
  - ✦ To learn how to install Eclipse and PTP
- ✦ Contents
  - ✦ System Prerequisites
  - ✦ Eclipse Download and Installation of “Eclipse IDE for Parallel Application Developers” – parallel package
  - ✦ Installation Confirmation
  - ✦ Updating the PTP within your Eclipse to the latest release

About the Tutorial Installation

- ✦ This tutorial assumes you have Eclipse and PTP pre-installed on your laptop
- ✦ If you already have Eclipse installed, go directly to “Starting Eclipse”, slide 5
- ✦ If you don’t have Eclipse installed, you will need to follow the handouts so that you can catch up with the rest of the class
- ✦ Note: up-to-date info on installing PTP and its pre-reqs is available from the release notes:
  - ✦ http://wiki.eclipse.org/PTP/release_notes/5.0
  - ✦ This information may supersede these slides

System Prerequisites

- ✦ Local system (running Eclipse)
  - ✦ Linux (just about any version)
  - ✦ Mac OS X (10.5/Leopard or later)
  - ✦ Windows (XP or later)
- ✦ Java: Eclipse requires Sun or IBM Java
  - ✦ Only need Java runtime environment (JRE)
  - ✦ Java 1.6 or higher
    - ✦ Java 1.6 is the same as Java SE 6.0
  - ✦ The GNU Java Compiler (GCJ), which comes standard on Linux, will not work!
  - ✦ OpenJDK, distributed with some Linux distributions, has not been tested by us but should work.
  - ✦ See http://wiki.eclipse.org/PTP/installjava

Eclipse Packages

- ✦ The current version of Eclipse (3.7) is also known as Indigo
- ✦ Eclipse is available in a number of different packages for different kinds of development
  - ✦ http://eclipse.org/downloads
- ✦ With Indigo, there is a new package directly relevant for HPC:
  - ✦ Eclipse IDE for Parallel Application Developers
  - ✦ This is recommended for all new installs

Eclipse IDE for Parallel Application Developers (includes Incubating components), 184 MB
Downloaded 39,008 Times Details

" Package"

- ✦ Can also add PTP to an existing Eclipse installation

Eclipse Installation

- ✦ Download the Eclipse IDE for Parallel Application Developers package
  - ✦ http://download.eclipse.org
- ✦ Make sure you match the architecture with that of your laptop
- ✦ If your machine is Linux or Mac OS X, untar the file
  - ✦ On Mac OS X you can just double-click in the Finder
- ✦ If your machine is Windows, unzip the file
- ✦ This creates an folder containing the executable as well as other support files and folders

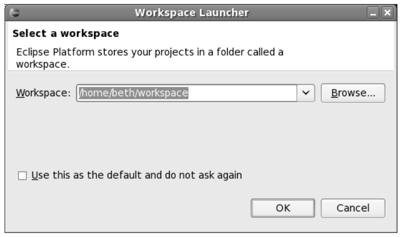
Starting Eclipse

- ✦ **Linux**
  - ✦ From a terminal window, enter "<eclipse_installation_path>/eclipse/eclipse &"
- ✦ **Mac OS X**
  - ✦ From finder, open the folder where you installed
  - ✦ Double-click on the application
  - ✦ Or launch from a terminal window instead (like Linux)
- ✦ **Windows**
  - ✦ Open the folder
  - ✦ Double-click on the executable

Specifying A Workspace

- Eclipse prompts for a workspace location at startup time
- The workspace contains all user-defined data
  - Projects and resources such as folders and files
  - The default workspace location is fine for this tutorial

The prompt can be turned off



Workspace Launcher

Select a workspace

Eclipse Platform stores your projects in a folder called a workspace.

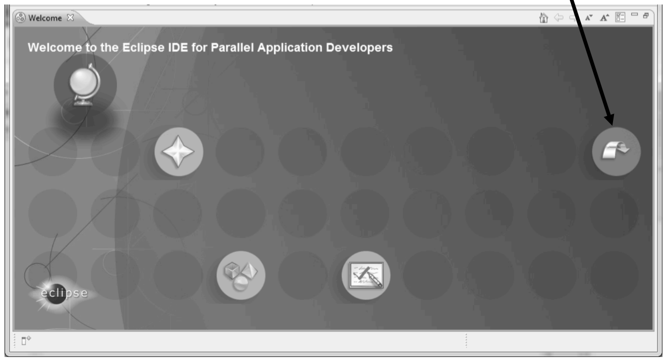
Workspace: /home/beth/workspace

Use this as the default and do not ask again

Eclipse Welcome Page

- Displayed when Eclipse is run for the first time

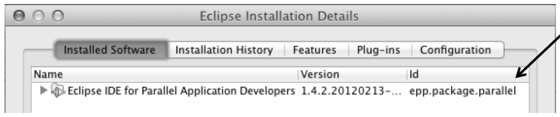
Select "Go to the workbench"


Welcome
Welcome to the Eclipse IDE for Parallel Application Developers
eclipse

Check Installation Details

- To confirm you have installed OK
  - Mac:
  - Others:
- Choose
- Confirm you have the following installed software

Differs depending on base download



Eclipse Installation Details

Name	Version	Id
Eclipse IDE for Parallel Application Developers	1.4.2.20120213-...	epp.package.parallel

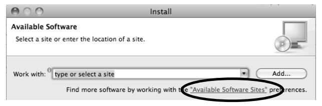
- Close the dialog:

Checking for PTP Updates

- From time-to-time there may be newer PTP releases than the Indigo release
  - Indigo and "Parallel package" updates are released only in Sept and February
- PTP maintains its own update site with the most recent release
  - Bug fix releases can be more frequent than Indigo's and what is within the parallel package
- You must enable the PTP-specific update site before the updates will be found

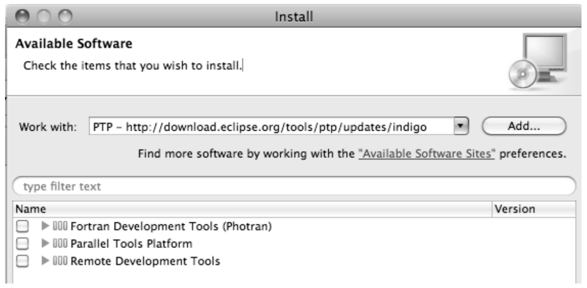
Updating PTP

- Enable PTP-specific update site
  - **Help>Install New Software…**
  - Click **Available Software Sites** link
  - Ensure this checkbox is selected for the PTP site:
    http://download.eclipse.org/tools/ptp/updates/indigo
  - Choose **OK**
  - Choose **Cancel** (to return to Eclipse workbench)
- Now select **Help>Check for updates**
  - If you see "No updates were found"…
  - It's only because there are no updates in the "Eclipse IDE for Parallel Application Developers"
    - We will update the PTP within it


Install
Available Software
Select a site or enter the location of a site.
Work with:
type or select a site
Add...
Find more software by working with the "Available Software Sites" preferences.

Updating PTP (2)

- We will get the PTP release that is more recent than what is currently (Nov. 2011) within the parallel package
- Now select **Help>Install New Software…**
  - In the **Work With:** dropdown box, select the PTP update site you confirmed already:

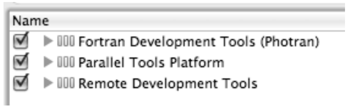

Install
Available Software
Check the items that you wish to install.
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version

Work with: PTP – http://download.eclipse.org/tools/ptp/updates/indigo

- ☐ Fortran Development Tools (Photran)
- ☐ Parallel Tools Platform
- ☐ Remote Development Tools

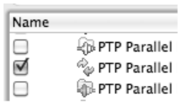
Updating PTP (3)

- Quick and dirty:
  - Check everything - which updates existing features and adds a few more



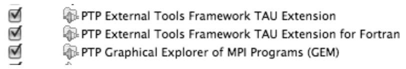
- ☑ Fortran Development Tools (Photran)
- ☑ Parallel Tools Platform
- ☑ Remote Development Tools

- Detailed:
  - Open each feature and check the ones you want to update
  - Icons indicate: Grey plug: already installed and up to date
    Double arrow: can be updated
    Color plug: Not installed yet



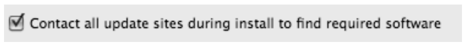
- ☐ PTP Parallel
- ☑ PTP Parallel
- ☐ PTP Parallel

Note: For this tutorial, install GEM and TAU



- ☑ PTP External Tools Framework TAU Extension
- ☑ PTP External Tools Framework TAU Extension for Fortran
- ☑ PTP Graphical Explorer of MPI Programs (GEM)

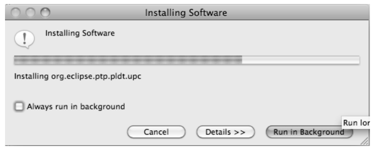
Note: if conference network is slow, consider unchecking:



☑ Contact all update sites during install to find required software

Updating PTP (4)

- Select **Next** to continue updating PTP
- Select **Next** to confirm features to install
- Accept the License agreement and select **Finish**



Installing Software

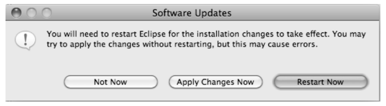
Installing org.eclipse.ptp.pldt.upc

☐ Always run in background

Wait for installation to finish

If conference network is too slow, we have this cached on USB

- Select **Restart Now** when prompted

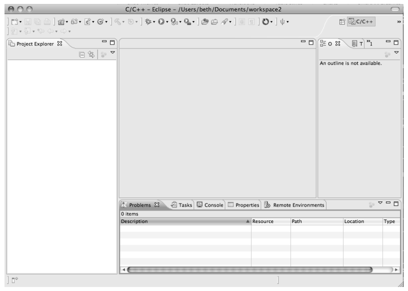


You will need to restart Eclipse for the installation changes to take effect. You may try to apply the changes without restarting, but this may cause errors.

Restart after Install

- ✦ If any top-level features are installed… Welcome page informs you of new features installed
- ✦ We only updated PTP, so we land back at C/C++ Perspective

  … Ready to go!



- ✦ **Help>About** or **Eclipse > About Eclipse** … will indicate the release of PTP installed
- ✦ Further **Help>Check for Updates** will find future updates on the PTP Update site

Module 2: Introduction

- ✦ Objective
  - ✦ To introduce the Eclipse platform and PTP
- ✦ Contents
  - ✦ New and Improved Features
  - ✦ What is Eclipse?
  - ✦ What is PTP?

New and Improved Features

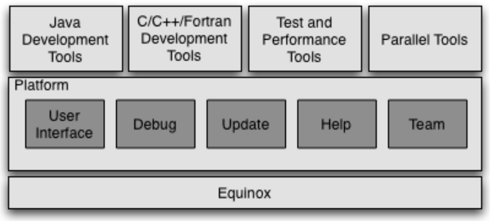
- ✦ More flexible projects
  - ✦ Synchronized projects overcome many problems of remote projects
  - ✦ Allows development when “off-line”
  - ✦ Works with non-C/C++ projects
- ✦ More customizable resource managers
  - ✦ Resource managers can now be added by users
  - ✦ Able to have site-specific configurations
  - ✦ Interactive launch using job schedulers now supported

New and Improved Features (2)

- ✦ Scalable system/job monitoring
  - ✦ New perspective allows monitoring of systems of virtually any size
  - ✦ View shows location of jobs on cluster
  - ✦ Active and inactive jobs views
- ✦ Remote support for performance tools
  - ✦ External Tools Framework has been extended to support remote systems
  - ✦ Performance tools such as TAU can now launch and collect data from remote systems

What is Eclipse?

- A vendor-neutral open-source workbench for multi-language development
- A extensible platform for tool integration
- Plug-in based framework to create, integrate and utilize software tools

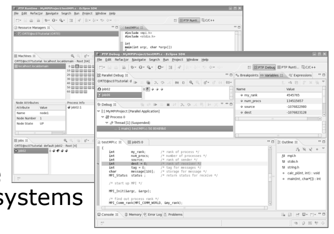


Eclipse Features

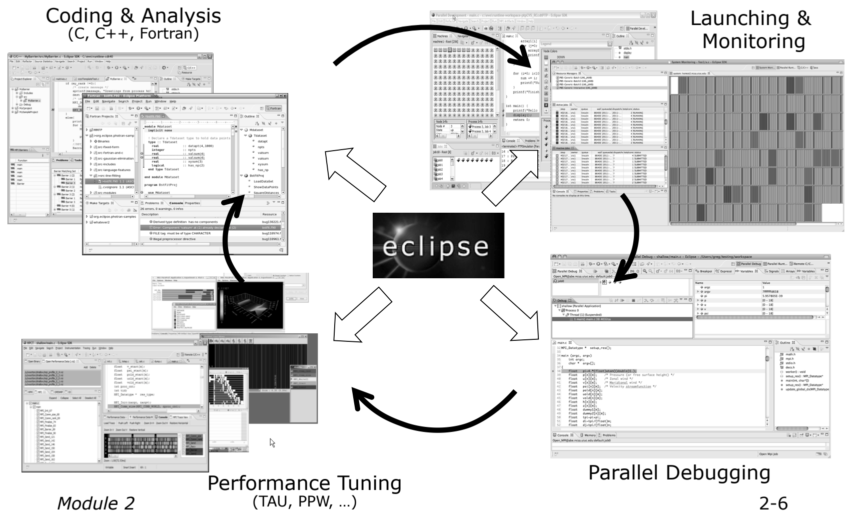
- Full development lifecycle support
- Revision control integration (CVS, SVN, Git)
- Project dependency management
- Incremental building
- Content assistance
- Context sensitive help
- Language sensitive searching
- Multi-language support
- Debugging

Parallel Tools Platform (PTP)

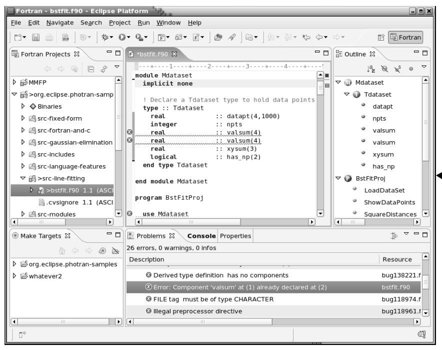
- The Parallel Tools Platform aims to provide a highly integrated environment specifically designed for parallel application development
- Features include:
  - An integrated development environment (IDE) that supports a wide range of parallel architectures and runtime systems
  - A scalable parallel debugger
  - Parallel programming tools (MPI, OpenMP, UPC, etc.)
  - Support for the integration of parallel tools
  - An environment that simplifies the end-user interaction with parallel systems
- http://www.eclipse.org/ptp



Eclipse PTP Family of Tools



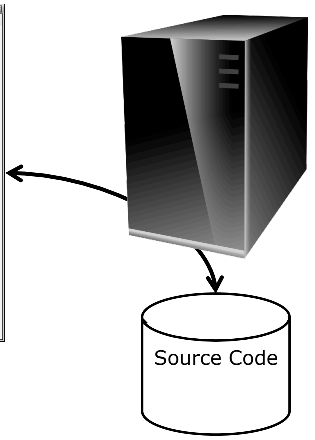
How Eclipse is Used
Editing/Compiling



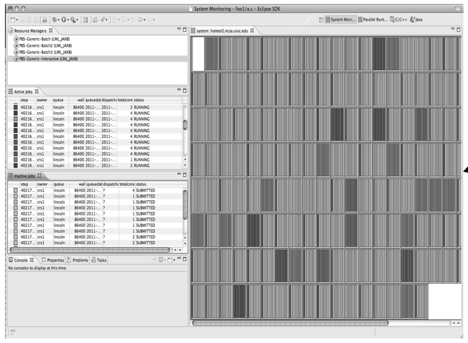
```fortran
module Mdataset
  implicit none

  ! Declare a Tdataset type to hold data points
  type :: Tdataset
    real              :: datapt(4,1000)
    integer           :: npts
    real              :: valsum(4)
    real              :: valsum(4)
    real              :: xysum(3)
    logical           :: has_np(2)
  end type Tdataset

end module Mdataset

program BstFitProj

  use Mdataset
```

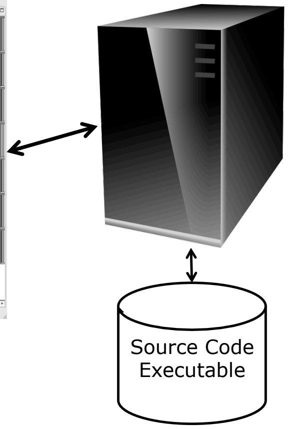
26 errors, 0 warnings, 0 infos

Description	Resource
Derived type definition has no components	bug138221.f
Error: Component 'valsum' at (1) already declared at (2)	bstfit.f90
FILE tag must be of type CHARACTER	bug118974.f
Illegal preprocessor directive	bug118961.f

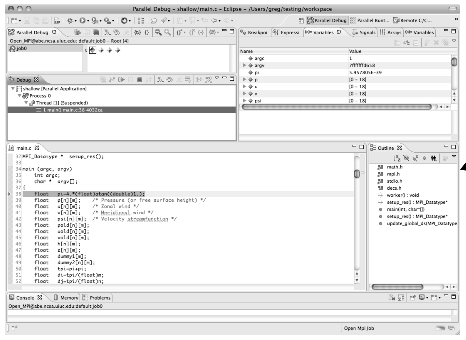


How Eclipse is Used
Launching/Monitoring





How Eclipse is Used
Debugging

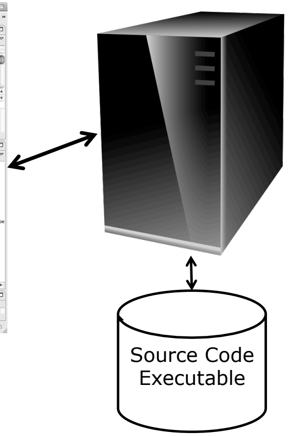


Name	Value
argv	7fffffffd658
pi	5.957805E-39
p	[0 - 18]
u	[0 - 18]
v	[0 - 18]
psi	[0 - 18]

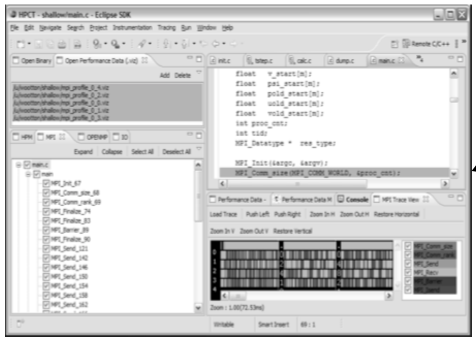
```
shallow [Parallel Application]
  Process 0
    Thread [1] (Suspended)
      1 main() main.c:38 4032ca
```

```c
MPI_Datatype *  setup_res();

main (argc, argv)
    int argc;
    char *  argv[];
{
    float   pi=4.*(float)atan((double)1.);
    float   p[n][m];    /* Pressure (or free surface height) */
    float   u[n][m];    /* Zonal wind */
    float   v[n][m];    /* Meridional wind */
    float   psi[n][m];  /* Velocity streamfunction */
    float   pold[n][m];
    float   uold[n][m];
    float   vold[n][m];
    float   h[n][m];
    float   z[n][m];
    float   dummy1[m];
    float   dummy2[n][m];
    float   tpi=pi+pi;
    float   di=tpi/(float)m;
    float   dj=tpi/(float)n;
```

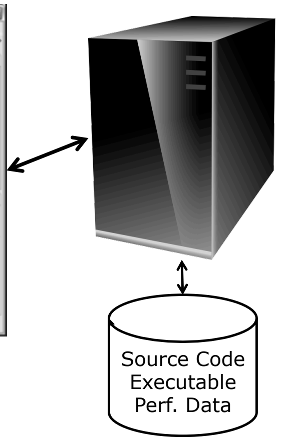
Open_MPI@abe.ncsa.uiuc.edu:default:job0



How Eclipse is Used
Performance Tuning



```c
    float   v_start[m];
    float   psi_start[m];
    float   pold_start[m];
    float   uold_start[m];
    float   vold_start[m];
    int proc_cnt;
    int tid;
    MPI_Datatype *  res_type;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &proc_cnt);
```



Module 3: Developing with Eclipse

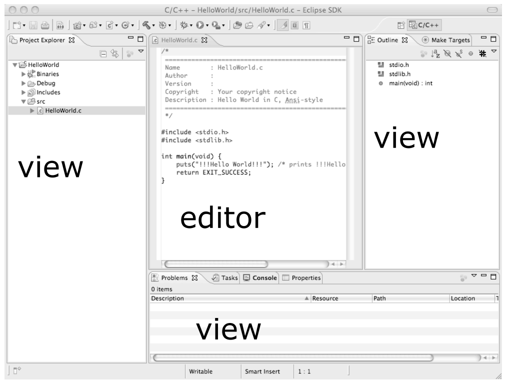
- Objective
  - Learn basic Eclipse concepts: Perspectives, Views, …
  - Learn about local, remote, and synchronized projects
  - Learn how to create and manage a C project
  - Learn about Eclipse editing features
  - Learn about Eclipse Team features
  - Learn about MPI features
  - Learn how to build and launch an MPI program on a remote system
  - Learn about Fortran projects
  - Learn about searching, refactoring, etc.

Contents

Basic Eclipse Features

Eclipse Basics

- A *workbench* contains the menus, toolbars, editors and views that make up the main Eclipse window
- The workbench represents the desktop development environment
  - Contains a set of tools for resource mgmt
  - Provides a common way of navigating through the resources
- Multiple workbenches can be opened at the same time
- Only one workbench can be open on a *workspace* at a time



```
/*
 ==========================================
 Name        : HelloWorld.c
 Author      :
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style
 ==========================================
 */

#include <stdio.h>
#include <stdlib.h>

int main(void) {
	puts("!!!Hello World!!!"); /* prints !!!Hello
	return EXIT_SUCCESS;
}
```

perspective

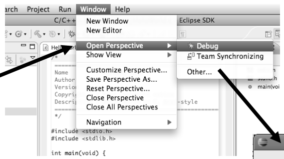
Perspectives

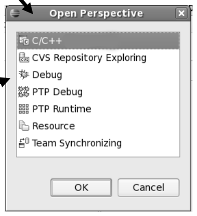
- ✦ Perspectives define the layout of views and editors in the workbench
- ✦ They are *task oriented*, i.e. they contain specific views for doing certain tasks:
  - ✦ There is a for manipulating resources
  - ✦ **C/C++ Perspective** for manipulating compiled code
  - ✦ **Debug Perspective** for debugging applications
- ✦ You can easily switch between perspectives

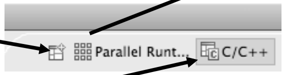
- ✦ If you are on the Welcome screen now, select "Go to Workbench" now

Switching Perspectives

- ✦ Three ways of changing perspectives
  1. Choose the **Perspective** menu option Then choose **Other…**
  2. Click on the **Open Perspective** button in the upper right corner of screen (hover over it to see names)
  3. Click on a perspective shortcut button
- ✦ Switch to the C/C++ Perspective

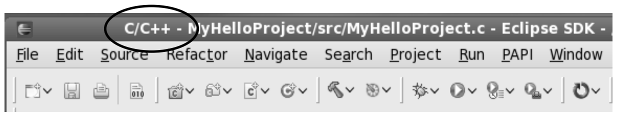




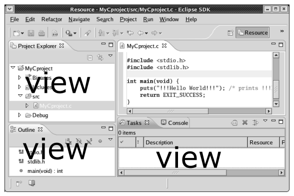


Which Perspective?

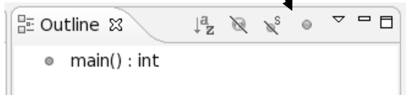
- ✦ Which Perspective am in in? See Title Bar


C/C++ - MyHelloProject/src/MyHelloProject.c - Eclipse SDK -
File Edit Source Refactor Navigate Search Project Run PAPI Window

Views

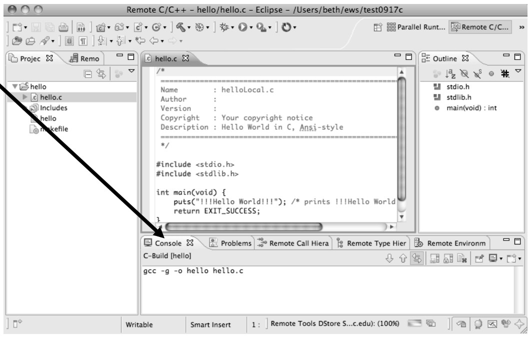


- ✦ The workbench window is divided up into Views
- ✦ The main purpose of a view is:
  - ✦ To provide alternative ways of presenting information
  - ✦ For navigation
  - ✦ For editing and modifying information
- ✦ Views can have their own menus and toolbars
  - ✦ Items available in menus and toolbars are available only in that view
  - ✦ Menu actions only apply to the view
- ✦ Views can be resized



Stacked Views

- ✦ Stacked views appear as tabs
- ✦ Selecting a tab brings that view to the foreground

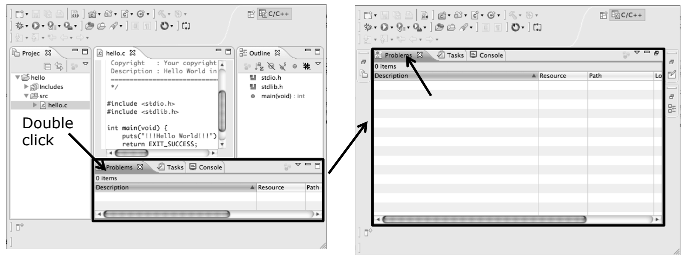


```
/*
 ===========================================
 Name        : helloLocal.c
 Author      :
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style
 ===========================================
 */

#include <stdio.h>
#include <stdlib.h>

int main(void) {
    puts("!!!Hello World!!!"); /* prints !!!Hello World
    return EXIT_SUCCESS;
}
```

```
gcc -g -o hello hello.c
```

Expand a View

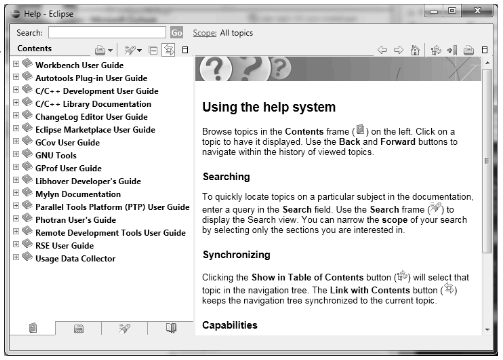
- ✦ Double-click on a view/editor's tab to fill the workbench with its content;
- ✦ Repeat to return to original size



- ✦ Window > Reset Perspective returns everything to original positions

Help

- ✦ To access help
  - ✦ **Help Help Contents**
  - ✦ **Help Search**
  - ✦ **Help>Dynamic Help**
- ✦ **Help Contents** provides detailed help on different Eclipse features *browser*
- ✦ **Search** allows you to search for help locally, or using Google or the Eclipse web site
- ✦ **Dynamic Help** shows help related to the current context (perspective, view, etc.)


Help - Eclipse
Search:
Go
Scope: All topics
Contents
Workbench User Guide
Autotools Plug-in User Guide
C/C++ Development User Guide
C/C++ Library Documentation
ChangeLog Editor User Guide
Eclipse Marketplace User Guide
GCov User Guide
GNU Tools
GProf User Guide
Libhover Developer's Guide
Mylyn Documentation
Parallel Tools Platform (PTP) User Guide
Photran User's Guide
Remote Development Tools User Guide
RSE User Guide
Usage Data Collector

Using the help system

Browse topics in the **Contents** frame ( ) on the left. Click on a topic to have it displayed. Use the **Back** and **Forward** buttons to navigate within the history of viewed topics.

Searching

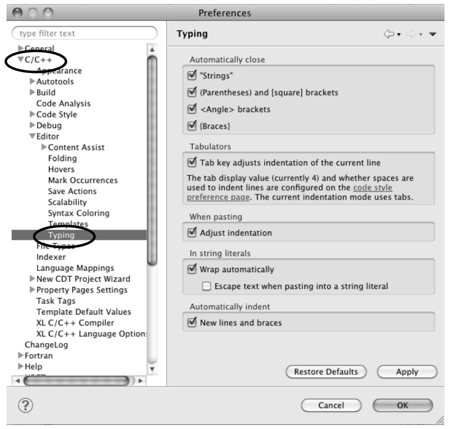
To quickly locate topics on a particular subject in the documentation, enter a query in the **Search** field. Use the **Search** frame ( ) to display the Search view. You can narrow the **scope** of your search by selecting only the sections you are interested in.

Synchronizing

Clicking the **Show in Table of Contents** button ( ) will select that topic in the navigation tree. The **Link with Contents** button ( ) keeps the navigation tree synchronized to the current topic.

Capabilities

Eclipse Preferences



Typing

Automatically close
- [✓] "Strings"
- [✓] (Parentheses) and [square] brackets
- [✓] <Angle> brackets
- [✓] {Braces}

Tabulators
- [✓] Tab key adjusts indentation of the current line

The tab display value (currently 4) and whether spaces are used to indent lines are configured on the code style preference page. The current indentation mode uses tabs.

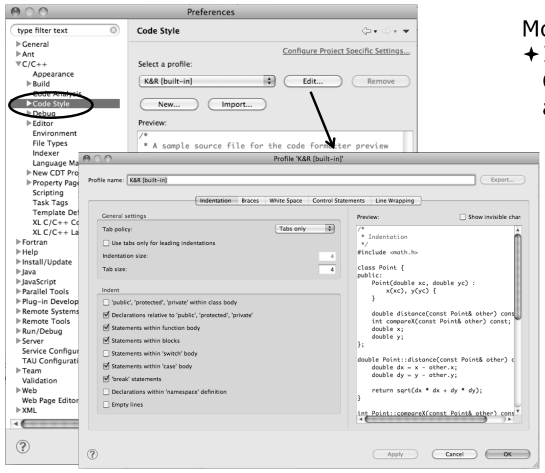
When pasting
- [✓] Adjust indentation

In string literals
- [✓] Wrap automatically
- [ ] Escape text when pasting into a string literal

Automatically indent
- [✓] New lines and braces

- ✦ Eclipse Preferences allow customization of almost everything
- ✦ To open use
  - ✦ Mac:
  - ✦ Others: **Window>Preferences…**
- ✦ The C/C++ preferences allow many options to be altered
- ✦ In this example you can adjust what happens in the editor as you type.

Preferences Example



Select a profile: K&R [built-in]

Preview:

```
/*
 * A sample source file for the code formatter preview
```

Profile name: K&R [built-in]

General settings

- Tab policy: Tabs only
- ☐ Use tabs only for leading indentations
- Indentation size: 4
- Tab size: 4

Indent

- ☐ 'public', 'protected', 'private' within class body
- ☑ Declarations relative to 'public', 'protected', 'private'
- ☑ Statements within function body
- ☑ Statements within blocks
- ☐ Statements within 'switch' body
- ☑ Statements within 'case' body
- ☑ 'break' statements
- ☐ Declarations within 'namespace' definition
- ☐ Empty lines

Preview: ☐ Show invisible char

```
/*
 * Indentation
 */
#include <math.h>

class Point {
public:
	Point(double xc, double yc) :
		x(xc), y(yc) {
	}

	double distance(const Point& other) cons
	int compareX(const Point& other) const;
	double x;
	double y;
};

double Point::distance(const Point& other) c
	double dx = x - other.x;
	double dy = y - other.y;

	return sqrt(dx * dx + dy * dy);
}

int Point::compareX(const Point& other) cons
```

More C/C++ preferences:

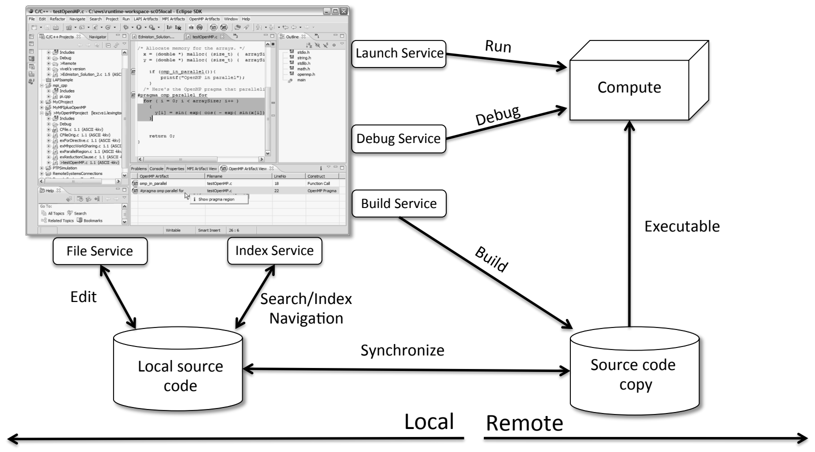
- ✦ In this example the Code Style preferences are shown
  - ✦ These allow code to be automatically formatted in different ways

Projects In Eclipse

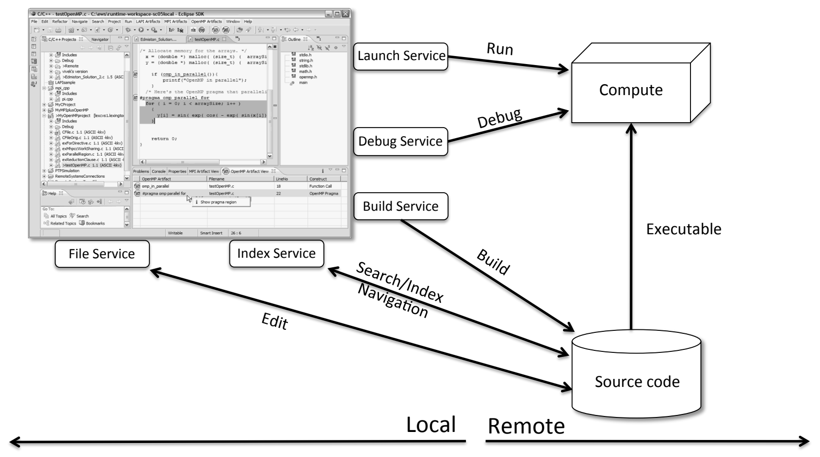
Project Types

- ✦ Local
  - ✦ Source is located on local machine, builds happen locally
- ✦ Synchronized
  - ✦ Source is local, then synchronized with remote machine(s)
  - ✦ Building and launching happens remotely (can also happen locally)
- ✦ Remote
  - ✦ Source is located on remote machine(s), build and launch takes place on remote machine(s)

Synchronized Projects



Remote Projects



C, C++, and Fortran Projects

Build types

- Makefile-based
  - Project contains its own makefile (or makefiles) for building the application
- Managed
  - Eclipse manages the build process, no makefile required

Parallel programs can be run on local machine or on a remote system

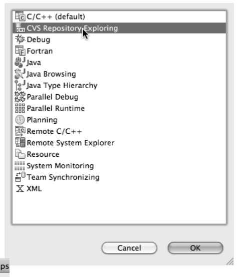
- MPI (or other runtime) needs to be installed
- An application built locally probably can't be run on a remote machine unless their architectures are the same

Checking out the project

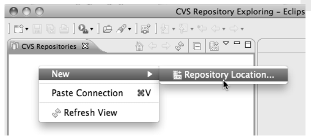
Using a Source Code Repository Introduction to Team Features

Importing a Project from CVS

- Switch to **Exploring** perspective
  - Window > Open Perspective > Other…
  - Select **CVS Repository Exploring**
  - Select **OK**

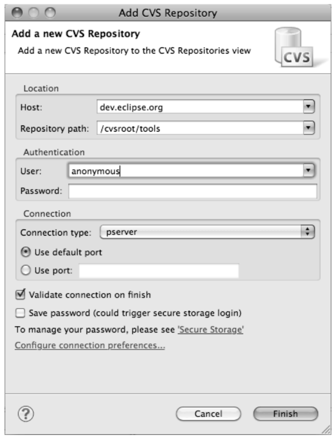


- Right click in **CVS Repositories** view and select **New>Repository Location…**



Add CVS Repository

- ✦ Enter **Host:** dev.eclipse.org
- ✦ **Repository path:** /cvsroot/tools
- ✦ For anonymous access:
  - ✦ **User**
  - ✦ No password is required
  - ✦ **Connection type:** pserver (default)
- ✦ For authorized access:
  - ✦ **User**: your userid
  - ✦ **Password**: your password
  - ✦ **Connection type:** change to **extssh**
- ✦ Select **Finish**


Add CVS Repository

Add a new CVS Repository
Add a new CVS Repository to the CVS Repositories view

Location
Host: dev.eclipse.org
Repository path: /cvsroot/tools

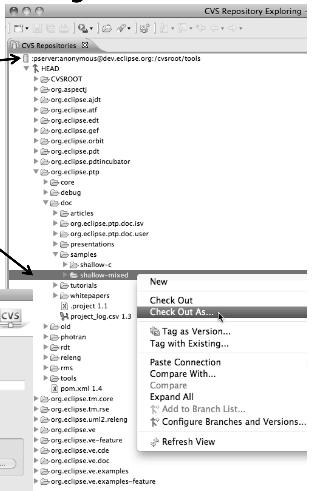
Authentication
User: anonymous
Password:

Connection
Connection type: pserver
(●) Use default port
( ) Use port:

☑ Validate connection on finish
☐ Save password (could trigger secure storage login)
To manage your password, please see 'Secure Storage'
Configure connection preferences...

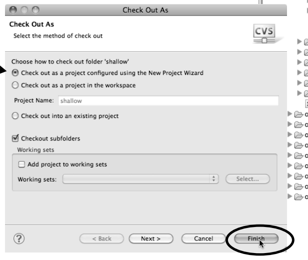
Checking out the Project

- ✦ Expand the repository location
- ✦ Expand **HEAD**
- ✦ Expand **org.eclipse.ptp, doc, and samples**
- ✦ Right click on **shallow-mixed** and select **Check Out As…**
- ✦ On **Check Out As** dialog, select **Finish**


CVS Repository Exploring -

CVS Repositories
```
:pserver:anonymous@dev.eclipse.org:/cvsroot/tools
  HEAD
    CVSROOT
    org.aspectj
    org.eclipse.ajdt
    org.eclipse.atf
    org.eclipse.edt
    org.eclipse.gef
    org.eclipse.orbit
    org.eclipse.pdt
    org.eclipse.pdtincubator
    org.eclipse.ptp
      core
      debug
      doc
        articles
        org.eclipse.ptp.doc.isv
        org.eclipse.ptp.doc.user
        presentations
        samples
          shallow-c
          shallow-mixed
        tutorials
        whitepapers
        .project 1.1
        project_log.csv 1.3
      old
      photran
      rdt
      releng
      rms
      tools
      pom.xml 1.4
    org.eclipse.tm.core
    org.eclipse.tm.rse
    org.eclipse.uml2.releng
    org.eclipse.ve
    org.eclipse.ve-feature
    org.eclipse.ve.cde
    org.eclipse.ve.doc
    org.eclipse.ve.examples
    org.eclipse.ve.examples-feature
```

The default of "Check out as a project configured using the New Project Wizard" is what we want


Check Out As

Check Out As
Select the method of check out

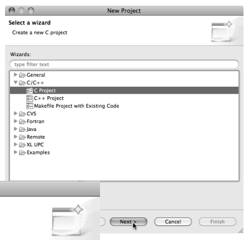
Choose how to check out folder 'shallow'
(●) Check out as a project configured using the New Project Wizard
( ) Check out as a project in the workspace
Project Name: shallow
( ) Check out into an existing project

☑ Checkout subfolders
Working sets
☐ Add project to working sets
Working sets:

New Project Wizard

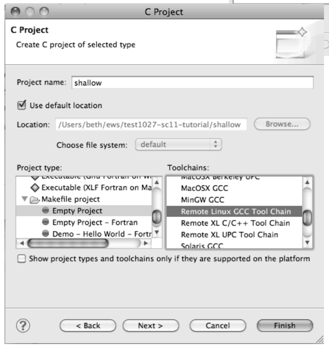
As project is checked out from CVS, the Wizard helps you configure the Eclipse information to be added to the project

- ✦Expand
- ✦Select and click on
- ✦Enter shallow as
- ✦ Under expand
  - scroll to the bottom
- ✦ Select
- ✦ Select a toolchain that matches your system from **Toolchains**
  - ✦ Since we will build/run this on the remote system, choose an appropriate toolchain
  - ✦ You may need to uncheck "Show project types and toolchains only if they are supported on the platform"
- ✦ Click on **Finish**


New Project

Select a wizard
Create a new C project

Wizards:
type filter text
```
General
C/C++
  C Project
  C++ Project
  Makefile Project with Existing Code
CVS
Fortran
Java
Remote
XL UPC
Examples
```


C Project

C Project
Create C project of selected type

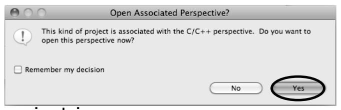
Project name: shallow
☑ Use default location
Location: /Users/beth/ews/test1027-sc11-tutorial/shallow
Choose file system: default

Project type:
```
Executable (Gnu Fortran on Wi
Executable (XLF Fortran on Ma
Makefile project
  Empty Project
  Empty Project - Fortran
  Demo - Hello World - Fortr
```
Toolchains:
```
MacOSX Berkeley UPC
MacOSX GCC
MinGW GCC
Remote Linux GCC Tool Chain
Remote XL C/C++ Tool Chain
Remote XL UPC Tool Chain
Solaris GCC
```
☐ Show project types and toolchains only if they are supported on the platform

For SC tutorial Choose Remote Linux GCC Tool Chain

C/C++ Perspective

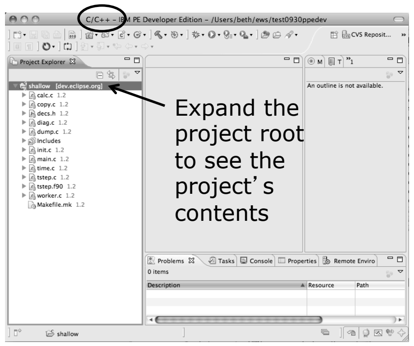
- ✦ Switch to the C/C++ Perspective when Prompted
- ✦ You should now see the shallow project in your workspace


Open Associated Perspective?

This kind of project is associated with the C/C++ perspective. Do you want to open this perspective now?

☐ Remember my decision

No [✓ Yes]


C/C++ - IBM PE Developer Edition - /Users/beth/ews/test0930ppedev

Project Explorer
```
shallow [dev.eclipse.org]
  calc.c 1.2
  copy.c 1.2
  decs.h 1.2
  diag.c 1.2
  dump.c 1.2
  Includes
  init.c 1.2
  main.c 1.2
  time.c 1.2
  tstep.c 1.2
  tstep.f90 1.2
  worker.c 1.2
  Makefile.mk 1.2
```

An outline is not available.

Problems: 0 items

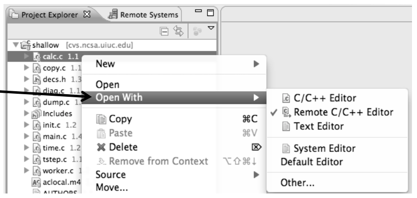
Description	Resource	Path

Expand the project root to see the project' s contents

Editor Features

Editors

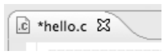
- An editor for a resource (e.g. a file) opens when you double-click on a resource
- The type of editor depends on the type of the resource
  - .c files are opened with the C/C++ editor by default
  - You can use to use another editor
  - In this case the default editor is fine (double-click)



  - Some editors do not just edit raw text
- When an editor opens on a resource, it stays open across different perspectives
- An active editor contains menus and toolbars specific to that editor

Saving File in Editor

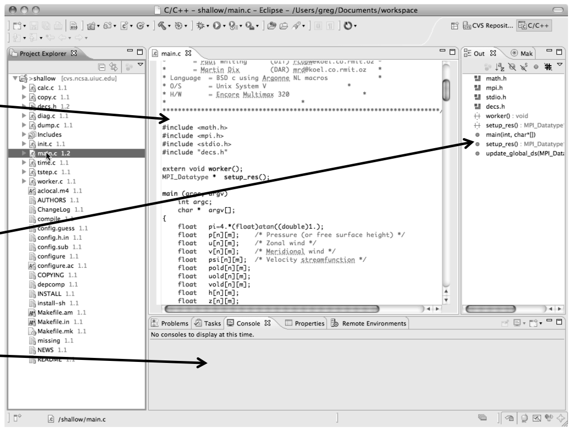
- When you change a file in the editor, an asterisk on the editor's title bar indicates unsaved changes



- Save the changes by using Command/Ctrl-S or
- Undo last change using **Command/Ctrl Z**

Editor and Outline View

- Double-click on source file
- Editor will open in main view
- Outline view is shown for file in editor
- Console shows results of build, local runs, etc.

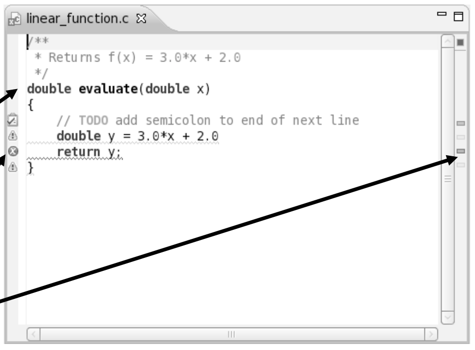


```
 *        = Martin Dix         (DAR) mrd@koel.co.rmit.oz    *
 * Language  = BSD c using Argonne NL macros                *
 * O/S       = Unix System V                        *
 * H/W       = Encore Multimax 320                      *
 *                                        *
 ***************************************************************************/

#include <math.h>
#include <mpi.h>
#include <stdio.h>
#include "decs.h"

extern void worker();
MPI_Datatype *  setup_res();

main (argc, argv)
    int argc;
    char *  argv[];
{
    float   pi=4.*(float)atan((double)1.);
    float   p[n][m];    /* Pressure (or free surface height) */
    float   u[n][m];    /* Zonal wind */
    float   v[n][m];    /* Meridional wind */
    float   psi[n][m];  /* Velocity streamfunction */
    float   pold[n][m];
    float   uold[n][m];
    float   vold[n][m];
    float   h[n][m];
    float   z[n][m];
```

No consoles to display at this time.

Source Code Editors & Markers

- A source code editor is a special type of editor for manipulating source code
- Language features are highlighted
- Marker bars for showing
  - Breakpoints
  - Errors/warnings
  - Task Tags, Bookmarks
- Location bar for navigating to interesting features in the entire file



```
/**
 * Returns f(x) = 3.0*x + 2.0
 */
double evaluate(double x)
{
    // TODO add semicolon to end of next line
    double y = 3.0*x + 2.0
    return y;
}
```

Icons:

✓	Task tag
⚠	Warning
⊗	Error

Code Analysis (Codan)

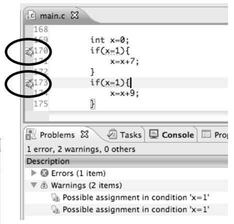
- If you see bug icons in the editor marker bar, they are likely suggestions from Codan
- Code checkers can flag possible errors, even if code is technically correct
- To turn them off, use Preferences

  Window > Preferences or Mac: Eclipse > Preferences

  C/C++ > Code Analysis

  and uncheck all problems
- Select OK to close Preferences

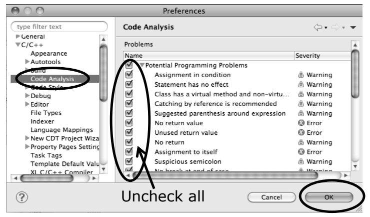
✦To remove icons: Rightmouse on Project > Run C/C++ Code Analysis



```
168
169         int x=0;
170         if(x=1){
171             x=x+7;
172         }
173         if(x=1){
174             x=x+9;
175         }
```

1 error, 2 warnings, 0 others

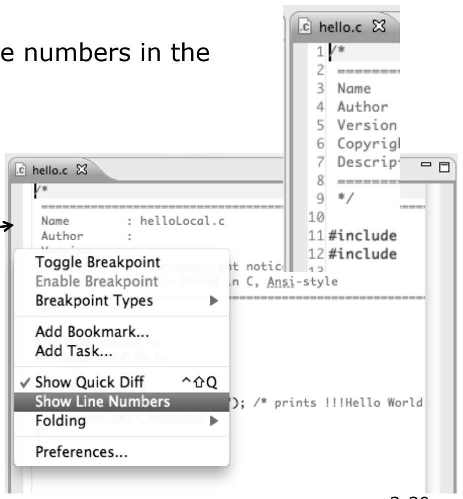
Description
- Errors (1 item)
- Warnings (2 items)
  - Possible assignment in condition 'x=1'
  - Possible assignment in condition 'x=1'



Name	Severity
☑ Potential Programming Problems	
☑ Assignment in condition	Warning
☑ Statement has no effect	Warning
☑ Class has a virtual method and non-virtu...	Warning
☑ Catching by reference is recommended	Warning
☑ Suggested parenthesis around expression	Warning
☑ No return value	Error
☑ Unused return value	Error
☑ No return	Warning
☑ Assignment to itself	Error
☑ Suspicious semicolon	Warning
☑ No break at end of case	Warning

Line Numbers

- Text editors can show line numbers in the left column
- To turn on line numbering:
  - Right-mouse click in the editor marker bar
  - Click on **Numbers**

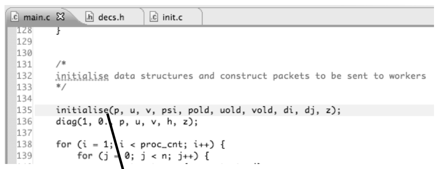


```
/*
 ==========
 Name        : helloLocal.c
 Author      :
 ...t notic...
 ...n C, Ansi-style
 ==========
); /* prints !!!Hello World
```

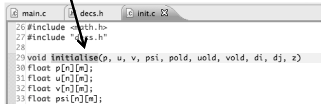
```
1 /*
2  =======
3  Name
4  Author
5  Version
6  Copyrig
7  Descrip
8  =======
9  */
10
11 #include
12 #include
```

Navigating to Other Files

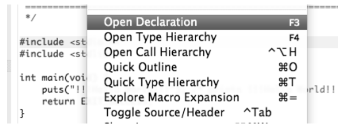
- On demand hyperlink
  - In main.c line 135:
  - Hold down Command/Ctrl key e.g. on call to `initialise`
  - Click on `initialise` to navigate to its definition in the header file (Exact key combination depends on your OS)
  - E.g. Command/Ctrl and click on initialise
- Open declaration
  - Right-click and select **Open Declaration** will also open the file in which the element is declared
  - E.g. in main.c line 29 right-click on decs.h and select **Open Declaration**



```
128     }
129
130
131     /*
132     initialise data structures and construct packets to be sent to workers
133     */
134
135     initialise(p, u, v, psi, pold, uold, vold, di, dj, z);
136     diag(1, 0., p, u, v, h, z);
137
138     for (i = 1; i < proc_cnt; i++) {
139         for (j = 0; j < n; j++) {
```



```
26 #include <math.h>
27 #include "decs.h"
28
29 void initialise(p, u, v, psi, pold, uold, vold, di, dj, z)
30 float p[n][m];
31 float u[n][m];
32 float v[n][m];
33 float psi[n][m];
```

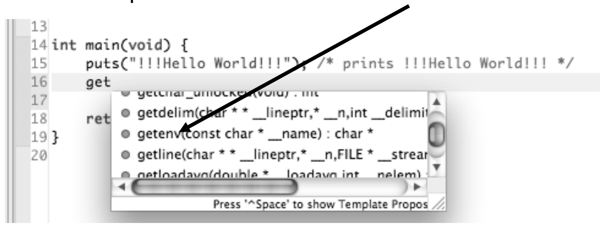


```
 */

#include <st
#include <st

int main(voi
    puts("!!                 rld!!
    return E
}
```

Note: may need to left-click before right-click works

Content Assist & Templates

- Type an incomplete function name e.g. get into the editor, and hit **ctrl-space**
- Select desired completion value with cursor or mouse

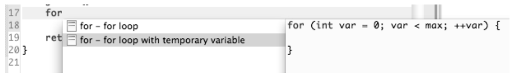


```
int main(void) {
    puts("!!!Hello World!!!"); /* prints !!!Hello World!!! */
    get
    ...
    ret
}
```

```
getchar_unlocked(void) : int
getdelim(char * * __lineptr,* __n,int __delimit
getenv(const char * __name) : char *
getline(char * * __lineptr,* __n,FILE * __strear
getloadavg(double * __loadavg,int __nelem)
Press '^Space' to show Template Propos
```

- Code Templates: type 'for' and Ctrl-space

Hit ctrl-space again for code templates



```
    for
    ret
}
```

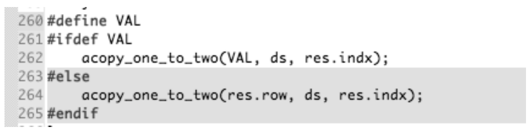
```
for - for loop
for - for loop with temporary variable
```

```
for (int var = 0; var < max; ++var) {

}
```

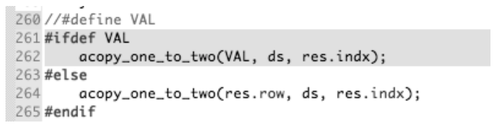
More info on code templates later

Inactive code

- Inactive code will appear grayed out in the CDT editor



```
#define VAL
#ifdef VAL
    acopy_one_to_two(VAL, ds, res.indx);
#else
    acopy_one_to_two(res.row, ds, res.indx);
#endif
```



```
//#define VAL
#ifdef VAL
    acopy_one_to_two(VAL, ds, res.indx);
#else
    acopy_one_to_two(res.row, ds, res.indx);
#endif
```

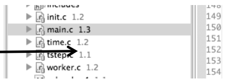
Team Features

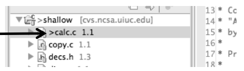
"Team" Features

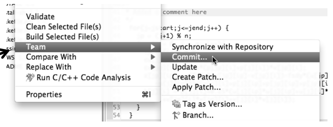
- Eclipse supports integration with multiple version control systems (VCS)
  - CVS, SVN, Git, and others
  - Collectively known as "Team" services
- Many features are common across VCS
  - Compare/merge
  - History
  - Check-in/check-out
- Some differences
  - Version numbers
  - Branching

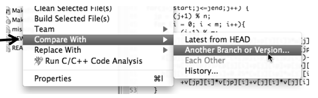
CVS Features

- ✦ Shows version numbers next to each resource
- ✦ Marks resources that have changed
  - ✦ Can also change color (preference option)
- ✦ Context menu for Team operations
- ✦ Compare to latest, another branch, or history
- ✦ Synchronize whole project (or any selected resources)



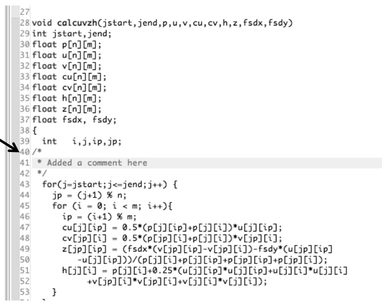




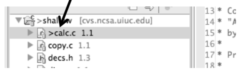


File Modification

- ✦ Open "calc.c"
- ✦ Add comment at line 40
- ✦ Save file
- ✦ File will be marked to indicate that is has been modified

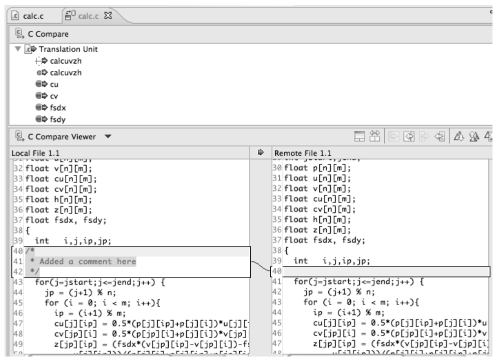


```
27
28 void calcuvzh(jstart,jend,p,u,v,cu,cv,h,z,fsdx,fsdy)
29 int jstart,jend;
30 float p[n][m];
31 float u[n][m];
32 float v[n][m];
33 float cu[n][m];
34 float cv[n][m];
35 float h[n][m];
36 float z[n][m];
37 float fsdx, fsdy;
38 {
39   int   i,j,ip,jp;
40 /*
41  * Added a comment here
42  */
43   for(j=jstart;j<=jend;j++) {
44     jp = (j+1) % n;
45     for (i = 0; i < m; i++){
46       ip = (i+1) % m;
47       cu[j][ip] = 0.5*(p[j][ip]+p[j][i])*u[j][ip];
48       cv[jp][i] = 0.5*(p[jp][i]+p[j][i])*v[jp][i];
49       z[jp][ip] = (fsdx*(v[jp][ip]-v[jp][i])-fsdy*(u[jp][ip]
50          -u[j][ip]))/(p[j][i]+p[j][ip]+p[jp][ip]+p[jp][i]);
51       h[j][i] = p[j][i]+0.25*(u[j][ip]*u[j][ip]+u[j][i]*u[j][i]
52            +v[jp][i]*v[jp][i]+v[j][i]*v[j][i]);
53     }
```



View Changes

- ✦ Right-click on calc.c and select **Compare With>Latest from HEAD**
- ✦ Compare editor will open showing differences between local (changed) file and the original
- ✦ Buttons allow changes to be merged from right to left
- ✦ Can also navigate between changes using buttons

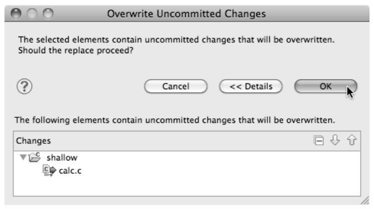


```
float v[n][m];
float cu[n][m];
float cv[n][m];
float h[n][m];
float z[n][m];
float fsdx, fsdy;
{
  int   i,j,ip,jp;
/*
 * Added a comment here
 */
  for(j=jstart;j<=jend;j++) {
    jp = (j+1) % n;
    for (i = 0; i < m; i++){
      ip = (i+1) % m;
      cu[j][ip] = 0.5*(p[j][ip]+p[j][i])*u[j][
      cv[jp][i] = 0.5*(p[jp][i]+p[j][i])*v[jp]
      z[jp][ip] = (fsdx*(v[jp][ip]-v[jp][i])-f
```

```
float p[n][m];
float u[n][m];
float v[n][m];
float cu[n][m];
float cv[n][m];
float h[n][m];
float z[n][m];
float fsdx, fsdy;
{
  int   i,j,ip,jp;

  for(j=jstart;j<=jend;j++) {
    jp = (j+1) % n;
    for (i = 0; i < m; i++){
      ip = (i+1) % m;
      cu[j][ip] = 0.5*(p[j][ip]+p[j][i])*u
      cv[jp][i] = 0.5*(p[jp][i]+p[j][i])*v
      z[jp][ip] = (fsdx*(v[jp][ip]-v[jp][i]
```

Revert To The Latest Version

- ✦ Right-click on the "shallow" project and select **Replace With>Latest from HEAD**
- ✦ Review the resources that will be replaced, then click **OK**



Overwrite Uncommitted Changes

The selected elements contain uncommitted changes that will be overwritten. Should the replace proceed?

The following elements contain uncommitted changes that will be overwritten.

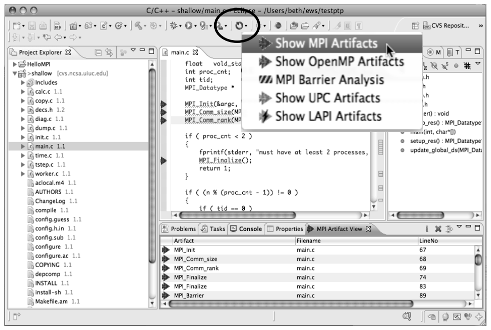
MPI Features

MPI-Specific Features

- PTP's Parallel Language Development Tools (PLDT) has several features specifically for developing MPI code
  - Show MPI Artifacts
  - Code completion
  - Context Sensitive Help for MPI
  - Hover Help
  - MPI Templates in the editor
  - MPI Barrier Analysis

Show MPI Artifacts

- In Project Explorer, select a project, folder, or a single source file
  - The analysis will be run on the selected resources
- Select **Artifacts**
- Run the analysis by clicking on drop-down menu next to the analysis button
- Works on local and remote files

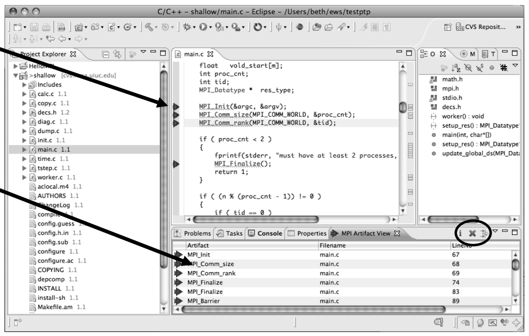


```
float   vold_sta
int proc_cnt;
int tid;
MPI_Datatype *

MPI_Init(&argc,
MPI_Comm_size(MP
MPI_Comm_rank(MP

if ( proc_cnt < 2 )
{
    fprintf(stderr, "must have at least 2 processes,
    MPI_Finalize();
    return 1;
}

if ( (n % (proc_cnt - 1)) != 0 )
{
    if ( tid == 0 )
```

Artifact	Filename	LineNo
MPI_Init	main.c	67
MPI_Comm_size	main.c	68
MPI_Comm_rank	main.c	69
MPI_Finalize	main.c	74
MPI_Finalize	main.c	83
MPI_Barrier	main.c	89

MPI Artifact View

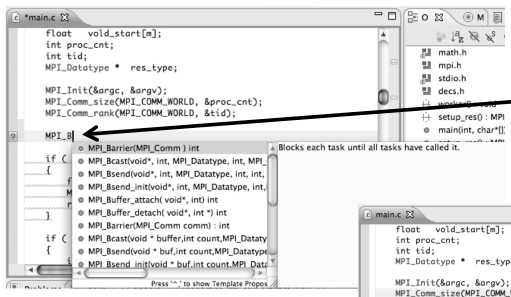
- Markers indicate the location of artifacts in editor
- The **MPI Artifact View** lists the type and location of each artifact
- Navigate to source code line by double-clicking on the artifact
- Run the analysis on another file (or entire project!) and its markers will be added to the view
- Click on column headings to sort
- Remove markers via ✖


C/C++ – shallow/main.c – Eclipse – /Users/beth/ews/testptp
Project Explorer
main.c
math.h
mpi.h
stdio.h
decs.h
worker() : void
setup_res() : MPI_Datatype
main(int, char*[])
setup_res() : MPI_Datatype
update_global_ds(MPI_Data
Problems
Tasks
Console
Properties
MPI Artifact View

```
float   vold_start[m];
int proc_cnt;
int tid;
MPI_Datatype *  res_type;

MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &proc_cnt);
MPI_Comm_rank(MPI_COMM_WORLD, &tid);

if ( proc_cnt < 2 )
{
    fprintf(stderr, "must have at least 2 processes,
    MPI_Finalize();
    return 1;
}

if ( (n % (proc_cnt - 1)) != 0 )
{
    if ( tid == 0 )
```

Artifact	Filename	LineNo
MPI_Init	main.c	67
MPI_Comm_size	main.c	68
MPI_Comm_rank	main.c	69
MPI_Finalize	main.c	74
MPI_Finalize	main.c	83
MPI_Barrier	main.c	89

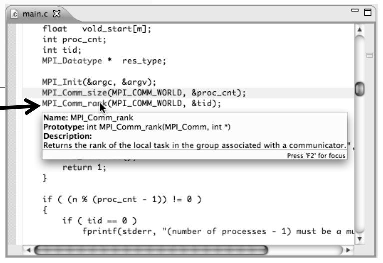
MPI Editor Features



```
float   vold_start[m];
int proc_cnt;
int tid;
MPI_Datatype *  res_type;

MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &proc_cnt);
MPI_Comm_rank(MPI_COMM_WORLD, &tid);

MPI_B
```

*main.c
math.h
mpi.h
stdio.h
decs.h
setup_res() : MPI
main(int, char*[])
MPI_Barrier(MPI_Comm) int
MPI_Bcast(void*, int, MPI_Datatype, int, MPI_
MPI_Bsend(void*, int, MPI_Datatype, int, int,
MPI_Bsend_init(void*, int, MPI_Datatype, int,
MPI_Buffer_attach(void*, int) int
MPI_Buffer_detach(void*, int *) int
MPI_Barrier(MPI_Comm comm) : int
MPI_Bcast(void * buffer,int count,MPI_Dataty
MPI_Bsend(void * buf,int count,MPI_Datatype
MPI_Bsend_init(void * buf,int count,MPI_Data
Blocks each task until all tasks have called it.
Press '^,' to show Template Propos

- ✦ Code completion will show all the possible MPI keyword completions
- ✦ Enter the start of a keyword then press <ctrl-space>



```
float   vold_start[m];
int proc_cnt;
int tid;
MPI_Datatype *  res_type;

MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &proc_cnt);
MPI_Comm_rank(MPI_COMM_WORLD, &tid);
```

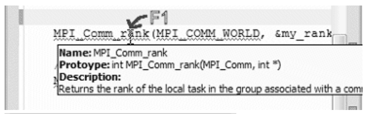
Name: MPI_Comm_rank
Prototype: int MPI_Comm_rank(MPI_Comm, int *)
Description:
Returns the rank of the local task in the group associated with a communicator."
Press 'F2' for focus

```
    return 1;
}

if ( (n % (proc_cnt - 1)) != 0 )
{
    if ( tid == 0 )
        fprintf(stderr, "(number of processes - 1) must be a mu
```

- ✦ Hover over MPI API
- ✦ Displays the function prototype and a description

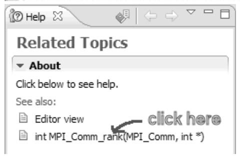
Context Sensitive Help

- ✦ Click mouse, then press help key when the cursor is within a function name
  - ✦ Windows: **F1** key
  - ✦ Linux: **ctrl-F1** key
  - ✦ MacOS X: **Help** key or **Help▸Dynamic Help**
- ✦ A help view appears (**Related Topics**) which shows additional information (You may need to click on MPI API in editor again, to populate)
- ✦ Click on the function name to see more information
- ✦ Move the help view within your Eclipse workbench, if you like, by dragging its title tab

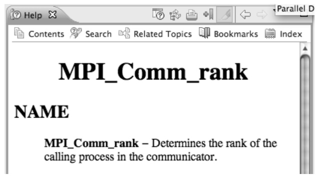


```
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank
```

Name: MPI_Comm_rank
Protoype: int MPI_Comm_rank(MPI_Comm, int *)
Description:
Returns the rank of the local task in the group associated with a comm



Some special info has been added for MPI APIs



MPI_Comm_rank

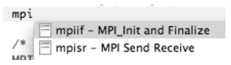
NAME

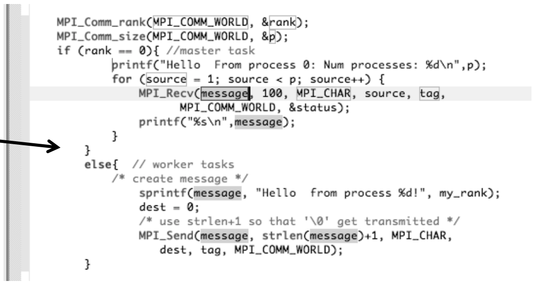
MPI_Comm_rank – Determines the rank of the calling process in the communicator.

MPI Templates

✦Allows quick entry of common patterns in MPI programming

- ✦ Example:
  MPI send-receive
- ✦ Enter:
  `mpisr` <ctrl-space>
- ✦ Expands to a send-receive pattern
- ✦ Highlighted variable names can all be changed at once
- ✦ Type `mpi` <ctrl-space> <ctrl-space> to see all templates



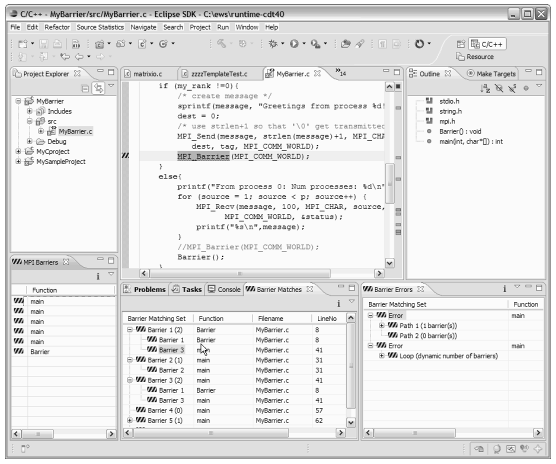


```
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &p);
if (rank == 0){ //master task
        printf("Hello  From process 0: Num processes: %d\n",p);
        for (source = 1; source < p; source++) {
            MPI_Recv(message, 100, MPI_CHAR, source, tag,
                    MPI_COMM_WORLD, &status);
            printf("%s\n",message);
        }
    }
    else{  // worker tasks
        /* create message */
            sprintf(message, "Hello  from process %d!", my_rank);
            dest = 0;
            /* use strlen+1 so that '\0' get transmitted */
            MPI_Send(message, strlen(message)+1, MPI_CHAR,
                dest, tag, MPI_COMM_WORLD);
    }
```

Add more templates using Eclipse preferences!
C/C++>Editor>Templates
Extend to other common patterns

MPI Barrier Analysis

Local files only



```
if (my_rank !=0){
    /* create message */
    sprintf(message, "Greetings from process %d!
    dest = 0;
    /* use strlen+1 so that '\0' get transmittec
    MPI_Send(message, strlen(message)+1, MPI_CHA
       dest, tag, MPI_COMM_WORLD);
    MPI_Barrier(MPI_COMM_WORLD);
}
else{
    printf("From process 0: Num processes: %d\n"
    for (source = 1; source < p; source++) {
        MPI_Recv(message, 100, MPI_CHAR, source,
              MPI_COMM_WORLD, &status);
        printf("%s\n",message);
    }
    //MPI_Barrier(MPI_COMM_WORLD);
    Barrier();
```

	Function
▨	main
▨	main
▨	main
▨	main
▨	main
▨	Barrier

Barrier Matching Set	Function	Filename	LineNo
Barrier 1 (2)	Barrier	MyBarrier.c	8
Barrier 1	Barrier	MyBarrier.c	8
Barrier 3	main	MyBarrier.c	41
Barrier 2 (1)	main	MyBarrier.c	31
Barrier 2	main	MyBarrier.c	31
Barrier 3 (2)	main	MyBarrier.c	41
Barrier 1	Barrier	MyBarrier.c	8
Barrier 3	main	MyBarrier.c	41
Barrier 4 (0)	main	MyBarrier.c	57
Barrier 5 (1)	main	MyBarrier.c	62

Barrier Matching Set	Function
Error	main
Path 1 (1 barrier(s))	
Path 2 (0 barrier(s))	
Error	main
Loop (dynamic number of barriers)	

Verify barrier synchronization in C/MPI programs

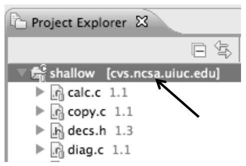
Interprocedural static analysis outputs:

- ✦For verified programs, lists barrier statements that synchronize together (match)
- ✦ For synchronization errors, reports counter example that illustrates and explains the error

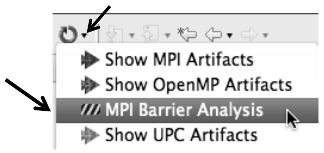
MPI Barrier Analysis – Try it

Run the Analysis:

✦ In the Project Explorer, select the project (or directory, or file) to analyze

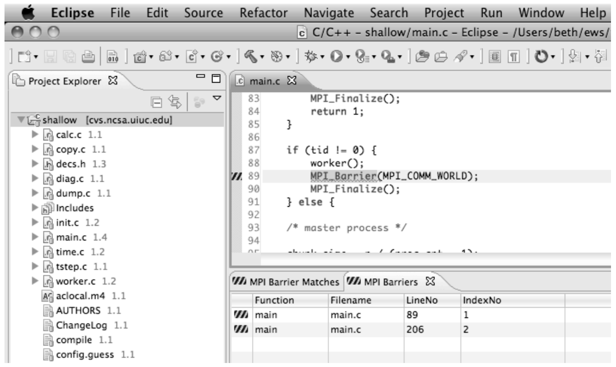

Project Explorer
shallow [cvs.ncsa.uiuc.edu]
calc.c 1.1
copy.c 1.1
decs.h 1.3
diag.c 1.1

✦ Select the MPI Barrier Analysis action in the pull-down menu


Show MPI Artifacts
Show OpenMP Artifacts
MPI Barrier Analysis
Show UPC Artifacts

MPI Barrier Analysis – Try It (2)

✦ No Barrier Errors are found (no pop-up indicating error); Two barriers are found


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
C/C++ – shallow/main.c – Eclipse – /Users/beth/ews/

Project Explorer
shallow [cvs.ncsa.uiuc.edu]
calc.c 1.1
copy.c 1.1
decs.h 1.3
diag.c 1.1
dump.c 1.1
Includes
init.c 1.2
main.c 1.4
time.c 1.2
tstep.c 1.1
worker.c 1.2
aclocal.m4 1.1
AUTHORS 1.1
ChangeLog 1.1
compile 1.1
config.guess 1.1

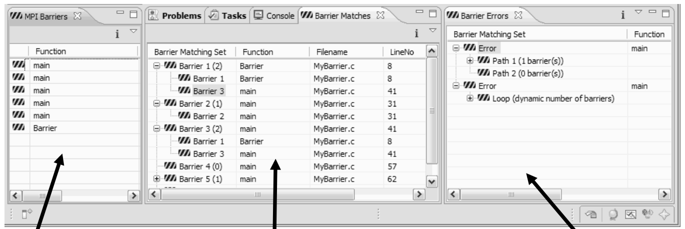
main.c

```
83          MPI_Finalize();
84          return 1;
85      }
86
87      if (tid != 0) {
88          worker();
89          MPI_Barrier(MPI_COMM_WORLD);
90          MPI_Finalize();
91      } else {
92
93      /* master process */
94
```

MPI Barrier Matches | MPI Barriers

Function	Filename	LineNo	IndexNo
main	main.c	89	1
main	main.c	206	2

MPI Barrier Analysis - views


MPI Barriers

Function
main
main
main
main
main
Barrier

Problems | Tasks | Console | Barrier Matches

Barrier Matching Set	Function	Filename	LineNo
Barrier 1 (2)	Barrier	MyBarrier.c	8
Barrier 1	Barrier	MyBarrier.c	8
Barrier 3	main	MyBarrier.c	41
Barrier 2 (1)	main	MyBarrier.c	31
Barrier 2	main	MyBarrier.c	31
Barrier 3 (2)	main	MyBarrier.c	41
Barrier 1	Barrier	MyBarrier.c	8
Barrier 3	main	MyBarrier.c	41
Barrier 4 (0)	main	MyBarrier.c	57
Barrier 5 (1)	main	MyBarrier.c	62

Barrier Errors

Barrier Matching Set	Function
Error	main
Path 1 (1 barrier(s))	
Path 2 (0 barrier(s))	
Error	main
Loop (dynamic number of barriers)	

MPI Barriers view

Simply lists the barriers

Like MPI Artifacts view, double-click to navigate to source code line (all 3 views)

Barrier Matches view
Groups barriers that match together in a barrier set – all processes must go through a barrier in the set to prevent a deadlock

Barrier Errors view

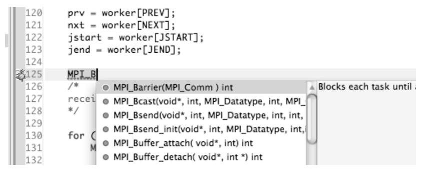
If there are errors, a counter-example shows paths with mismatched number of barriers

Barrier Errors

✦ Let’s cause a barrier mismatch error

✦ Open worker.c in the editor by double-clicking on it in Project Explorer

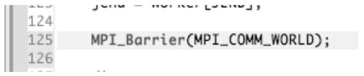
✦ At about line 125, enter a barrier:
- ✦ Type MPI_B
- ✦ Hit Ctl-space
- ✦ Select MPI_Barrier
- ✦ Add communicator arg MPI_COMM_WORLD and closing semicolon



```
120     prv = worker[PREV];
121     nxt = worker[NEXT];
122     jstart = worker[JSTART];
123     jend = worker[JEND];
124
125     MPI_B
126     /*
127     recei
128     */
129
130     for (
131        M
132
```

MPI_Barrier(MPI_Comm) int
MPI_Bcast(void*, int, MPI_Datatype, int, MPI_
MPI_Bsend(void*, int, MPI_Datatype, int, int,
MPI_Bsend_init(void*, int, MPI_Datatype, int,
MPI_Buffer_attach(void*, int) int
MPI_Buffer_detach(void*, int *) int

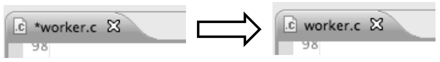
Blocks each task until



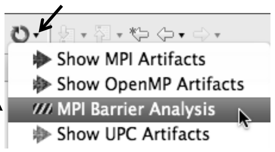
```
124
125     MPI_Barrier(MPI_COMM_WORLD);
126
```

Barrier Errors (2)

- ✦ Save the file
  - ✦ Ctl-S (Mac Command-S) or File > Save
  - ✦ Tab should lose asterisk indicating file saved

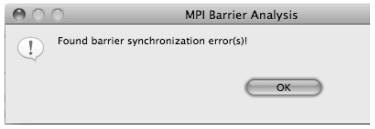


- ✦ Run barrier analysis on shallow project again
  - ✦ Select shallow project in Project Explorer first

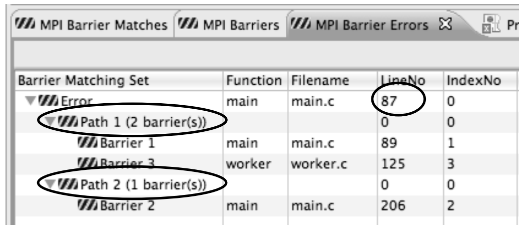


Barrier Errors (3)

- ✦ Barrier Error is found
- ✦ Hit OK to dismiss dialog



- ✦ Code diverges on line 87
  - ✦ One path has 2 barriers, other has 1

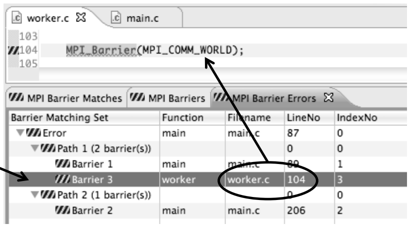


Barrier Matching Set	Function	Filename	LineNo	IndexNo
Error	main	main.c	87	0
Path 1 (2 barrier(s))			0	0
Barrier 1	main	main.c	89	1
Barrier 3	worker	worker.c	125	3
Path 2 (1 barrier(s))			0	0
Barrier 2	main	main.c	206	2

Double-click on a row in Barrier Errors view to find the line it references in the code

Fix Barrier Error

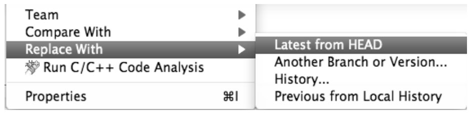
- ✦ Fix the Barrier Error before continuing
- ✦ Double-click on the barrier in worker.c to quickly navigate to it



```
103
104     MPI_Barrier(MPI_COMM_WORLD);
105
```

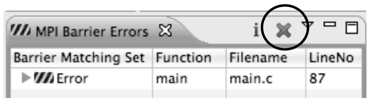
Barrier Matching Set	Function	Filename	LineNo	IndexNo
Error	main	main.c	87	0
Path 1 (2 barrier(s))			0	0
Barrier 1	main	main.c	89	1
Barrier 3	worker	worker.c	104	3
Path 2 (1 barrier(s))			0	0
Barrier 2	main	main.c	206	2

- ✦ Remove the line and save the file
  -or-
  Right mouse on worker.c in Project Explorer and do **Replace With > Latest from HEAD**



Remove Barrier Markers

- ✦ Run Barrier Analysis again to remove the error
  - and/or -
- ✦ Remove the Barrier Markers via the “X” in one of the MPI Barrier views



Barrier Matching Set	Function	Filename	LineNo
Error	main	main.c	87

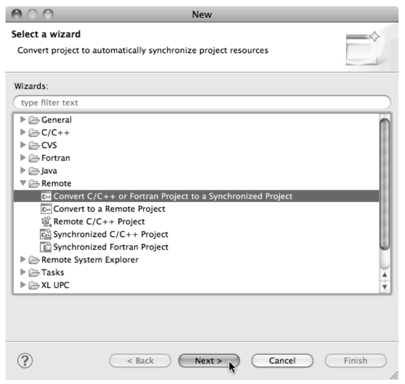
Synchronizing the Project

Synchronizing the Project

- Because we will be running on a remote system, we must also build on that system
- Source files must be available to build
- We will use a synchronized project to do this
  - Only needs to be done once for each project
  - A synchronized project could have been created initially
- Files are synchronized automatically when they are saved
- A full synchronize is also performed prior to a build

Converting To Synchronized

- Select **File>New>Other…**
- Open the Remote folder
- Select **Convert C/C++ or Fortran Project to a Synchronized Project**
- Click **Next>**


New

Select a wizard

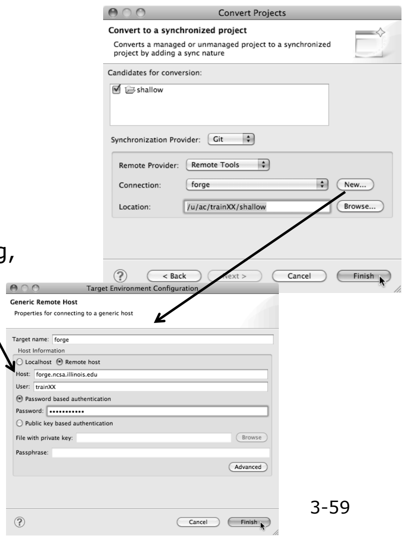
Convert project to automatically synchronize project resources

Wizards:

type filter text
General
C/C++
CVS
Fortran
Java
Remote
Convert C/C++ or Fortran Project to a Synchronized Project
Convert to a Remote Project
Remote C/C++ Project
Synchronized C/C++ Project
Synchronized Fortran Project
Remote System Explorer
Tasks
XL UPC
< Back
Next >
Cancel
Finish

Convert Projects Wizard

- Select checkbox next to shallow
- For **Connection:**, click on **New…** Enter as directed:
  - **Target name**
  - **Host** name of remote system
  - **User** id and **Password**
- Click **Finish** to close it
- Back in the **Convert Projects** dialog,
- Enter a directory name in the **Location** field; select **Browse…**
  - Sample: /u/ac/trainXX/shallow
  - Project files will be copied under this directory
- Click **Finish**


Convert Projects

Convert to a synchronized project

Converts a managed or unmanaged project to a synchronized project by adding a sync nature

Candidates for conversion: [✓] shallow

Synchronization Provider: Git

Remote Provider: Remote Tools

Connection: forge

Location: /u/ac/trainXX/shallow

Target Environment Configuration

Generic Remote Host

Properties for connecting to a generic host

Target name: forge

Host Information

Localhost [✓ Remote host]

Host: forge.ncsa.illinois.edu

User: trainXX

[✓ Password based authentication]

Password: ••••••••••••

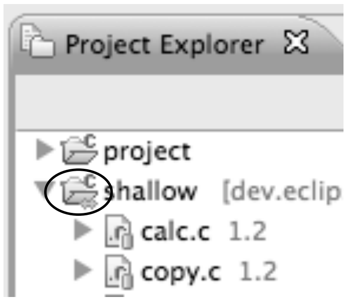
Public key based authentication

File with private key:

Passphrase:

Synchronized Project

- Back in the Project Explorer, decorator on project icon indicates synchronized project
- Double-+ icon

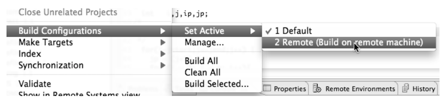


- Before sync

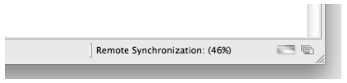
- After sync

Set Active Build Configuration

- The Active build configuration determines which system will be used for both synchronizing and building
- Right-click on the project and select **Build Configurations>Set Active>Remote (Build on remote machine)**



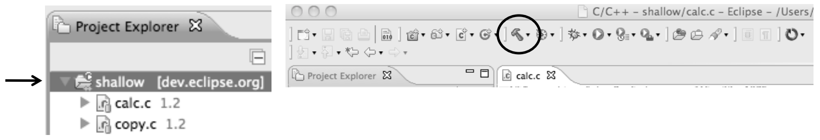
- The project should synchronize immediately



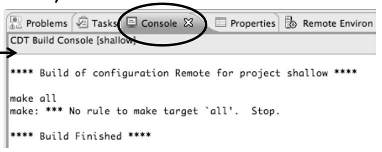
Building the Project

Building the Project

- Select the project in Project Explorer, click on the hammer button to run the build


Project Explorer
shallow [dev.eclipse.org]
calc.c 1.2
copy.c 1.2
C/C++ – shallow/calc.c – Eclipse – /Users/
Project Explorer
calc.c

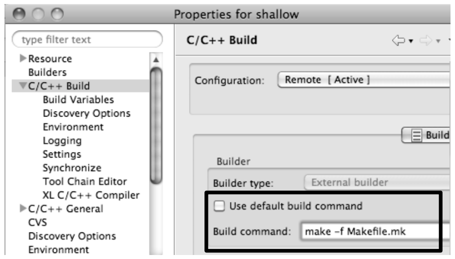
- By default, the Build Configuration assumes there is a Makefile (or makefile) for the project
- In this case, there is no Makefile, so the build will fail. See Console:
  - We ll see how to change it if the build command is different from 'make –f Makefile'



```
**** Build of configuration Remote for project shallow ****

make all
make: *** No rule to make target `all'.  Stop.

**** Build Finished ****
```

Fixing the Build Command: Editing Project Properties

- The build command is specified in the project properties
- Open the properties by right-clicking on shallow and selecting **Properties** (bottom of the context menu list)
- Click on **C/C++ Build**
- Uncheck **Use default build command**
- Enter make –f Makefile.mk in the **Build Command** field
- Click **OK** to close project properties dialog



Configuration: Remote [Active]

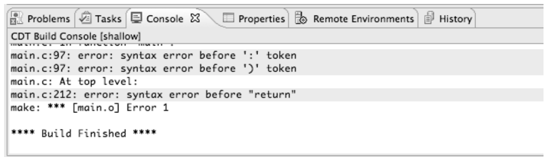
Builder type: External builder

Use default build command

Build command: make –f Makefile.mk

Re-Building the Project

- Click on the button again to run the build
- Build output will be shown in the **Console** view



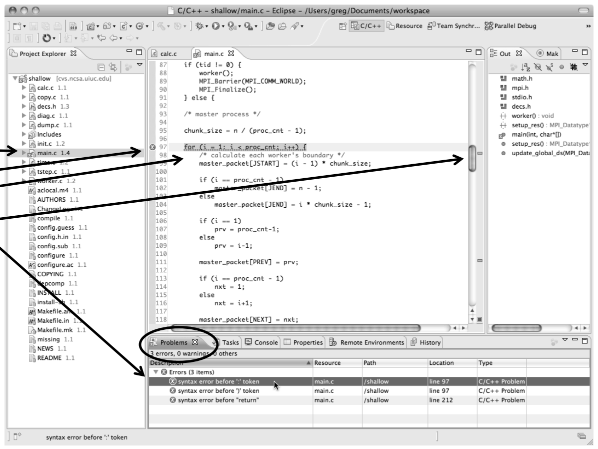
CDT Build Console [shallow]

```
main.c:97: error: syntax error before ':' token
main.c:97: error: syntax error before ')' token
main.c: At top level:
main.c:212: error: syntax error before "return"
make: *** [main.o] Error 1

**** Build Finished ****
```

- Exact output depends on your compiler

Build Problems

- Build problems will be shown in a variety of ways
  - Marker on file
  - Marker on editor line
  - Line is highlighted
  - Marker on overview ruler
  - Listed in the **Problems view**
- Double-click on line in **Problems view** to go to location of error in the editor



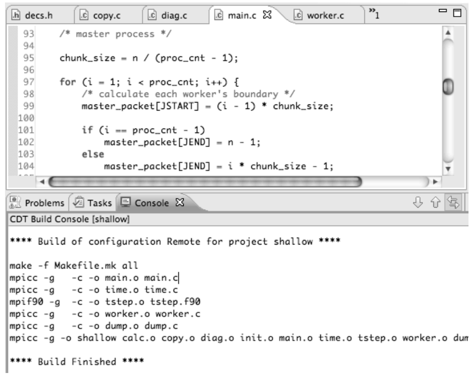
3 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (3 items)				
syntax error before ':' token	main.c	/shallow	line 97	C/C++ Problem
syntax error before ')' token	main.c	/shallow	line 97	C/C++ Problem
syntax error before "return"	main.c	/shallow	line 212	C/C++ Problem

syntax error before ':' token

Fix Build Problems

- Fix errors by changing ':' to ';' on line 97
- Save the file
- Rebuild by pressing build button
- Error markers have been removed
- Check console for correct build output



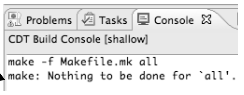
```
 93        /* master process */
 94
 95        chunk_size = n / (proc_cnt - 1);
 96
 97        for (i = 1; i < proc_cnt; i++) {
 98            /* calculate each worker's boundary */
 99            master_packet[JSTART] = (i - 1) * chunk_size;
100
101            if (i == proc_cnt - 1)
102                master_packet[JEND] = n - 1;
103            else
104                master_packet[JEND] = i * chunk_size - 1;
```

CDT Build Console [shallow]

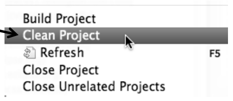
```
**** Build of configuration Remote for project shallow ****

make -f Makefile.mk all
mpicc -g   -c -o main.o main.c
mpicc -g   -c -o time.o time.c
mpif90 -g  -c -o tstep.o tstep.f90
mpicc -g   -c -o worker.o worker.c
mpicc -g   -c -o dump.o dump.c
mpicc -g -o shallow calc.o copy.o diag.o init.o main.o time.o tstep.o worker.o dum

**** Build Finished ****
```

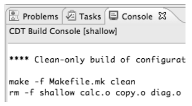
Forcing a Rebuild

- ✦ If no changes have been made, `make` doesn't think a build is needed
- ✦ In Project Explorer, Rightmouse on project, select **Clean Project**
- ✦ See console
- ✦ Then rebuild



```
CDT Build Console [shallow]
make -f Makefile.mk all
make: Nothing to be done for `all'.
```





```
CDT Build Console [shallow]

**** Clean-only build of configurat

make -f Makefile.mk clean
rm -f shallow calc.o copy.o diag.o
```

Running the Program

Resource Managers

Running the Program

- ✦ Creating a resource manager
- ✦ Starting the resource manager
- ✦ Creating a run configuration
- ✦ Running (launching) the application
- ✦ Viewing the application run

Do this once

Much of the following setup is configuration that you only need to do once: This icon will remind you.

Resource Managers

- ✦ PTP uses the term "resource manager" to refer to any subsystem that controls the resources required for launching a parallel job.
- ✦ Examples:
  - ✦ Batch scheduler (e.g. LoadLeveler, PBS, SLURM)
  - ✦ Interactive execution (e.g. Open MPI, MPICH2, etc.)
- ✦ Each resource manager controls one target system
- ✦ Resource Managers can be local or remote

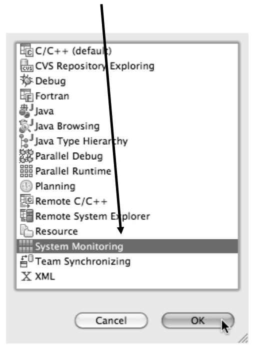
Monitoring/Runtime Perspectives

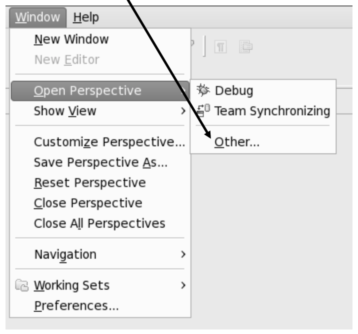
- Parallel Runtime Perspective
  - Used for legacy PTP Resource Managers
- System Monitoring Perspective
  - Used for newer Configurable Resource Managers (since PTP 5.0)
- Which one is used?

Resource Manager	System Monitoring	Parallel Runtime
IBM LoadLeveler		✔
IBM Parallel Env		✔
MPICH2		✔
Open MPI		✔
PBS-Batch-Generic	✔	
PBS-Batch-Interactive	✔	
Remote Launch		✔
SLURM		✔

Preparing to Launch

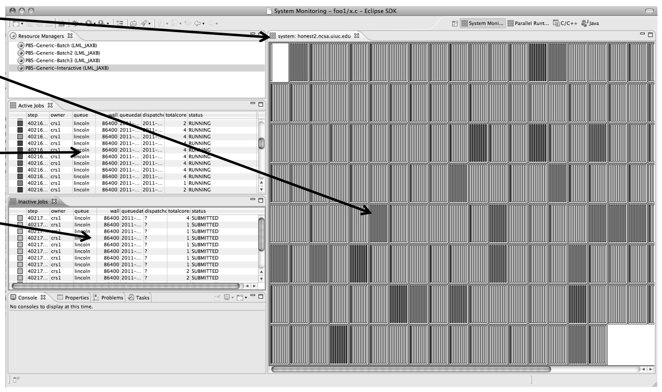
- Setting up a resource manager is done in the System Monitoring perspective
  - (For PTP 5.0, this applies to PBS)
- Select **Window>Open Perspective>Other…**
- Choose **System Monitoring** and click **OK**





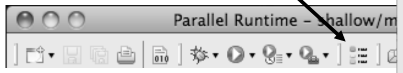
System Monitoring Perspective

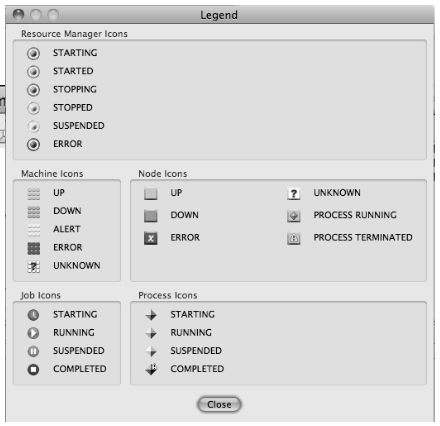
- System view
- Jobs running on system
- Active jobs
- Inactive jobs



About PTP Icons

- Open using legend icon in toolbar





Legend

Resource Manager Icons

- STARTING
- STARTED
- STOPPING
- STOPPED
- SUSPENDED
- ERROR

Machine Icons

- UP
- DOWN
- ALERT
- ERROR
- UNKNOWN

Node Icons

- UP
- DOWN
- ERROR
- UNKNOWN
- PROCESS RUNNING
- PROCESS TERMINATED

Job Icons

- STARTING
- RUNNING
- SUSPENDED
- COMPLETED

Process Icons

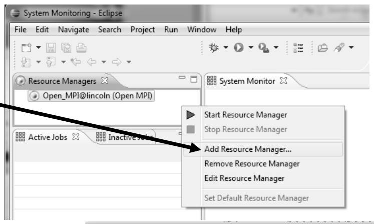
- STARTING
- RUNNING
- SUSPENDED
- COMPLETED

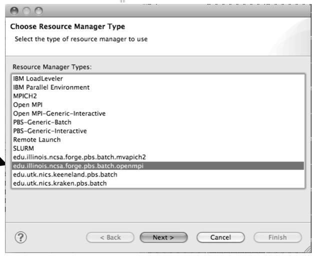
Close

Configuring Job Scheduler

Do this once

- Right-click in Resource Managers view and select **Add Resource Manager**
- Choose Resource Manager Type: **edu.illinois.ncsa.forge .pbs.batch.openmpi**
- Select **Next>**





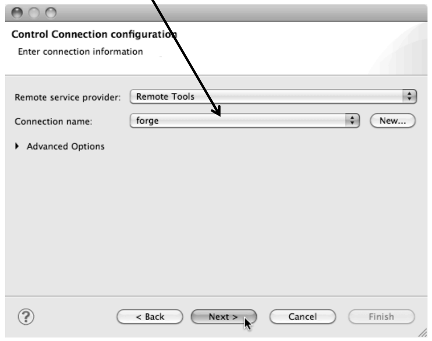
Choose Resource Manager Type
Select the type of resource manager to use

Resource Manager Types:
- IBM LoadLeveler
- IBM Parallel Environment
- MPICH2
- Open MPI
- Open MPI-Generic-Interactive
- PBS-Generic-Batch
- PBS-Generic-Interactive
- Remote Launch
- SLURM
- edu.illinois.ncsa.forge.pbs.batch.mvapich2
- edu.illinois.ncsa.forge.pbs.batch.openmpi
- edu.utk.nics.keeneland.pbs.batch
- edu.utk.nics.kraken.pbs.batch

Configure Control Connection

Do this once

- Choose **Remote Tools** for **Remote service provider**
- Choose the remote connection you made previously
- Click **Next>**



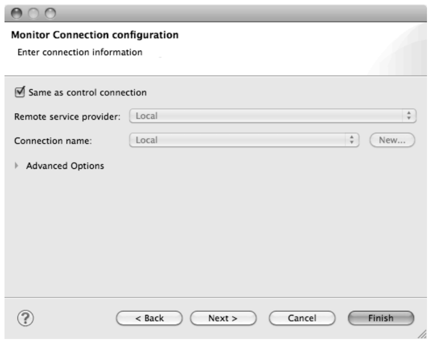
Control Connection configuration
Enter connection information

Remote service provider: Remote Tools
Connection name: forge

Configure Monitor Connection

- Keep default Monitor Connection (same as Control Connection), click **Next**

Do this once



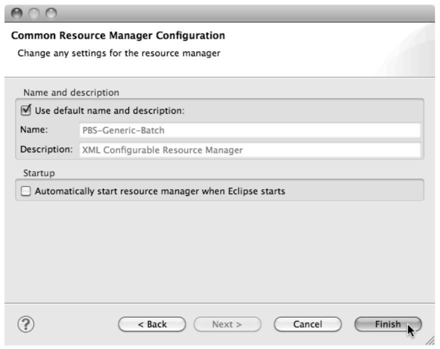
Monitor Connection configuration
Enter connection information

☑ Same as control connection
Remote service provider: Local
Connection name: Local

Common Configuration

- Keep default name
- Can automatically start Resource Manager (leave unselected today)
- Click **Finish**

Do this once



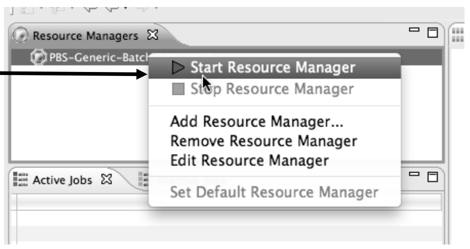
Common Resource Manager Configuration
Change any settings for the resource manager

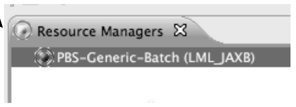
Name and description
☑ Use default name and description:
Name: PBS-Generic-Batch
Description: XML Configurable Resource Manager

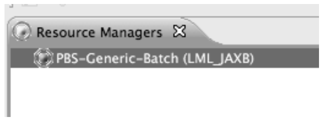
Startup
☐ Automatically start resource manager when Eclipse starts

Starting the Resource Manager

- ✦ Right click on new resource manager and select **Start resource manager**
- ✦ If everything is ok, you should see the resource manager change to green
- ✦ If something goes wrong, it will change to red



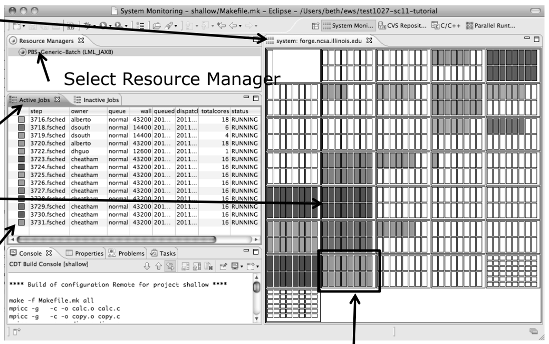




System Monitoring

forge.ncsa.illinois.edu

- ✦ **System** view, with abstraction of nodes for selected Resource Manager
- ✦ Active and inactive jobs
- ✦ Hover over node in **System** view to see job running on node in **Active Jobs** view
- ✦ Hold mouse button down on a job in **Active Jobs** view to see where it is running in **System** view


System Monitoring – shallow/Makefile.mk – Eclipse – /Users/beth/ews/test1027-sc11-tutorial
Resource Managers
PBS-Generic-Batch (LML_JAXB)
Select Resource Manager
system: forge.ncsa.illinois.edu
Active Jobs
Inactive Jobs
Console
Properties
Problems
Tasks
CDT Build Console [shallow]

	step	owner	queue	wall	queued	dispatcl	totalcores	status
▨	3716.fsched	alberto	normal	43200	201...	2011...	18	RUNNING
▨	3718.fsched	dsouth	normal	14400	201...	2011...	6	RUNNING
▨	3719.fsched	dsouth	normal	14400	201...	2011...	4	RUNNING
▨	3720.fsched	alberto	normal	43200	201...	2011...	18	RUNNING
▨	3722.fsched	dhguo	normal	12600	201...	2011...	1	RUNNING
▨	3723.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3724.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3725.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3726.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3727.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3728.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3729.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3730.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING
▨	3731.fsched	cheatham	normal	43200	201...	2011...	16	RUNNING

```
**** Build of configuration Remote for project shallow ****

make -f Makefile.mk all
mpicc -g   -c -o calc.o calc.c
mpicc -g   -c -o copy.o copy.c
```

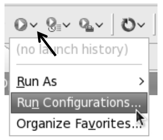
One node with 16 cores

Running the Program

(Launching a Job)

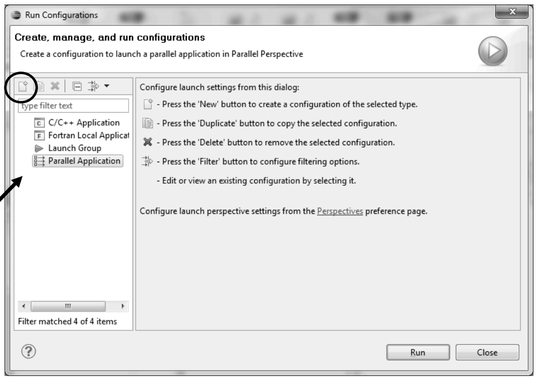
Create a Run Configuration

Do this once



- ✦ Open the run configuration dialog **Run Configurations…**
- ✦ Select **Parallel Application**
- ✦ Select the **New** button

Or, just double-click on **Parallel Application** to create a new one



Run Configurations

Create, manage, and run configurations

Create a configuration to launch a parallel application in Parallel Perspective

type filter text

- C/C++ Application
- Fortran Local Applicat
- Launch Group
- Parallel Application

Filter matched 4 of 4 items

Configure launch settings from this dialog:

- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the Perspectives preference page.

Run   Close

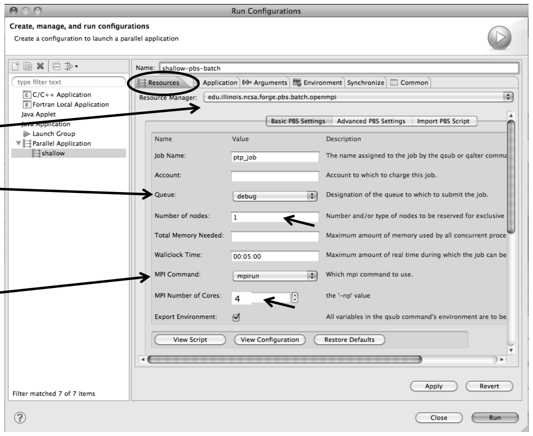
Depending on which flavor of Eclipse you installed, you might have more choices in Application types

Note: we sometimes interchange the terms "Run Configuration" and "Launch Configuration"

Complete the Resources Tab

Do this once

- Enter a name for this run configuration, e.g. “shallow-pbs-batch”
- In **Resources** tab, select the PBS resource manager you just created (edu.illinois.ncsa.forge….)
- Select the destination queue – **debug**
- The **MPI Command** field allows this job to be run as an MPI job
  - Choose **mpirun**
- Enter the resources needed to run this job
  - Use 1 nodes, 4 cores (MPI tasks)



Create, manage, and run configurations
Create a configuration to launch a parallel application

type filter text
- C/C++ Application
- Fortran Local Application
- Java Applet
- Java Application
- Launch Group
- Parallel Application
  - shallow

Filter matched 7 of 7 items

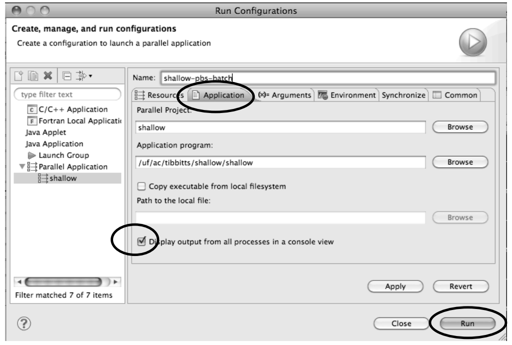
Name: shallow-pbs-batch
Resource Manager: edu.illinois.ncsa.forge.pbs.batch.openmpi

Name	Value	Description
Job Name:	ptp_job	The name assigned to the job by the qsub or qalter comma
Account:		Account to which to charge this job.
Queue:	debug	Designation of the queue to which to submit the job.
Number of nodes:	1	Number and/or type of nodes to be reserved for exclusive
Total Memory Needed:		Maximum amount of memory used by all concurrent proce
Wallclock Time:	00:05:00	Maximum amount of real time during which the job can be
MPI Command:	mpirun	Which mpi command to use.
MPI Number of Cores:	4	the '-np' value
Export Environment:	✓	All variables in the qsub command's environment are to be

Complete the Application Tab

Do this once

- Select the **Application** tab
- Choose the **Application program** by clicking the **Browse** button and locating the executable on the remote machine
  - Use the same “shallow” executable
- Select **Display output from all processes in a console view**
- Click **Run** to submit the application to the job scheduler



Create, manage, and run configurations
Create a configuration to launch a parallel application

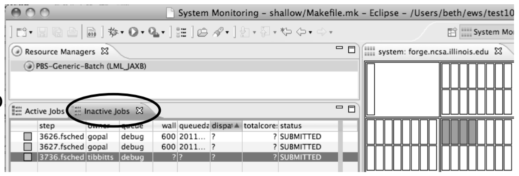
type filter text
- C/C++ Application
- Fortran Local Applicati
- Java Applet
- Java Application
- Launch Group
- Parallel Application
  - shallow

Filter matched 7 of 7 items

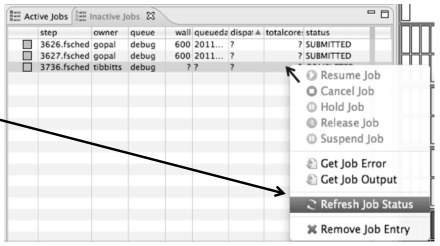
Name: shallow-pbs-batch
Parallel Project: shallow
Application program: /uf/ac/tibbitts/shallow/shallow
☐ Copy executable from local filesystem
Path to the local file:
☑ Display output from all processes in a console view

Job Monitoring

- Job initially appears in “Inactive Jobs”, then in “Active Jobs”, then returns to Inactive on completion
- This short-running program may not run long enough to appear in “Active Jobs”
- Status refreshes automatically every 60 sec Or force refresh with menu
- After status = COMPLETED, Can view output or error by right clicking on job, selecting appropriate output


System Monitoring – shallow/Makefile.mk – Eclipse – /Users/beth/ews/test10
System Mo
Resource Managers
PBS-Generic-Batch (LML_JAXB)
system: forge.ncsa.illinois.edu
Active Jobs
Inactive Jobs

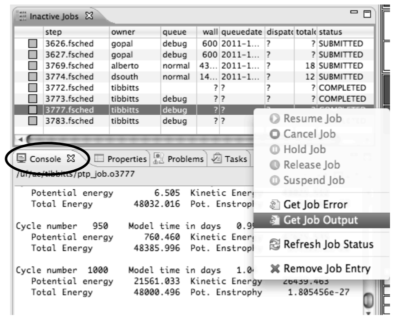
	step	owner	queue	wall	queueda	dispa	totalcore	status
☐	3626.fsched	gopal	debug	600	2011…	?	?	SUBMITTED
☐	3627.fsched	gopal	debug	600	2011…	?	?	SUBMITTED
☐	3736.fsched	tibbitts	debug	?	?	?	?	SUBMITTED



	step	owner	queue	wall	queueda	dispa	totalcore	status
☐	3626.fsched	gopal	debug	600	2011…	?	?	SUBMITTED
☐	3627.fsched	gopal	debug	600	2011…	?	?	SUBMITTED
☐	3736.fsched	tibbitts	debug	?	?	?		

Job Output

- After status = COMPLETED, Can view output or error by right clicking on job, selecting appropriate output
- Output/Error info shows in Console View



	step	owner	queue	wall	queuedate	dispatc	totalc	status
☐	3626.fsched	gopal	debug	600	2011-1…	?	?	SUBMITTED
☐	3627.fsched	gopal	debug	600	2011-1…	?	?	SUBMITTED
☐	3769.fsched	alberto	normal	43…	2011-1…	?	18	SUBMITTED
☐	3774.fsched	dsouth	normal	14…	2011-1…	?	12	SUBMITTED
☐	3772.fsched	tibbitts		?	?	?	?	COMPLETED
☐	3773.fsched	tibbitts	debug	?	?	?	?	COMPLETED
☐	3777.fsched	tibbitts	debug	?	?			
☐	3783.fsched	tibbitts	debug	?	?			

```
/uf/ac/tibbitts/ptp_job.o3777
   Potential energy        6.505   Kinetic Ener
   Total Energy        48032.016   Pot. Enstropl

Cycle number   950    Model time in days    0.9
   Potential energy      760.460   Kinetic Ener
   Total Energy        48385.996   Pot. Enstropl

Cycle number  1000    Model time in days    1.0
   Potential energy    21561.033   Kinetic Energy      26439.463
   Total Energy        48000.496   Pot. Enstrophy    1.805456e-27
```

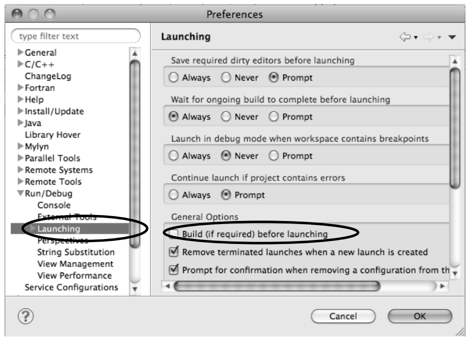
Building before Run

Do this once

- ✦ If projects build prior to launch, you can turn it off.
  - ✦ Go into **Preferences>Run/ Debug** and click on **Launching**.
  - ✦ Uncheck "**Build (if required) before launching**"

  - ✦ Should be set by default now

To bring up **Preferences** dialog, use Window>Preferences or Mac: Eclipse>Preferences



Launching

Save required dirty editors before launching: Always Never [✓ Prompt]

Wait for ongoing build to complete before launching: [✓ Always] Never Prompt

Launch in debug mode when workspace contains breakpoints: Always [✓ Never] Prompt

Continue launch if project contains errors: Always [✓ Prompt]

General Options

- ☐ Build (if required) before launching
- ✓ Remove terminated launches when a new launch is created
- ✓ Prompt for confirmation when removing a configuration from th

Exercise

- ✦ Start with your 'shallow' project
- ✦ Create and start Resource Manager
- ✦ Build; Run shallow
- ✦ See results
- ✦ Change something
  - ✦ Change m and n in decs.h
- ✦ Rebuild and re-run

Advanced Features

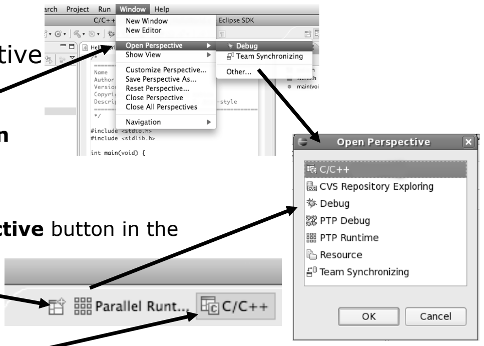
Searching
Fortran
Refactoring

Searching

Switching Perspectives

✦ Switch to C/C++ Perspective one of three ways:

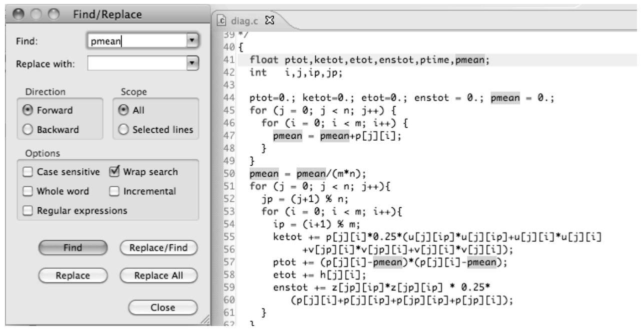
1. Choose the **Window>Open Perspective** menu option Then choose **Other…**
2. Click on the **Open Perspective** button in the upper right corner of screen (hover over it to see names)
3. Click on a perspective shortcut button



Find/Replace within Editor

✦ Simple Find within editor buffer

✦ Ctrl-F (Mac: Command-F)


Find/Replace

Find: pmean

Replace with:

Direction: [✓ Forward] Backward

Scope: [✓ All] Selected lines

Options: Case sensitive; [✓ Wrap search]; Whole word; Incremental; Regular expressions

```
*/
{
  float ptot,ketot,etot,enstot,ptime,pmean;
  int   i,j,ip,jp;

  ptot=0.; ketot=0.; etot=0.; enstot = 0.; pmean = 0.;
  for (j = 0; j < n; j++) {
    for (i = 0; i < m; i++) {
      pmean = pmean+p[j][i];
    }
  }
  pmean = pmean/(m*n);
  for (j = 0; j < n; j++){
    jp = (j+1) % n;
    for (i = 0; i < m; i++){
      ip = (i+1) % m;
      ketot += p[j][i]*0.25*(u[j][ip]*u[j][ip]+u[j][i]*u[j][i]
           +v[jp][i]*v[jp][i]+v[j][i]*v[j][i]);
      ptot += (p[j][i]-pmean)*(p[j][i]-pmean);
      etot += h[j][i];
      enstot += z[jp][ip]*z[jp][ip] * 0.25*
         (p[j][i]+p[j][ip]+p[jp][ip]+p[jp][i]);
    }
```

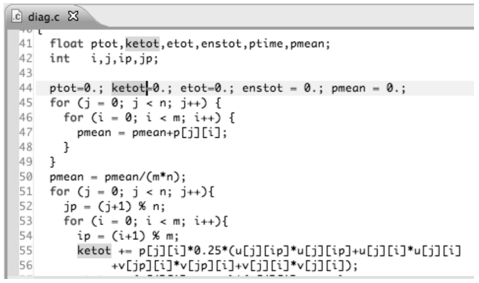
Mark Occurrences

(C/C++ Only)

✦ Double-click on a variable in the CDT editor

✦ All occurrences in the source file are highlighted to make locating the variable easier

✦ Alt-shift-O to turn off (Mac: Alt-Cmd-O)



```
  float ptot,ketot,etot,enstot,ptime,pmean;
  int   i,j,ip,jp;

  ptot=0.; ketot=0.; etot=0.; enstot = 0.; pmean = 0.;
  for (j = 0; j < n; j++) {
    for (i = 0; i < m; i++) {
      pmean = pmean+p[j][i];
    }
  }
  pmean = pmean/(m*n);
  for (j = 0; j < n; j++){
    jp = (j+1) % n;
    for (i = 0; i < m; i++){
      ip = (i+1) % m;
      ketot += p[j][i]*0.25*(u[j][ip]*u[j][ip]+u[j][i]*u[j][i]
           +v[jp][i]*v[jp][i]+v[j][i]*v[j][i]);
```

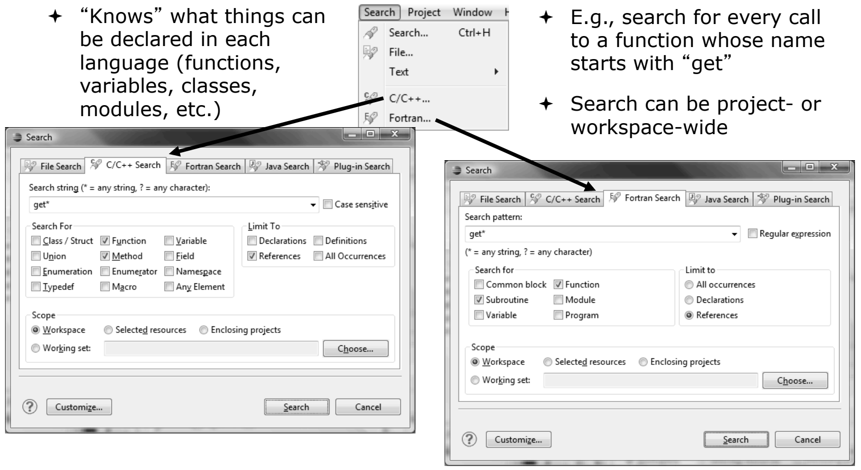
Language-Based Searching

(C/C++ and Fortran)

✦ "Knows" what things can be declared in each language (functions, variables, classes, modules, etc.)

✦ E.g., search for every call to a function whose name starts with "get"

✦ Search can be project- or workspace-wide



C/C++ Search: Search string (* = any string, ? = any character): get*; Case sensitive

Search For: Class / Struct; [✓ Function]; Variable; Union; [✓ Method]; Field; Enumeration; Enumerator; Namespace; Typedef; Macro; Any Element

Limit To: Declarations; Definitions; [✓ References]; All Occurrences

Scope: [✓ Workspace] Selected resources Enclosing projects Working set:

Fortran Search: Search pattern: get*; Regular expression; (* = any string, ? = any character)

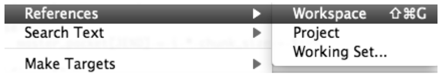
Search for: Common block; [✓ Function]; [✓ Subroutine]; Module; Variable; Program

Limit to: All occurrences Declarations [✓ References]

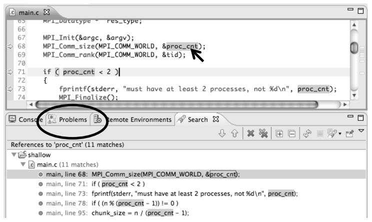
Scope: [✓ Workspace] Selected resources Enclosing projects Working set:

Find References
(C/C++ and Fortran)

- Finds all of the places where a variable, function, etc., is used
  - Right-click on an identifier in the editor
  - Click **References ▸** or **References ▸**



  - **Search** view shows matches

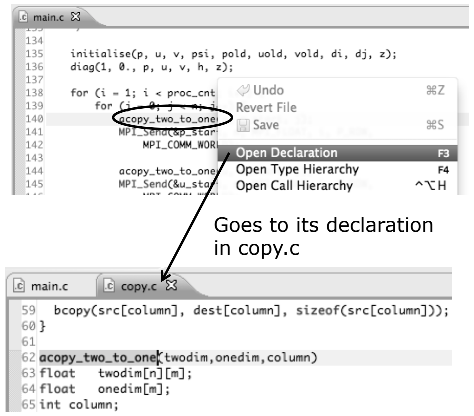


```
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &proc_cnt);
MPI_Comm_rank(MPI_COMM_WORLD, &tid);

if ( proc_cnt < 2 )
{
    fprintf(stderr, "must have at least 2 processes, not %d\n", proc_cnt);
    MPI_Finalize();
```

```
References to 'proc_cnt' (11 matches)
shallow
  main.c (11 matches)
    main, line 68: MPI_Comm_size(MPI_COMM_WORLD, &proc_cnt);
    main, line 71: if ( proc_cnt < 2 )
    main, line 73: fprintf(stderr, "must have at least 2 processes, not %d\n", proc_cnt);
    main, line 78: if ( (n % (proc_cnt - 1)) != 0 )
    main, line 95: chunk_size = n / (proc_cnt - 1);
```

Open Declaration
(C/C++ and Fortran)

- Jumps to the declaration of a variable, function, etc., even if it's in a different file
- Left-click to select identifier
- Right-click on identifier
- Click **Open Declaration**
- C/C++ only:
  Can also Ctrl-click (Mac: Cmd-click) on an identifier to "hyperlink" to its declaration



```
initialise(p, u, v, psi, pold, uold, vold, di, dj, z);
diag(1, 0., p, u, v, h, z);

for (i = 1; i < proc_cnt
    for (j = 0; j < n; j
        acopy_two_to_one
        MPI_Send(&p_star
            MPI_COMM_WOR

        acopy_two_to_one
        MPI_Send(&u_sta
```

```
  bcopy(src[column], dest[column], sizeof(src[column]));
}

acopy_two_to_one(twodim,onedim,column)
float   twodim[n][m];
float   onedim[m];
int column;
```

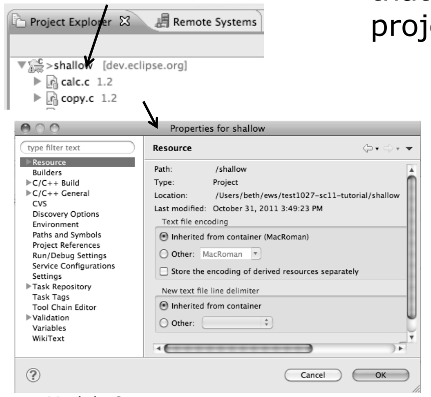
Search – Try It!

1. Find every call to   in Shallow.
2. In worker.c, on line 42, there is a declaration `float p[n][m]`.
   a) What is `m` (local? global? function parameter?)
   b) Where is `m` defined?
   c) How many times is `m` used in the project?
3. Find every function whose name contains the word `time`

Fortran Specifics

Project Properties

- Right-click Project
- Select **Properties**…



Resource

Path: /shallow
Type: Project
Location: /Users/beth/ews/test1027-sc11-tutorial/shallow
Last modified: October 31, 2011 3:49:23 PM

Text file encoding
- [✓ Inherited from container (MacRoman)]
- Other: MacRoman
- ☐ Store the encoding of derived resources separately

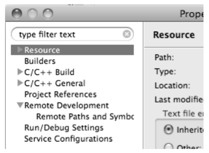
New text file line delimiter
- [✓ Inherited from container]
- Other:

- *Project properties* are settings that can be changed for each project
  - Contrast with *workspace preferences,* which are the same regardless of what project is being edited
    - e.g., editor colors
    - Set in **Window▶ Preferences** (on Mac, **Eclipse▶ Preferences**)
    - Careful! Dialog is very similar

Converting to a Fortran Project

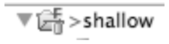
- Are there categories labeled **Fortran General** and **Fortran Build** in the project properties?

No Fortran categories



Do this once

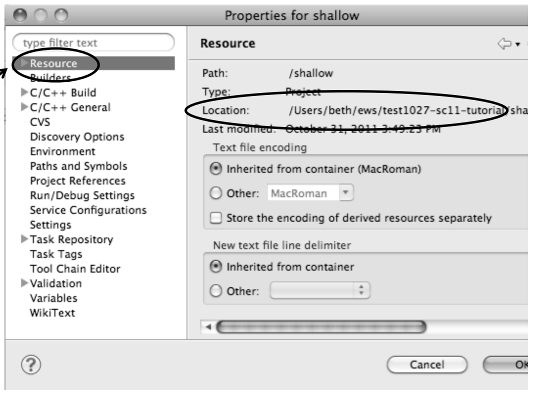
- If not, the project is not a Fortran Project
  - Switch to the Fortran Perspective
  - In the Project Explorer view, right-click on the project, and click **Convert to Fortran Project**
  - Don't worry; it's still a C/C++ project, too



- *Every* Fortran project is also a C/C++ Project

Project Location

- How to tell where a project resides?
- In the project properties dialog, select the **Resource** category



Resource

Path: /shallow
Type: Project
Location: /Users/beth/ews/test1027-sc11-tutorial/sha
Last modified: October 31, 2011 3:49:23 PM

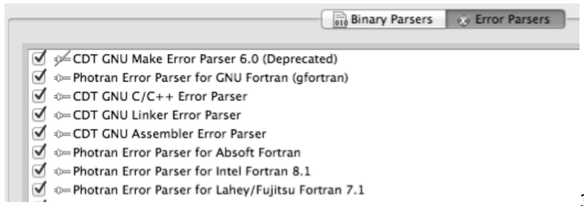
Text file encoding
- [✓ Inherited from container (MacRoman)]
- Other: MacRoman
- ☐ Store the encoding of derived resources separately

New text file line delimiter
- [✓ Inherited from container]
- Other:

Error Parsers

- Are compiler errors not appearing in the Problems view?
  - Make sure the correct *error parser* is enabled
  - In the project properties, navigate to **C++ Build▶Settings** or **Fortran Build▶Settings**
  - Switch to the **Error Parsers** tab
  - Check the error parser(s) for your compiler(s)

Do this once



- ✓ CDT GNU Make Error Parser 6.0 (Deprecated)
- ✓ Photran Error Parser for GNU Fortran (gfortran)
- ✓ CDT GNU C/C++ Error Parser
- ✓ CDT GNU Linker Error Parser
- ✓ CDT GNU Assembler Error Parser
- ✓ Photran Error Parser for Absoft Fortran
- ✓ Photran Error Parser for Intel Fortran 8.1
- ✓ Photran Error Parser for Lahey/Fujitsu Fortran 7.1

Fortran Source Form Settings

- Fortran files are either *free form* or *fixed form;* some Fortran files are *preprocessed* (#define, #ifdef, etc.)
  - Source form determined by filename extension
  - Defaults are similar to most Fortran compilers:

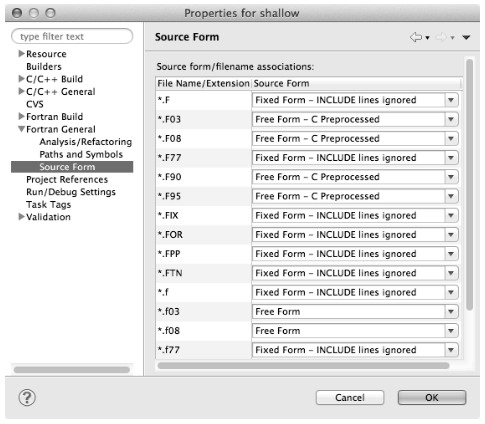
Fixed form:	.f	.fix	.for	.fpp	.ftn	.f77	
Free form:	.f08	.f03	.f95	.f90			< unpreprocessed
	.F08	.F03	.F95	.F90			< preprocessed

- Many features *will not work* if filename extensions are associated with the wrong source form (outline view, content assist, search, refactorings, etc.)

Fortran Source Form Settings

Do this once

- In the project properties, select **Fortran General ▶ Source Form**
- Select source form for each filename extension
- Click **OK**



Source Form

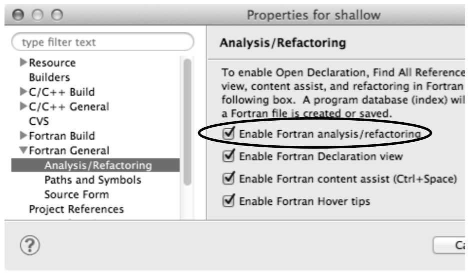
Source form/filename associations:

File Name/Extension	Source Form
*.F	Fixed Form – INCLUDE lines ignored
*.F03	Free Form – C Preprocessed
*.F08	Free Form – C Preprocessed
*.F77	Fixed Form – INCLUDE lines ignored
*.F90	Free Form – C Preprocessed
*.F95	Free Form – C Preprocessed
*.FIX	Fixed Form – INCLUDE lines ignored
*.FOR	Fixed Form – INCLUDE lines ignored
*.FPP	Fixed Form – INCLUDE lines ignored
*.FTN	Fixed Form – INCLUDE lines ignored
*.f	Fixed Form – INCLUDE lines ignored
*.f03	Free Form
*.f08	Free Form
*.f77	Fixed Form – INCLUDE lines ignored

Enabling Fortran Advanced Features

- Some Fortran features are by default
- Must be explicitly enabled
  - In the project properties dialog, select **Fortran General ▶ Analysis/Refactoring**
  - Click **Enable Analysis/ Refactoring**
  - Close and re-open any Fortran editors
- This turns on the "Photran Indexer"
  - Turn it off if it's slow

Do this once



Analysis/Refactoring

To enable Open Declaration, Find All Reference view, content assist, and refactoring in Fortran following box. A program database (index) wil a Fortran file is created or saved.

- ✓ Enable Fortran analysis/refactoring
- ✓ Enable Fortran Declaration view
- ✓ Enable Fortran content assist (Ctrl+Space)
- ✓ Enable Fortran Hover tips

Project Properties – Try It!

1. Convert shallow to a Fortran project
2. Make sure errors from the GNU Fortran compiler will be recognized
3. Make sure *.f90 files are treated as "Free Form" which is unpreprocessed
4. Make sure search and refactoring will work in Fortran

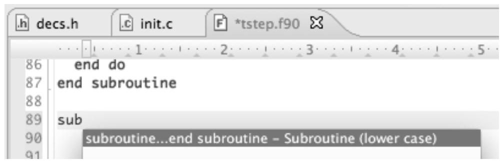
Advanced Editing

Code Templates

Code Templates

(C/C++ and Fortran)

- Auto-complete common code patterns
  - For loops/do loops, if constructs, etc.
  - Also MPI code templates
- Included with content assist proposals (when **Ctrl-Space** is pressed)
  - E.g., after the last line in tstep.f90, type "sub" and press **Ctrl-Space**
  - Press **Enter** to insert the template

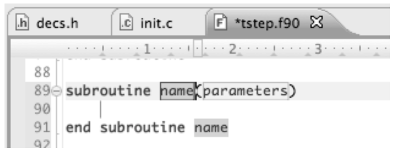


```
  end do
end subroutine

sub
```

Code Templates (2)

(C/C++ and Fortran)

- After pressing enter to insert the code template, completion fields are highlighted



```

subroutine name(parameters)

end subroutine name
```

- Press **Tab** to move between completion fields
- Changing one instance of a field changes all occurrences

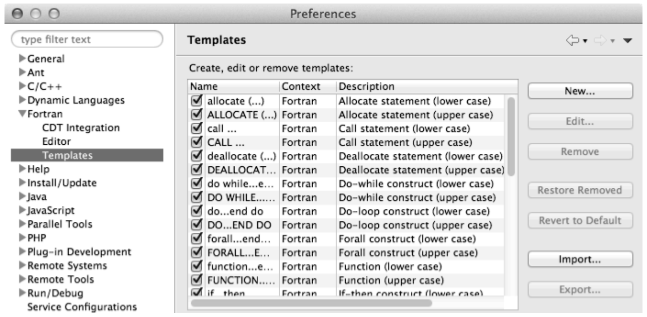
Advanced Editing – Try It!

- Open tstep.f90 and retype the last loop nest
  - Use the code template to complete the do-loops
  - Use content assist to complete variable names

Custom Code Templates

(Fortran)

- ✦ Customize code templates in **Window ▶ Preferences ▶ Fortran ▶ Templates**



Preferences

type filter text

- General
- Ant
- C/C++
- Dynamic Languages
- Fortran
  - CDT Integration
  - Editor
  - Templates
- Help
- Install/Update
- Java
- JavaScript
- Parallel Tools
- PHP
- Plug-in Development
- Remote Systems
- Remote Tools
- Run/Debug
- Service Configurations

Templates

Create, edit or remove templates:

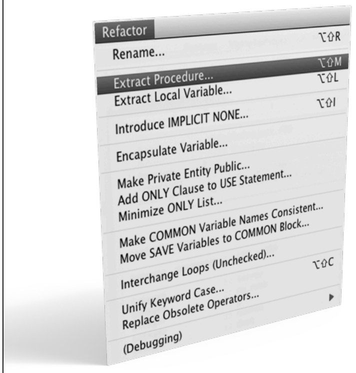
Name	Context	Description
✓ allocate (...)	Fortran	Allocate statement (lower case)
✓ ALLOCATE (...)	Fortran	Allocate statement (upper case)
✓ call ...	Fortran	Call statement (lower case)
✓ CALL ...	Fortran	Call statement (upper case)
✓ deallocate (...)	Fortran	Deallocate statement (lower case)
✓ DEALLOCAT...	Fortran	Deallocate statement (upper case)
✓ do while...e...	Fortran	Do-while construct (lower case)
✓ DO WHILE......	Fortran	Do-while construct (upper case)
✓ do...end do	Fortran	Do-loop construct (lower case)
✓ DO...END DO	Fortran	Do-loop construct (upper case)
✓ forall...end...	Fortran	Forall construct (lower case)
✓ FORALL...E...	Fortran	Forall construct (upper case)
✓ function...e...	Fortran	Function (lower case)
✓ FUNCTION.....	Fortran	Function (upper case)
✓ if...then	Fortran	If-then construct (lower case)

- ✦ Can import/export templates to XML files

Refactoring and Transformation

Refactoring

(making changes to source code that don't affect the behavior of the program)


Refactor
Rename... ⌥⇧R
Extract Procedure... ⌥⇧M
Extract Local Variable... ⌥⇧L
Introduce IMPLICIT NONE... ⌥⇧I
Encapsulate Variable...
Make Private Entity Public...
Add ONLY Clause to USE Statement...
Minimize ONLY List...
Make COMMON Variable Names Consistent...
Move SAVE Variables to COMMON Block...
Interchange Loops (Unchecked)... ⌥⇧C
Unify Keyword Case...
Replace Obsolete Operators...
(Debugging) ▸

- ✦ Refactoring is the research motivation for Photran @ Illinois
  - ✦ Illinois is a leader in refactoring research
  - ✦ "Refactoring" was coined in our group (Opdyke & Johnson, 1990)
  - ✦ We had the first dissertation… (Opdyke, 1992)
  - ✦ …and built the first refactoring tool… (Roberts, Brant, & Johnson, 1997)
  - ✦ …and first supported the C preprocessor (Garrido, 2005)
  - ✦ Photran's agenda: refactorings for HPC, language evolution, refactoring framework
- ✦ Photran 7.0: 31 refactorings

Refactoring Caveats

- ✦ Photran can only refactor free form code that is preprocessed
  - ✦ Determined by Source Form settings (recall from earlier that these are configured in **Project Properties: Fortran General▶Source Form**)

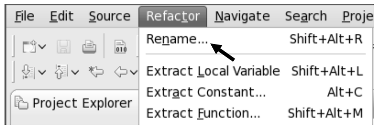
✔	**Free Form, Unpreprocessed:**			.f08	.f03	.f95	.f90
✖	**Free Form, Preprocessed:**			.F08	.F03	.F95	.F90
✖	**Fixed Form:**	.f	.fix	.for	.fpp	.ftn	.f77

- ✦ Refactor menu will be empty if
  - ✦ Refactoring not enabled in project properties (recall from earlier that it is enabled in **Project Properties: Fortran General▶Analysis/Refactoring**)
  - ✦ The file in the active editor is fixed form
  - ✦ The file in the active editor is preprocessed

Rename Refactoring

(also available in Fortran)

- ✦ Changes the name of a variable, function, etc., *including every use*
  (change is semantic, not textual, and can be workspace-wide)
- ✦ Only proceeds if the new name will be legal
  (aware of scoping rules, namespaces, etc.)


File
Edit
Source
Refactor
Navigate
Search
Proje
Rename... Shift+Alt+R
Extract Local Variable Shift+Alt+L
Extract Constant... Alt+C
Extract Function... Shift+Alt+M
Project Explorer

- ✦ Switch to C/C++ Perspective
- ✦ Open a source file
- ✦ In the editor, click on a variable or function name
- ✦ Select menu item **Refactor ▸ Rename**
  - ✦ Or use context menu
- ✦ Enter new name

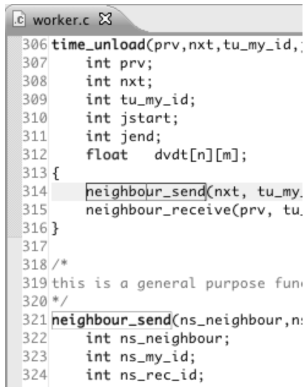
In Java (Murphy-Hill et al., ICSE 2008):

Refactoring	Uses	Percentage
Rename	179,871	74.8%
Extract Local Variable	13,523	5.6%
Move	13,208	5.5%
Extract Method	10,581	4.4%
Change Method Signature	4,764	2.0%
Inline	4,102	1.7%
Extract Constant	3,363	1.4%
(16 Other Refactorings)	10,924	4.5%

Rename in File

(C/C++ Only)

- ✦ Position the caret over an identifier.
- ✦ Press **Ctrl-1** (**Command-1** on Mac).
- ✦ Enter a new name. Changes are propagated within the file as you type.

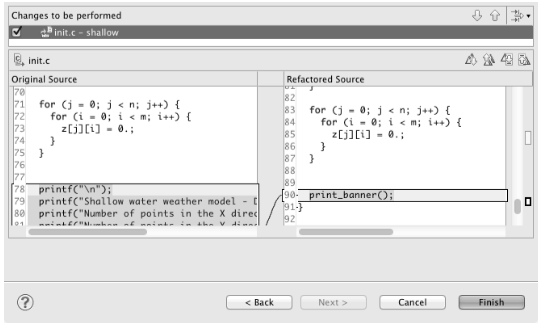

worker.c

```
time_unload(prv,nxt,tu_my_id,
    int prv;
    int nxt;
    int tu_my_id;
    int jstart;
    int jend;
    float   dvdt[n][m];
{
    neighbour_send(nxt, tu_my
    neighbour_receive(prv, tu
}

/*
this is a general purpose fun
*/
neighbour_send(ns_neighbour,n
    int ns_neighbour;
    int ns_my_id;
    int ns_rec_id;
```

Extract Function Refactoring

(also available in Fortran - “Extract Procedure”)

- ✦ Moves statements into a new function, replacing the statements with a call to that function
- ✦ Local variables are passed as arguments



Original Source:

```
  for (j = 0; j < n; j++) {
    for (i = 0; i < m; i++) {
      z[j][i] = 0.;
    }
  }


  printf("\n");
  printf("Shallow water weather model - [
  printf("Number of points in the X direc
```

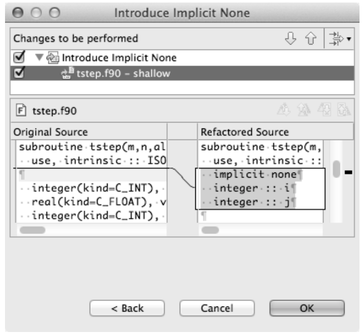
Refactored Source:

```
  for (j = 0; j < n; j++) {
    for (i = 0; i < m; i++) {
      z[j][i] = 0.;
    }
  }


  print_banner();
}
```

- ✦ Select a sequence of statements
- ✦ Select menu item **Refactor ▸ Extract Function…**
- ✦ Enter new name

Introduce IMPLICIT NONE Refactoring

- ✦ Fortran does not require variable declarations
  (by default, names starting with I-N are integer variables; others are reals)
- ✦ This adds an IMPLICIT NONE statement and adds explicit variable declarations for all implicitly declared variables



Original Source:

```
subroutine tstep(m,n,al
  use, intrinsic :: ISO

  integer(kind=C_INT),
  real(kind=C_FLOAT), v
  integer(kind=C_INT),
```

Refactored Source:

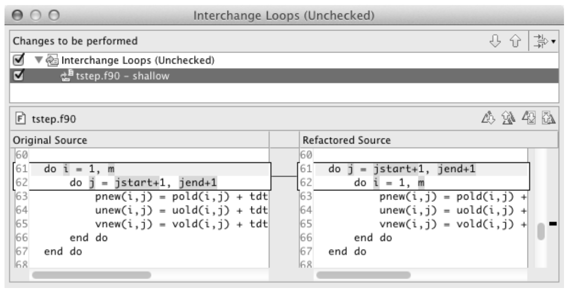
```
subroutine tstep(m,
  use, intrinsic ::
  implicit none
  integer :: i
  integer :: j

```

- ✦ Introduce in a single file by opening the file and selecting **Refactor ▸ Coding Style ▸ Introduce IMPLICIT NONE…**
- ✦ Introduce in multiple files by selecting them in the Project Explorer view, right-clicking on the selection, and choosing **Refactor ▸ Coding Style ▸ Introduce IMPLICIT NONE…**

Loop Transformations

(Fortran only)

- **Interchange Loops** **CAUTION**: No check for behavior preservation
  - Swaps the loop headers in a two-loop nest
  - Select the loop nest, click menu item ▶ ▶ **Interchange Loops (Unchecked)…**



Original Source:

```
  do i = 1, m
      do j = jstart+1, jend+1
            pnew(i,j) = pold(i,j) + tdt
            unew(i,j) = uold(i,j) + tdt
            vnew(i,j) = vold(i,j) + tdt
      end do
  end do
```

Refactored Source:

```
  do j = jstart+1, jend+1
      do i = 1, m
            pnew(i,j) = pold(i,j) +
            unew(i,j) = uold(i,j) +
            vnew(i,j) = vold(i,j) +
      end do
  end do
```

Old version traverses matrices in row-major order

New version traverses in column-major order (better cache performance)

Loop Transformations

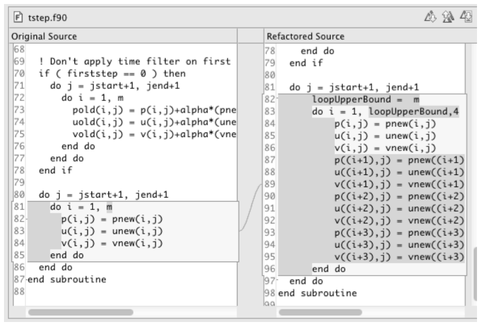
(Fortran only)

- **Unroll Loop**
- Select a loop, click **Refactor▶Do Loop▶Unroll Loop…**

```
do i = 1, 12
  print *, 10*i
end do
```

Unroll 4×

```
do i = 1, 12, 4
  print *, 10*i
  print *, 10*(i+1)
  print *, 10*(i+2)
  print *, 10*(i+3)
end do
```



Original Source:

```
  ! Don't apply time filter on first
  if ( firststep == 0 ) then
    do j = jstart+1, jend+1
      do i = 1, m
        pold(i,j) = p(i,j)+alpha*(pne
        uold(i,j) = u(i,j)+alpha*(une
        vold(i,j) = v(i,j)+alpha*(vne
      end do
    end do
  end if

  do j = jstart+1, jend+1
    do i = 1, m
      p(i,j) = pnew(i,j)
      u(i,j) = unew(i,j)
      v(i,j) = vnew(i,j)
    end do
  end do
end subroutine
```

Refactored Source:

```
    end do
  end if

  do j = jstart+1, jend+1
    loopUpperBound =  m
    do i = 1, loopUpperBound,4
        p(i,j) = pnew(i,j)
        u(i,j) = unew(i,j)
        v(i,j) = vnew(i,j)
        p((i+1),j) = pnew((i+1)
        u((i+1),j) = unew((i+1)
        v((i+1),j) = vnew((i+1)
        p((i+2),j) = pnew((i+2)
        u((i+2),j) = unew((i+2)
        v((i+2),j) = vnew((i+2)
        p((i+3),j) = pnew((i+3)
        u((i+3),j) = unew((i+3)
        v((i+3),j) = vnew((i+3)
    end do
  end do
end subroutine
```

Refactoring & Transformation – Try It!

In tstep.f90…

1. In init.c, extract the `printf` statements at the bottom of the file into a new function called `print_banner`
2. In worker.c, change the spellings of `neighbour_send` and `neighbour_receive` to American English
3. In tstep.f90, make the (Fortran) `tstep` subroutine IMPLICIT NONE

Module 4: Other Tools and Wrap-up

- Objective
  - How to find more information on PTP
  - Learn about other tools related to PTP
  - See PTP upcoming features
- Contents
  - Links to other tools, including performance tools
  - Planned features for new versions of PTP
  - Additional documentation
  - How to get involved

NCSA Blue Waters HPC Workbench

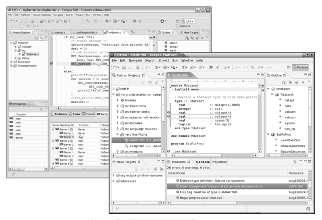
- Tools for NCSA Blue Waters
  - http://www.ncsa.illinois.edu/BlueWaters/
  - Sustained Petaflop system
- Based on Eclipse and PTP
- Includes some related tools
  - Performance tools
  - Workflow tools (https://wiki.ncsa.uiuc.edu/display/MRDPUB/MRD+Public+Space+Home+Page)
- Part of the enhanced computational environment described at:
  http://www.ncsa.illinois.edu/BlueWaters/ece.html

NSF SI2 Workbench for High Performance Computing

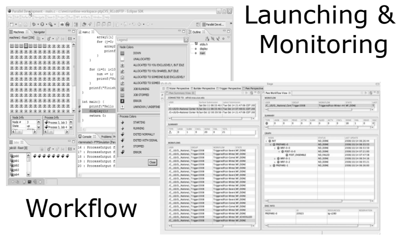
- “
  for HPC Applications”, which is supported by the National Science Foundation under award number OCI 1047956
- Produce a productive and accessible development workbench using Eclipse PTP
- Key Components
  - Determining Requirements, Ensuring Impact
  - Make improvements to Eclipse PTP
  - Engineering Process
  - Metrics
  - Outreach/Training/Education

NCSA HPC Workbench

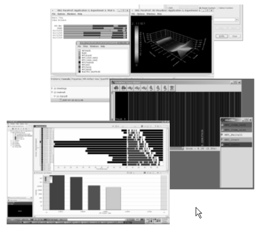
Coding & Analysis (C/C++, Fortran)



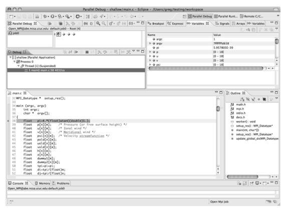
PTP Launching & Monitoring



Workflow



Performance Tuning



Parallel Debugger

Planned PTP Future Work

- Scalability improvements
  - UI to support 1M processes
  - Very large application support
- Usability improvements
  - New wizard to improve setup experience
  - Ability to share configuration information
- Resource Managers
  - More implementations of configurable resource managers
- Synchronized project improvements
  - Conversion wizard
  - Resolving merge conflicts

Useful Eclipse Tools

- Linux Tools (autotools, valgrind, Oprofile, Gprof)
  - http://eclipse.org/linuxtools
- Python
  - http://pydev.org
- Ruby
  - http://www.aptana.com/products/radrails
- Perl
  - http://www.epic-ide.org
- Git
  - http://www.eclipse.org/egit
- VI bindings
  - Vrapper (open source) - http://vrapper.sourceforge.net
  - viPlugin (commercial) - http://www.viplugin.com

Online Information

- Information about PTP
  - Main web site for downloads, documentation, etc.
    - http://eclipse.org/ptp
  - Wiki for designs, planning, meetings, etc.
    - http://wiki.eclipse.org/PTP
  - Articles and other documents
    - http://wiki.eclipse.org/PTP/articles
- Information about Photran
  - Main web site for downloads, documentation, etc.
    - http://eclipse.org/photran
  - User's manuals
    - http://wiki.eclipse.org/PTP/photran/documentation

Mailing Lists

- PTP Mailing lists
  - Major announcements (new releases, etc.) - low volume
    - http://dev.eclipse.org/mailman/listinfo/ptp-announce
  - User discussion and queries - medium volume
    - http://dev.eclipse.org/mailman/listinfo/ptp-user
  - Developer discussions - high volume
    - http://dev.eclipse.org/mailman/listinfo/ptp-dev
- Photran Mailing lists
  - User discussion and queries
    - http://dev.eclipse.org/mailman/listinfo/photran
  - Developer discussions –
    - Also on ptp-dev list (see above)

Getting Involved

- See http://eclipse.org/ptp
- Read the developer documentation on the wiki
  - http://wiki.eclipse.org/PTP
- Join the mailing lists
- Attend the monthly developer meetings
  - Conf Call Monthly: Second Tuesday, 1:00 pm ET
  - Details on the PTP wiki
- Attend the monthly user meetings
  - Teleconf Monthly: 4th Wednesday, 1:00 pm ET
  - Details on the PTP wiki

PTP will only succeed with your participation!