parallel tools platform

http://eclipse.org/ptp

Developing Scientific Applications Using Eclipse and the Parallel Tools Platform

Greg Watson, IBM
g.watson@computer.org

Beth Tibbitts, IBM
tibbitts@us.ibm.com

Jay Alameda, NCSA
jalameda@ncsa.uiuc.edu

Jeff Overbey, UIUC
overbey2@illinois.edu

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 Blue Waters sustained petascale computing project, which is supported by the National Science Foundation under award number OCI 07-25070.

IEEE
Cluster 2009

New Orleans,
Louisiana

Aug. 31-Sept. 4

Tutorial Outline

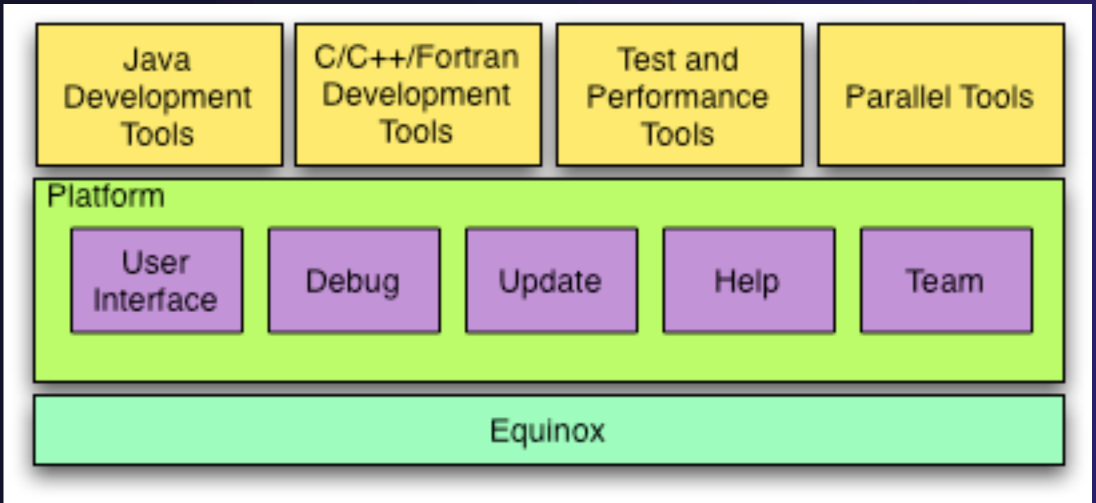
Time (Tentative!)	Module	Outcomes	Presenter
8:30-8:35	1. Overview of Eclipse and PTP	✦ Introduction to Eclipse/PTP	Greg
8:35-8:50	2. Installation	✦ Prerequisites ✦ Installation	Greg
8:50-9:20	3. Working with C/C++	✦ Eclipse basics ✦ Creating a new project ✦ Building and launching	Beth
9:20-10:50	4. Working with MPI	✦ CVS, Makefiles, autoconf, PLDT MPI tools ✦ Resource Managers ✦ Launching a parallel application	Jay
10:00 - 10:30	Break		
10:50-11:10	5. Fortran	✦ Photran overview ✦ MPI project creation ✦ Differences from CDT	Jeff
11:10-11:30	6. Debugging	✦ Introduction to parallel debugging ✦ Debugging an MPI program	Greg
11:30 – 11:50	7. Advanced Features	✦ Perspectives, Views, Preferences, Team ✦ Refactoring/Search (Fortran & C/C++) ✦ PLDT (MPI, OpenMP, UPC tools) ✦ Remote Development	Jeff/Beth
11:50- 12:00	8. Other Tools, Wrapup	✦ NCSA HPC WB, Perf and other Tools, website, mailing lists, future features	Jay/Beth

Module 1: Introduction

- Objective
    - To introduce the Eclipse platform and PTP
- Contents
    - What is Eclipse?
    - What is PTP?

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 Platform

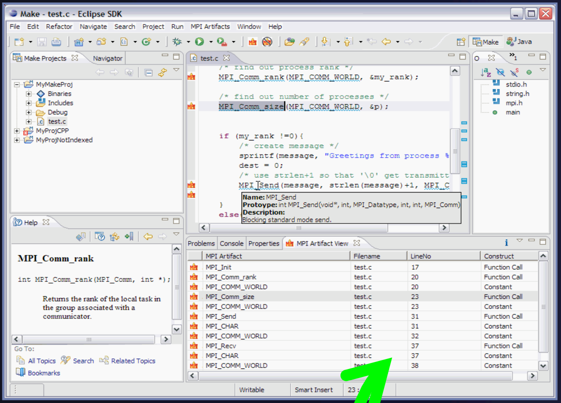
- Core frameworks and services with which all plug-in extensions are created
- Represents the common facilities required by most tool builders:
  - Workbench user interface
  - Project model for resource management
  - Portable user interface libraries (SWT and JFace)
  - Automatic resource delta management for incremental compilers and builders
  - Language-independent debug infrastructure
  - Distributed multi-user versioned resource management (CVS supported in base install)
  - Dynamic update/install service

Plug-ins

- Java Development Tools (JDT)
- Plug-in Development Environment (PDE)
- C/C++ Development Tools (CDT)
- Parallel Tools Platform (PTP)
- Fortran Development Tools (Photran)
- Test and Performance Tools Platform (TPTP)
- Business Intelligence and Reporting Tools (BIRT)
- Web Tools Platform (WTP)
- Data Tools Platform (DTP)
- Device Software Development Platform (DSDP)
- Many more…

Eclipse Parallel Tools Platform (PTP)

Coding & Analysis



```
    /* find out process rank */
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

    /* find out number of processes */
    MPI_Comm_size(MPI_COMM_WORLD, &p);


    if (my_rank !=0){
        /* create message */
        sprintf(message, "Greetings from process %
        dest = 0;
        /* use strlen+1 so that '\0' get transmitt
        MPI_Send(message, strlen(message)+1, MPI_C

    }
    else
```

Name: MPI_Send
Protoype: int MPI_Send(void*, int, MPI_Datatype, int, int, MPI_Comm)
Description:
Blocking standard mode send.

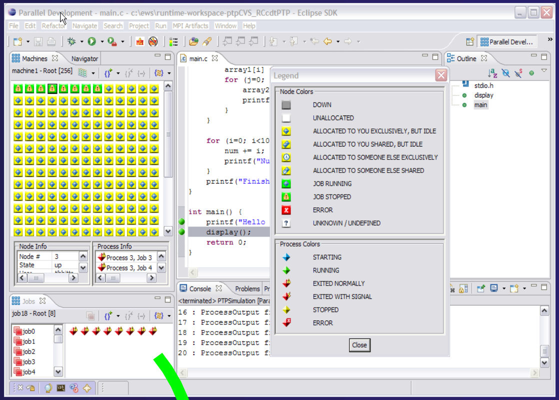
MPI_Comm_rank

`int MPI_Comm_rank(MPI_Comm, int *);`

Returns the rank of the local task in the group associated with a communicator.

MPI Artifact	Filename	LineNo	Construct
MPI_Init	test.c	17	Function Call
MPI_Comm_rank	test.c	20	Function Call
MPI_COMM_WORLD	test.c	20	Constant
MPI_Comm_size	test.c	23	Function Call
MPI_COMM_WORLD	test.c	23	Constant
MPI_Send	test.c	31	Function Call
MPI_CHAR	test.c	31	Constant
MPI_COMM_WORLD	test.c	32	Constant
MPI_Recv	test.c	37	Function Call
MPI_CHAR	test.c	37	Constant
MPI_COMM_WORLD	tes	38	Constant

Launching & Monitoring


Parallel Development - main.c - c:\ews\runtime-workspace-ptpCVS_RCcdtPTP - Eclipse SDK
File Edit Refactor Navigate Search Project Run MPI Artifacts Window Help
Parallel Devel...
Machines
Navigator
machine1 - Root [256]
main.c
Outline
stdio.h
display
main
Jobs
job18 - Root [8]
job0
job1
job2
job3
job4
Console
Problems

```
        array1[i]
        for (j=0;
            array2
            printf
        }
    }

    for (i=0; i<10
        num += i;
        printf("Nu
    }
    printf("Finish
}

int main() {
    printf("Hello
    display();
    return 0;
}
```

Node Info	
Node #	3
State	up

Process Info
Process 3, Job 3
Process 3, Job 4

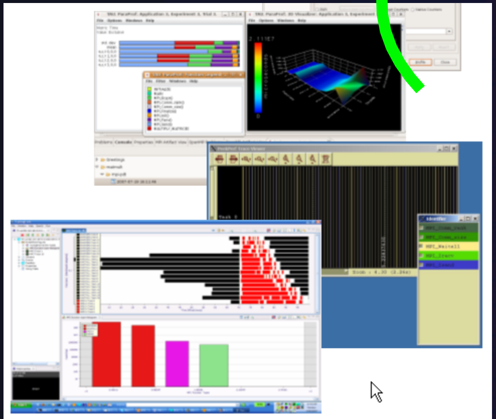
```
<terminated> PTPSimulation [Para
16 : ProcessOutput f
17 : ProcessOutput f
18 : ProcessOutput f
19 : ProcessOutput f
20 : ProcessOutput f
```

Legend

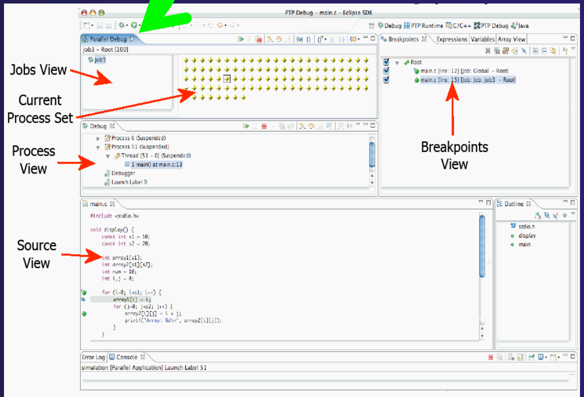
Node Colors
- ■ DOWN
- □ UNALLOCATED
- ◆ ALLOCATED TO YOU EXCLUSIVELY, BUT IDLE
- ◆ ALLOCATED TO YOU SHARED, BUT IDLE
- ◆ ALLOCATED TO SOMEONE ELSE EXCLUSIVELY
- ◆ ALLOCATED TO SOMEONE ELSE SHARED
- ◆ JOB RUNNING
- ◆ JOB STOPPED
- ☒ ERROR
- ? UNKNOWN / UNDEFINED

Process Colors
- ◆ STARTING
- ◆ RUNNING
- ◆ EXITED NORMALLY
- ◆ EXITED WITH SIGNAL
- ◆ STOPPED
- ◆ ERROR

Close



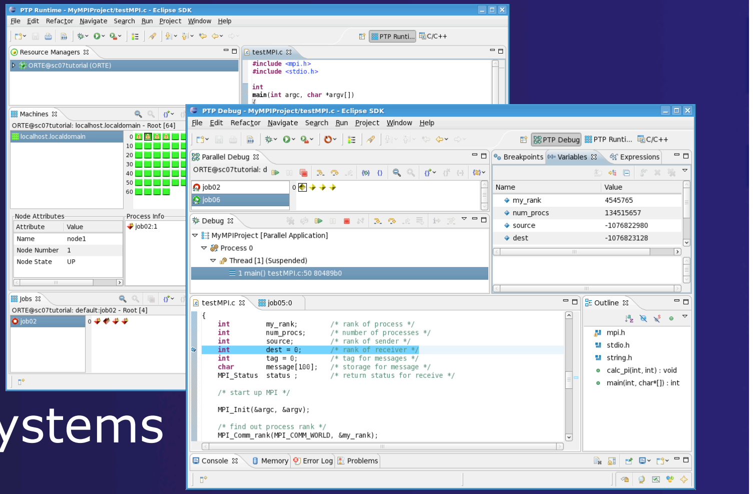
Performance Tuning



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)
  - Support for the integration of parallel tools
  - An environment that simplifies the end-user interaction with parallel systems
- http://www.eclipse.org/ptp



```
#include <mpi.h>
#include <stdio.h>

int
main(int argc, char *argv[])
```

Attribute	Value
Name	node1
Node Number	1
Node State	UP

```
MyMPIProject [Parallel Application]
  Process 0
    Thread [1] (Suspended)
      1 main() testMPI.c:50 80489b0
```

Name	Value
my_rank	4545765
num_procs	134515657
source	-1076822980
dest	-1076823128

```
{
    int          my_rank;       /* rank of process */
    int          num_procs;     /* number of processes */
    int          source;        /* rank of sender */
    int          dest = 0;      /* rank of receiver */
    int          tag = 0;       /* tag for messages */
    char         message[100];  /* storage for message */
    MPI_Status   status ;       /* return status for receive */

    /* start up MPI */

    MPI_Init(&argc, &argv);

    /* find out process rank */
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
```

```
mpi.h
stdio.h
string.h
calc_pi(int, int) : void
main(int, char*[]) : int
```

Module 2: Installation

- Objective
  - To learn how to install Eclipse and PTP
- Contents
  - System Prerequisites
  - Software Prerequisites
  - Eclipse Installation
  - PTP Installation

About PTP Installation

- PTP 3.0 isn’t “official” yet. Planned for late Oct.
- 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/3.0
- This information may supersede these slides

System Prerequisites

- Local system (running Eclipse)
  - Linux (just about any version)
  - MacOSX (Leopard)
  - Windows (XP on)
- Remote system (running/debugging application)
  - Must be supported by a resource manager
  - Open MPI 1.2+
  - MPICH 2
  - IBM PE & LoadLeveler (AIX or Linux)
  - SLURM (Linux)

Software Prerequisites

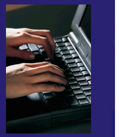
- Java (1.5 or later)
- Cygwin or MinGW (for local development on Windows)
- Unix make or equivalent
- Supported compilers (gcc, gfortran, Intel, etc.)
- Gdb for debugging (or a gdb-like interface)
- Gcc for building the debugger and SLURM proxies from source
- IBM C for building the PE/LoadLeveler proxies from source

Java Prerequisite

- Eclipse requires Sun or IBM versions of Java
  - Only need Java runtime environment (JRE)
  - Java 1.5 is the same as JRE 5.0
  - The GNU Java Compiler (GCJ), which comes standard on Linux, will not work!

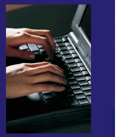
Eclipse and PTP Installation

- Eclipse is installed in two steps
  - First, the base Eclipse package is downloaded and installed
  - Additional functionality is obtained by adding ‘features’
    - This can be done via an `update site’ that automatically downloads and installs the features
    - Update site archives can be downloaded to install features offline.
- PTP requires the following Eclipse features
  - C/C++ Development Tools (CDT)
  - Remote Systems Explorer (RSE) end-user runtime



Eclipse Packages

- Eclipse is available in a number of different packages for different kinds of development
- Two packages are more relevant for HPC:
  - Eclipse Classic
    - The full software development kit (SDK), including Java and Plug-in development tools
  - Eclipse IDE for C/C++ developers
    - Base Eclipse distribution
    - Base C/C++ Development Tools (CDT) (does not include UPC)
- Either is ideal for PTP use

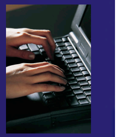


Eclipse Installation

- The current version of Eclipse is 3.5 (Galileo)
  - PTP 3.0 will only work with this version
- Eclipse is downloaded as a single zip or gzipped tar file from http://eclipse.org/downloads
- You *must* download the correct version to suit your local environment
  - Must have correct operating system version
  - Must have correct window system version
- Unzipping or untarring this file creates a directory containing the main executable

Platform Differences

- Single button mouse (e.g. MacBook)
  - Use Control-click for right mouse / context menu
- Context-sensitive help key differences
  - Windows: use **F1** key
  - Linux: use **Shift-F1** keys
  - MacOS X
    - Full keyboard, use **Help** key
    - MacBooks or aluminum keyboard, create a key binding for **Dynamic Help** to any key you want
- Accessing preferences
  - Windows & Linux: **Window▶Preferences…**
  - MacOS X: **Eclipse▶Preferences…**



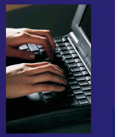
Starting Eclipse

- **Linux**
  - From a terminal window, enter

```
<eclipse_installation>/eclipse/eclipse &
```

- **MacOS X**
  - From finder, open the **eclipse** folder where you installed
  - Double-click on the **Eclipse** application
  - Or from a terminal window
- **Windows**
  - Open the **eclipse** folder
  - Double-click on the **eclipse** executable

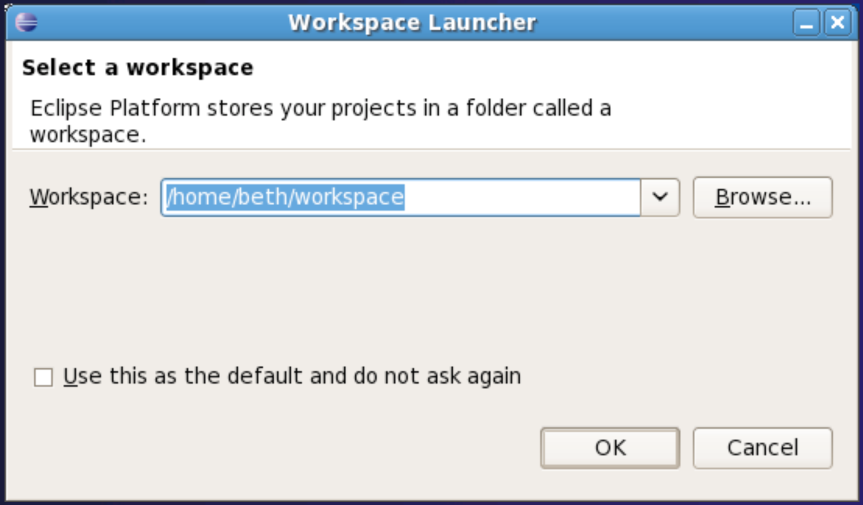
- Accept default workspace when asked
- Select workbench icon from welcome page



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 prompt can be turned off



Select a workspace

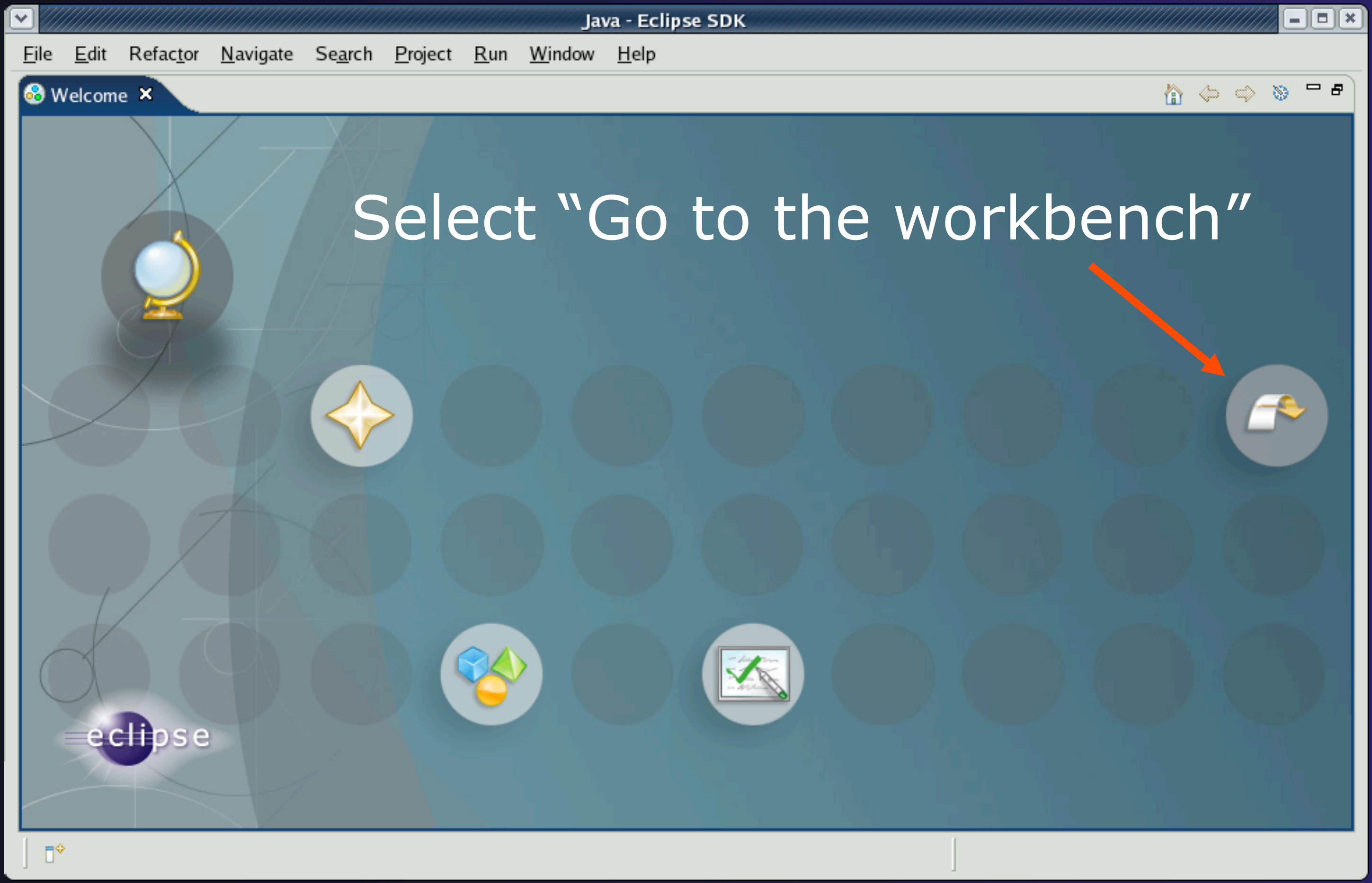
Eclipse Platform stores your projects in a folder called a workspace.

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"

Adding Features

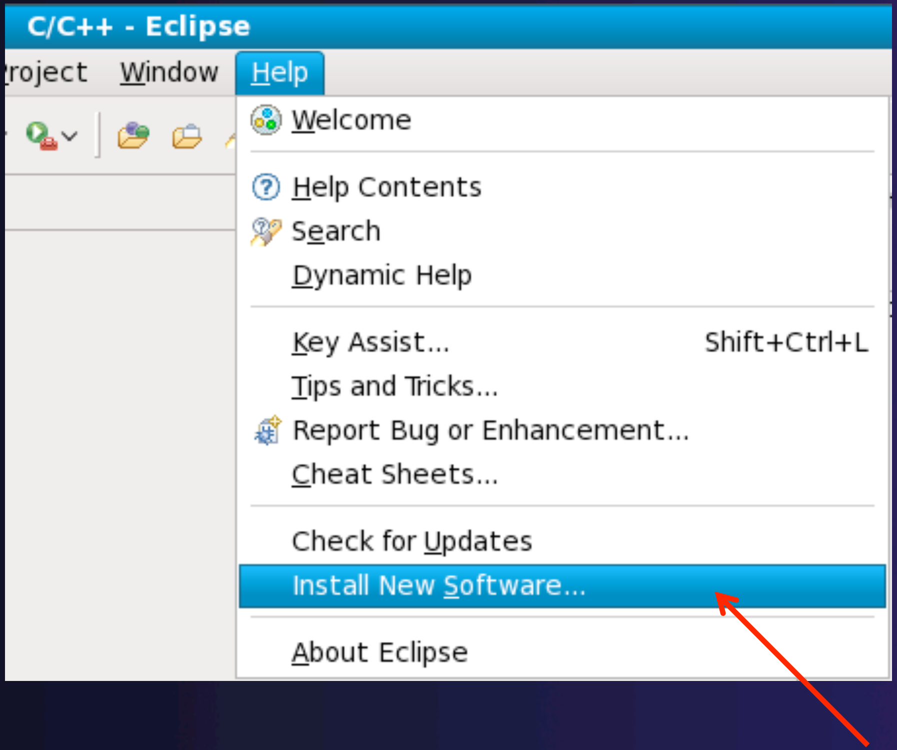
- New functionality is added to Eclipse using *features*
- Features are obtained and installed from an update site (like a web site)
- Features can also be installed from a local copy of the update site (which can be zipped)

Installing Eclipse Features from an Update Site

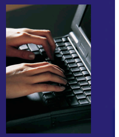
- Three types of update sites
  - **Remote** - download and install from remote server
  - **Local** - install from local directory
  - **Archived** - a local site packaged as a zip or jar file
- Eclipse 3.5 comes preconfigured with a link to the **Galileo** Update Site
  - This is a remote site that contains a large number of official features
  - Galileo projects are guaranteed to work with Eclipse 3.5
- Many other sites offer Eclipse features
  - Use at own risk

Installing from an Update Site

- From the **Help** menu, choose **Install New Software…**

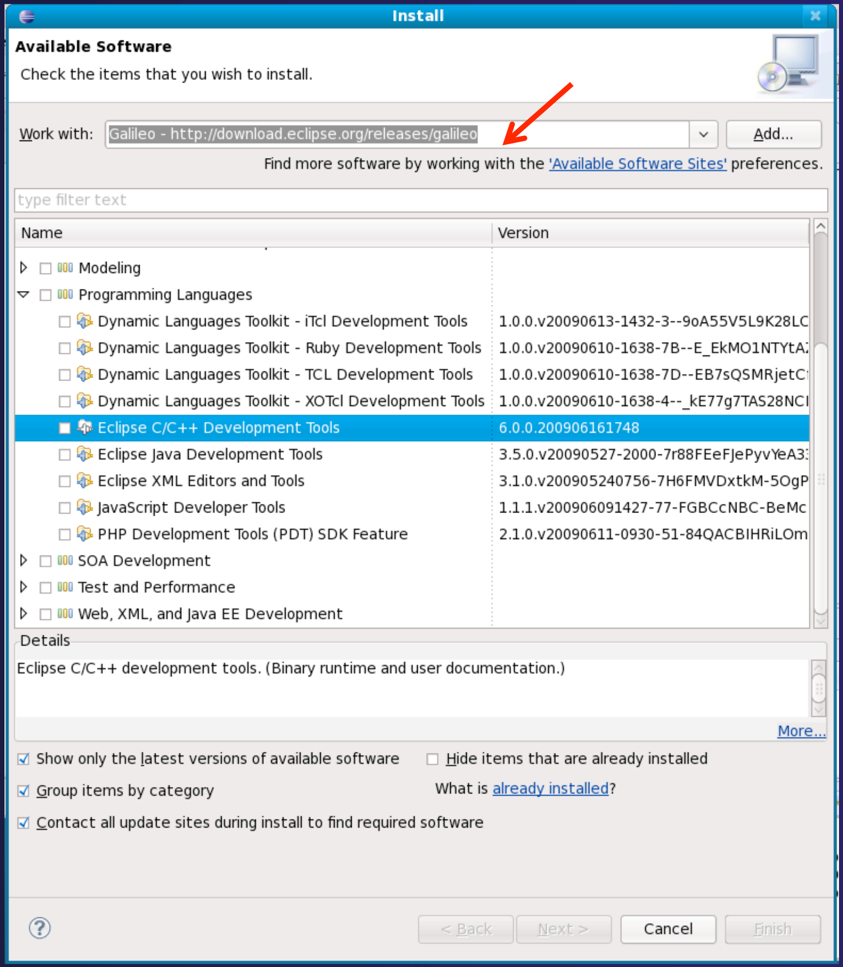


C/C++ - Eclipse



Galileo Update Site

- The Galileo site comes already configured with Eclipse
- For example, some of the contents of the Galileo site: →
- You can get C/C++ Dev. Tools from the Galileo site, but…
  - Basic tools, does not include UPC
  - More complete CDT install shown later
  - PTP 3.0 needs CDT 6.0.1, not available yet



Available Software

Check the items that you wish to install.

Work with: Galileo - http://download.eclipse.org/releases/galileo

Find more software by working with the 'Available Software Sites' preferences.

type filter text

Name	Version
▷ □ Modeling	
▽ □ Programming Languages	
□ Dynamic Languages Toolkit - iTcl Development Tools	1.0.0.v20090613-1432-3--9oA55V5L9K28LC
□ Dynamic Languages Toolkit - Ruby Development Tools	1.0.0.v20090610-1638-7B--E_EkMO1NTYtA
□ Dynamic Languages Toolkit - TCL Development Tools	1.0.0.v20090610-1638-7D--EB7sQSMRjetC
□ Dynamic Languages Toolkit - XOTcl Development Tools	1.0.0.v20090610-1638-4--_kE77g7TAS28NCI
□ Eclipse C/C++ Development Tools	6.0.0.200906161748
□ Eclipse Java Development Tools	3.5.0.v20090527-2000-7r88FEeFJePyvYeA3
□ Eclipse XML Editors and Tools	3.1.0.v200905240756-7H6FMVDxtkM-5OgP
□ JavaScript Developer Tools	1.1.1.v200906091427-77-FGBCcNBC-BeMc
□ PHP Development Tools (PDT) SDK Feature	2.1.0.v20090611-0930-51-84QACBIHRiLOm
▷ □ SOA Development	
▷ □ Test and Performance	
▷ □ Web, XML, and Java EE Development	

Details

Eclipse C/C++ development tools. (Binary runtime and user documentation.)

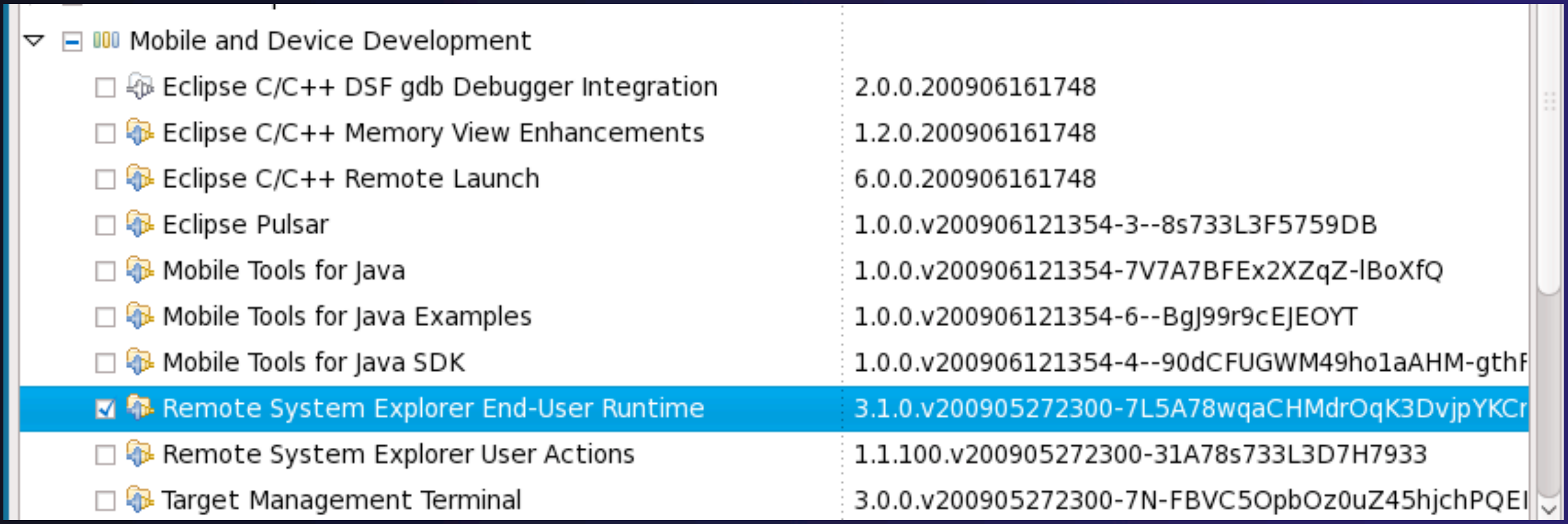
☑ Show only the latest versions of available software □ Hide items that are already installed

☑ Group items by category What is already installed?

☑ Contact all update sites during install to find required software

Installation: RSE

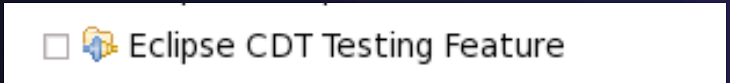
- The RSE End-User Runtime should be installed from the Galileo update site



⊟ Mobile and Device Development	
□ Eclipse C/C++ DSF gdb Debugger Integration	2.0.0.200906161748
□ Eclipse C/C++ Memory View Enhancements	1.2.0.200906161748
□ Eclipse C/C++ Remote Launch	6.0.0.200906161748
□ Eclipse Pulsar	1.0.0.v200906121354-3--8s733L3F5759DB
□ Mobile Tools for Java	1.0.0.v200906121354-7V7A7BFEx2XZqZ-lBoXfQ
□ Mobile Tools for Java Examples	1.0.0.v200906121354-6--BgJ99r9cEJEOYT
□ Mobile Tools for Java SDK	1.0.0.v200906121354-4--90dCFUGWM49ho1aAHM-gthF
☑ Remote System Explorer End-User Runtime	3.1.0.v200905272300-7L5A78wqaCHMdrOqK3DvjpYKCr
□ Remote System Explorer User Actions	1.1.100.v200905272300-31A78s733L3D7H7933
□ Target Management Terminal	3.0.0.v200905272300-7N-FBVC5OpbOz0uZ45hjchPQEI

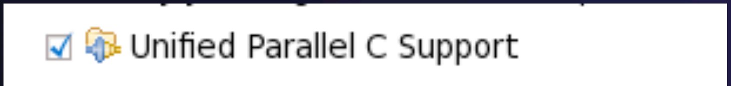
Installation: CDT

- PTP 3.0 needs CDT 6.0.1
  - Update site will contain latest version
- Update site: http://download.eclipse.org/tools/cdt/releases/galileo
- Install any features you want
  - Omit the testing feature:



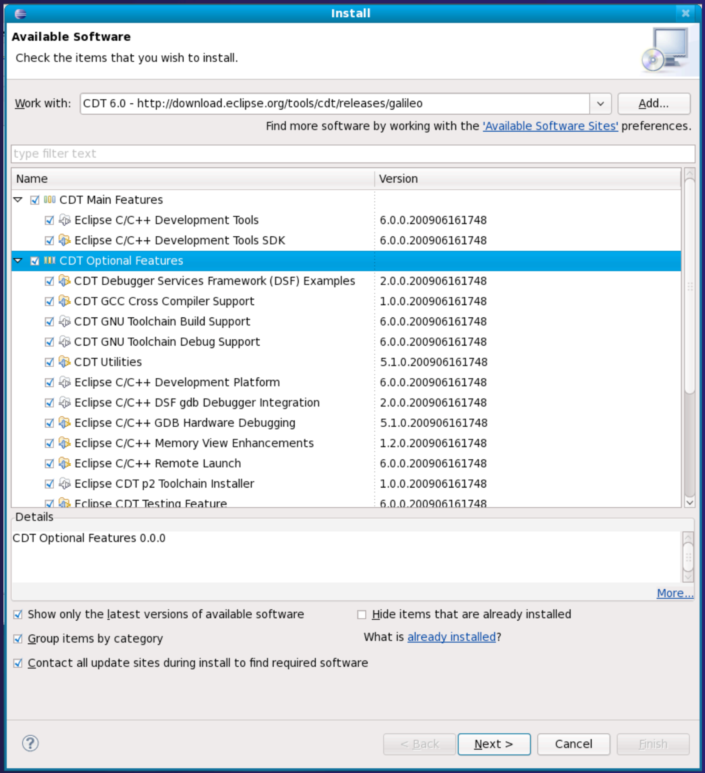
□ Eclipse CDT Testing Feature

  - If you want UPC, include:



☑ Unified Parallel C Support

- CDT 6.0.1 is due at the end of Sept



Available Software

Check the items that you wish to install.

Work with: CDT 6.0 - http://download.eclipse.org/tools/cdt/releases/galileo

Find more software by working with the 'Available Software Sites' preferences.

Name	Version
☑ CDT Main Features	
☑ Eclipse C/C++ Development Tools	6.0.0.200906161748
☑ Eclipse C/C++ Development Tools SDK	6.0.0.200906161748
☑ CDT Optional Features	
☑ CDT Debugger Services Framework (DSF) Examples	2.0.0.200906161748
☑ CDT GCC Cross Compiler Support	1.0.0.200906161748
☑ CDT GNU Toolchain Build Support	6.0.0.200906161748
☑ CDT GNU Toolchain Debug Support	6.0.0.200906161748
☑ CDT Utilities	5.1.0.200906161748
☑ Eclipse C/C++ Development Platform	6.0.0.200906161748
☑ Eclipse C/C++ DSF gdb Debugger Integration	2.0.0.200906161748
☑ Eclipse C/C++ GDB Hardware Debugging	5.1.0.200906161748
☑ Eclipse C/C++ Memory View Enhancements	1.2.0.200906161748
☑ Eclipse C/C++ Remote Launch	6.0.0.200906161748
☑ Eclipse CDT p2 Toolchain Installer	1.0.0.200906161748
☑ Eclipse CDT Testing Feature	6.0.0.200906161748

Details

CDT Optional Features 0.0.0

☑ Show only the latest versions of available software

□ Hide items that are already installed

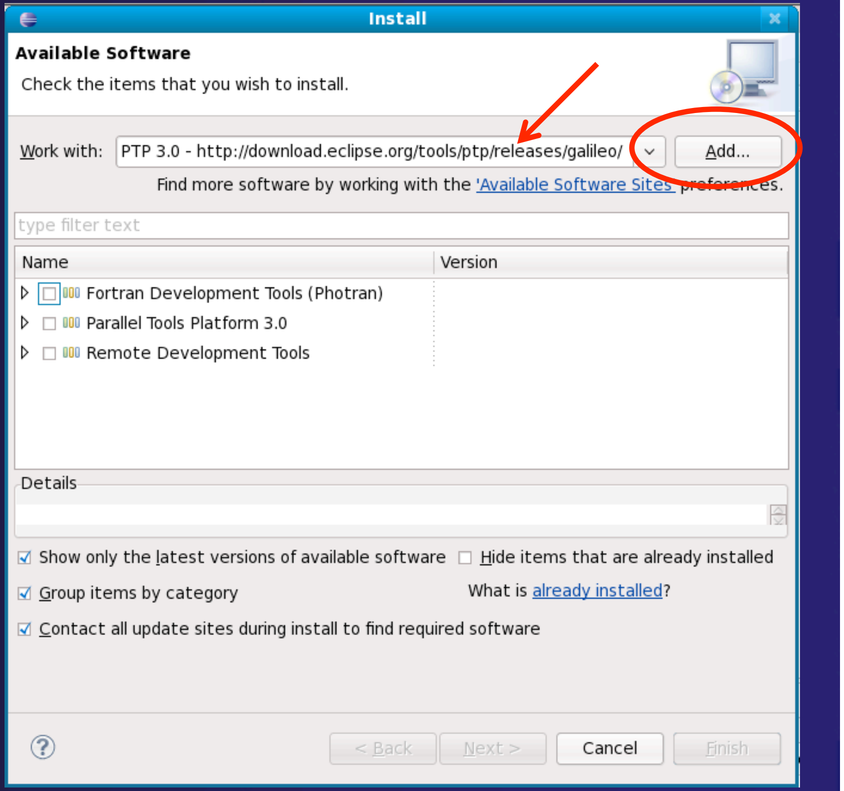
☑ Group items by category

What is already installed?

☑ Contact all update sites during install to find required software

Installing PTP

- In **Work with** type the PTP update site URL: http://download.eclipse.org/tools/ptp/releases/galileo/
- Click **Add…**
- Enter a name (optional) e.g. “PTP 3.0”
- Click **OK** and the list of features on the update site will be populated
- Select all the components you require
- Click **Next>**



Install

Available Software

Check the items that you wish to install.

Work with: PTP 3.0 - http://download.eclipse.org/tools/ptp/releases/galileo/

Find more software by working with the 'Available Software Sites' preferences.

type filter text

Name	Version
□ Fortran Development Tools (Photran)	
□ Parallel Tools Platform 3.0	
□ Remote Development Tools	

Details

☑ Show only the latest versions of available software □ Hide items that are already installed

☑ Group items by category

What is already installed?

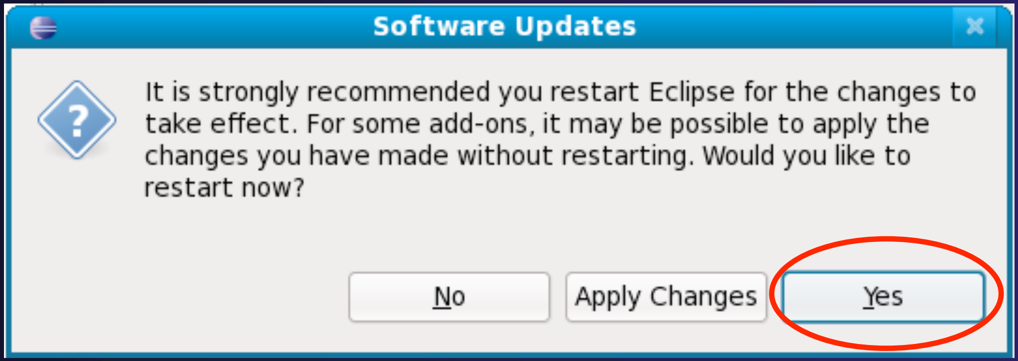
☑ Contact all update sites during install to find required software

- See PTP release notes for most recent info on installing 3.0

Installing PTP (2)

- You will be prompted to accept the License terms
- Accept the License terms
- Click **Finish**

- Restart Eclipse when prompted

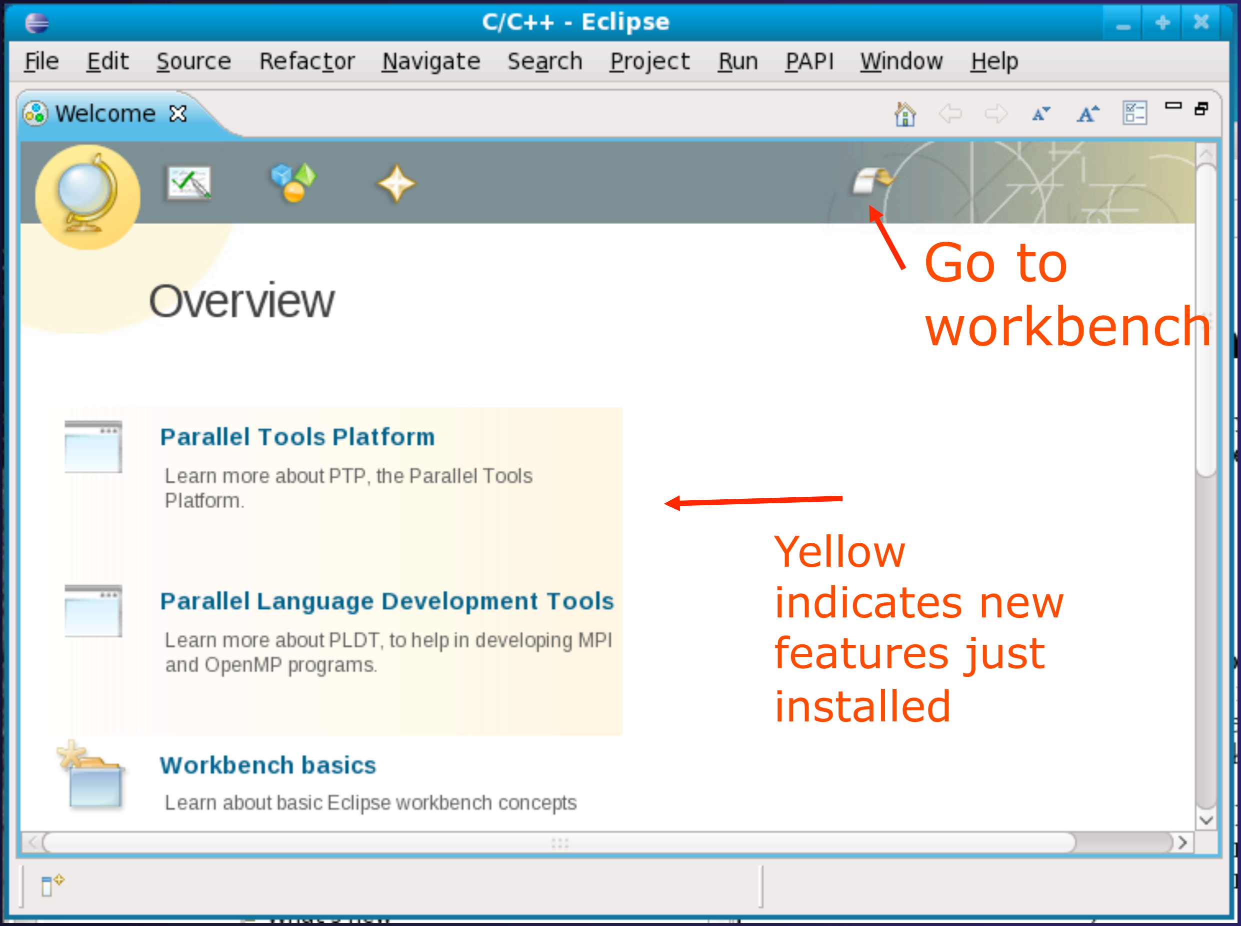


Software Updates

It is strongly recommended you restart Eclipse for the changes to take effect. For some add-ons, it may be possible to apply the changes you have made without restarting. Would you like to restart now?

Restarting Eclipse

- Welcome page informs you of new features installed
- Select workbench icon to go to workbench


C/C++ - Eclipse
File Edit Source Refactor Navigate Search Project Run PAPI Window Help
Welcome

Overview

Parallel Tools Platform

Learn more about PTP, the Parallel Tools Platform.

Parallel Language Development Tools

Learn more about PLDT, to help in developing MPI and OpenMP programs.

Workbench basics

Learn about basic Eclipse workbench concepts

Go to workbench

Yellow indicates new features just installed

Installing Additional PTP Components

- PTP has a number of additional components depending on the installation
  - Scalable Debug Manager (SDM) – required for all platforms to support debugging
  - PE and LoadLeveler proxy – IBM systems only
  - SLURM proxy – systems using the SLURM resource manager
- Installation of these components is beyond the scope of the tutorial
- See the release notes for details of installing these components

Module 3: Working with C/C++

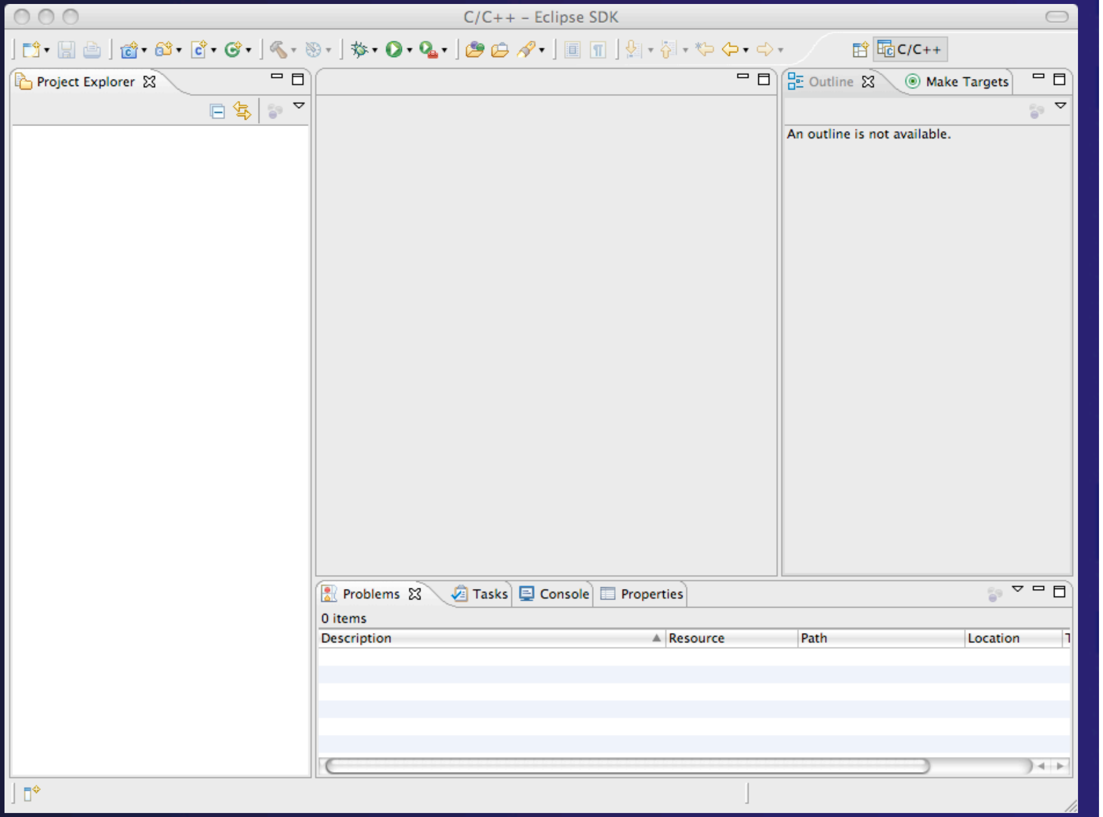
- Objective
  - Learn how to use Eclipse to develop parallel programs
  - Learn how to run and monitor a parallel program
- Contents
  - Brief introduction to the C/C++ Development Tools
  - Create a simple application
  - Learn to launch a parallel job and view it via the PTP Runtime Perspective

Installation recap

- Download and unzip/untar eclipse
- Use Help >Install new software to get
  - CDT for C/C++ tools
  - PTP and related tools for Parallel application work
  - Build PTP binary on target machine (local or remote)
- Launch eclipse!
  Run the ‘eclipse’ executable, from icon or from command line

Workbench

- The Workbench represents the desktop development environment
  - It contains a set of tools for resource management
  - It provides a common way of navigating through the resources
- Multiple workbenches can be opened at the same time

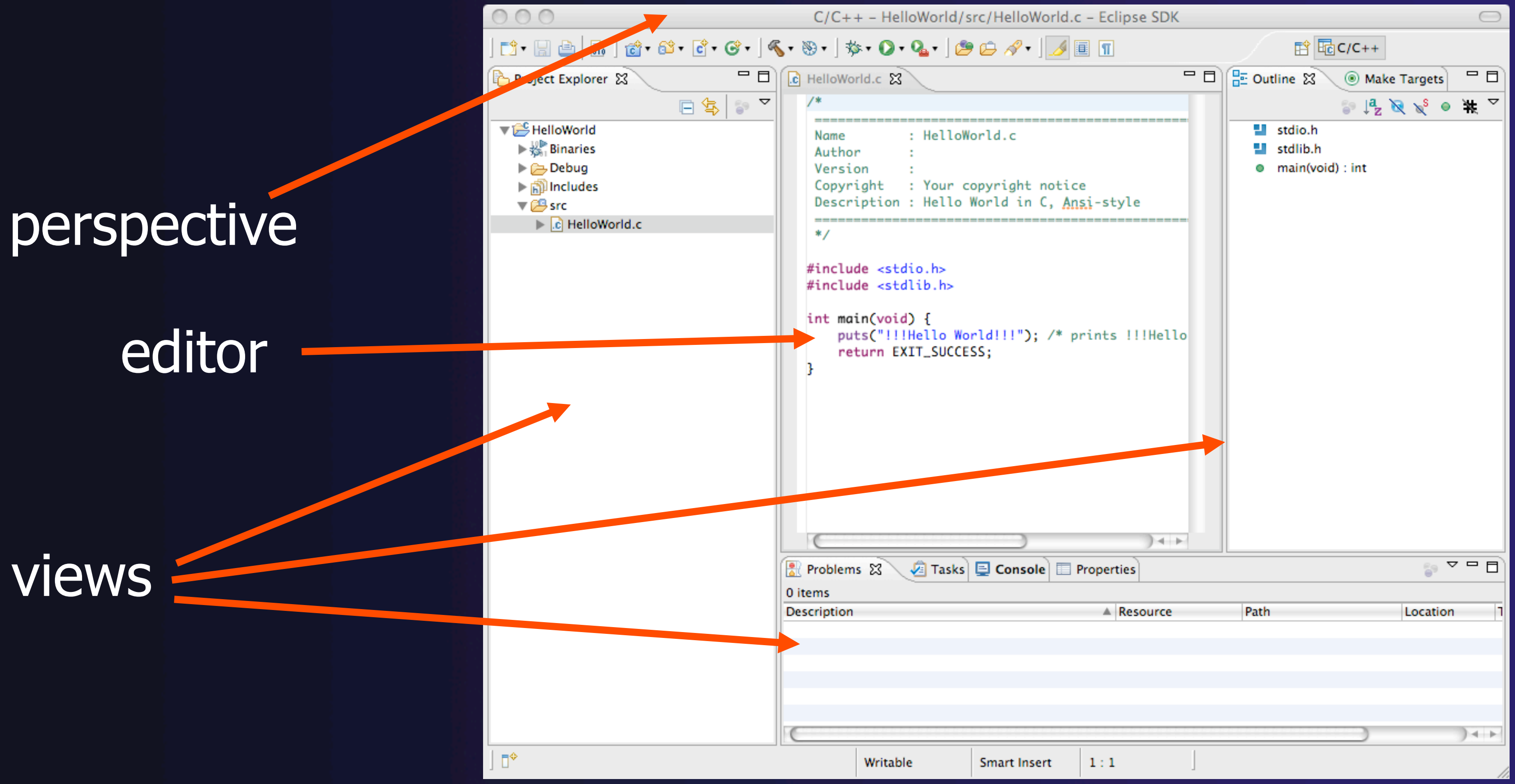


An outline is not available.

0 items

Workbench Components

- A Workbench contains perspectives
- A Perspective contains views and editors



perspective

editor

views

```
/*
 ============================================================================
 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;
}
```

0 items

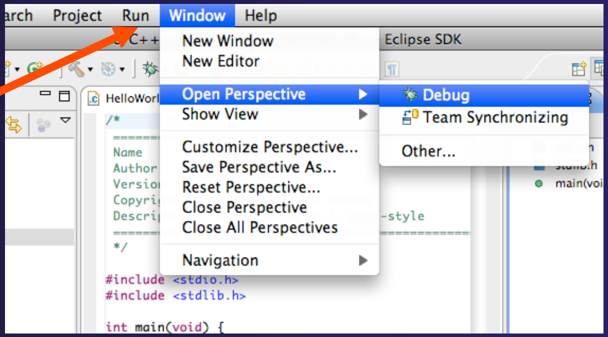
Description	Resource	Path	Location

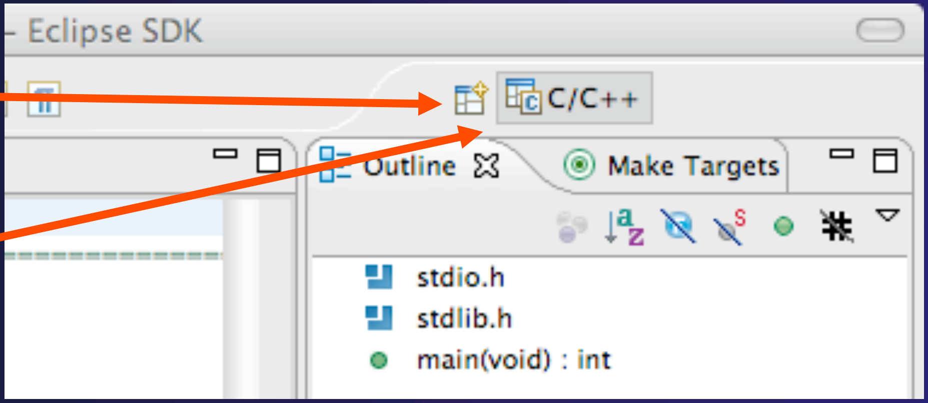
Perspectives

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

Switching Perspectives

- You can switch Perspectives by:
  - Choosing the **Window▸Open Perspective** menu option
  - Clicking on the **Open Perspective** button
  - Clicking on a perspective shortcut button

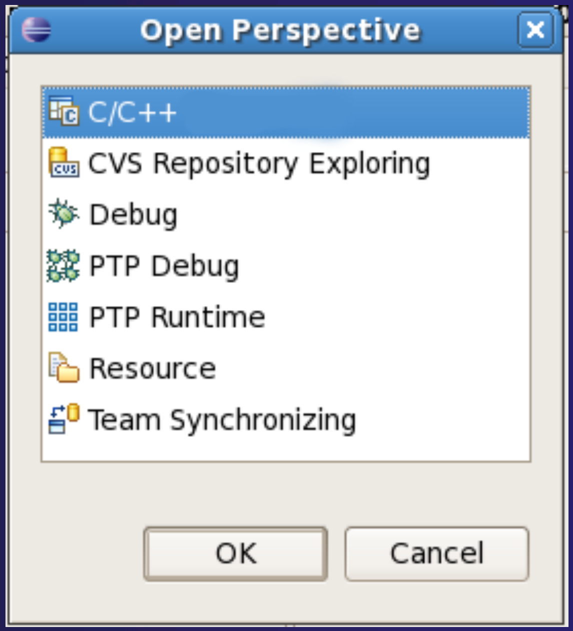




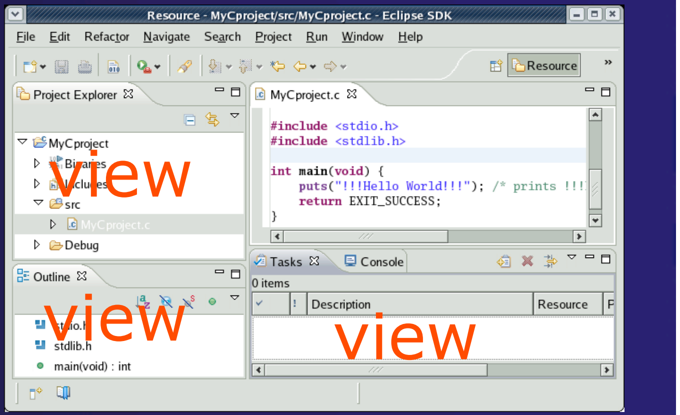
Available Perspectives

- By default, certain perspectives are available in the Workbench
- We'll use:
  - C/C++
  - PTP Runtime
  - PTP Debug

Window ▸ Open Perspective

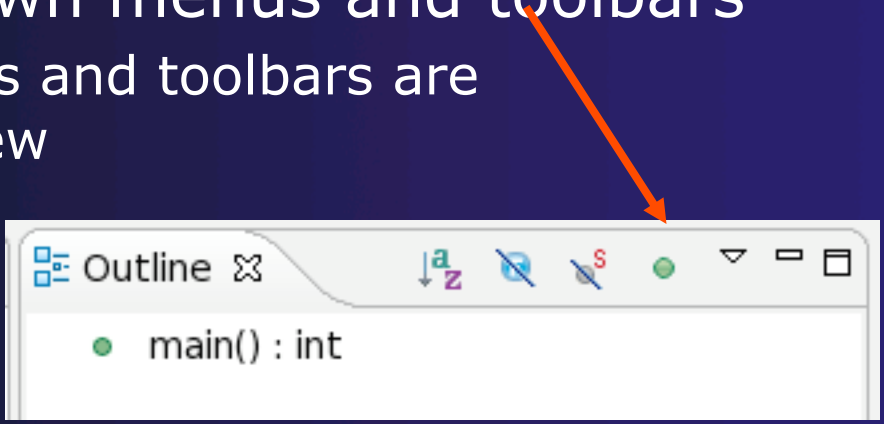


Views



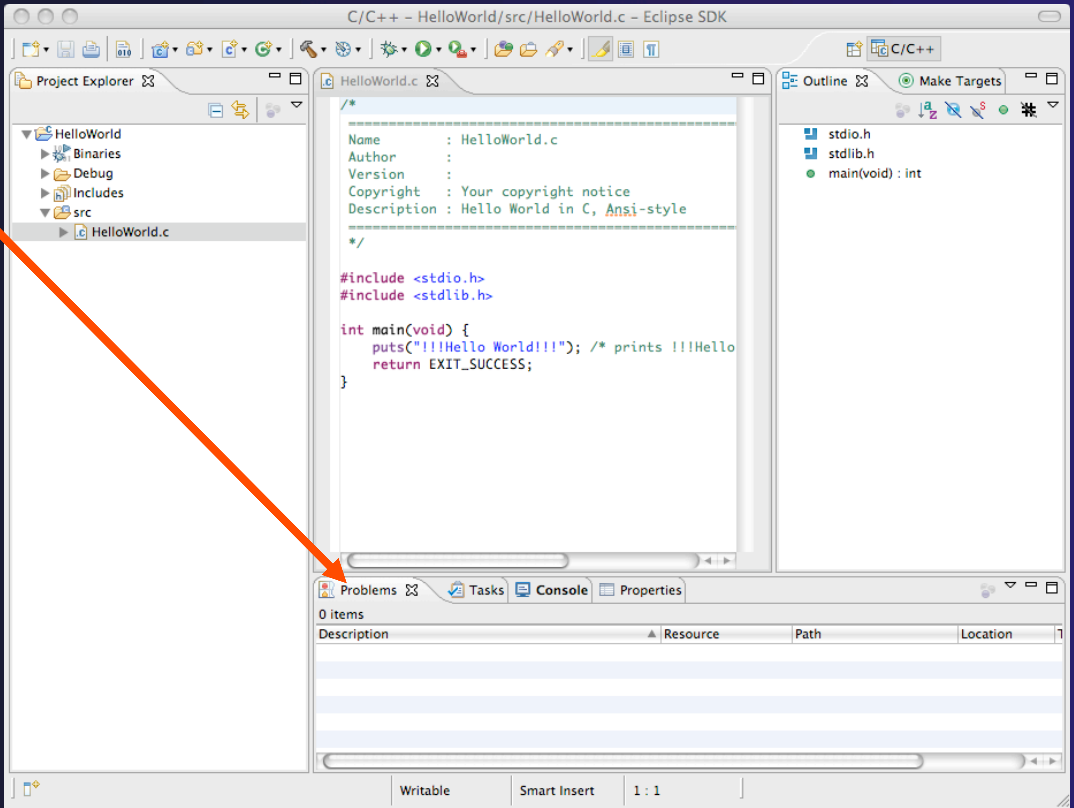
```
#include <stdio.h>
#include <stdlib.h>

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

- ✦ 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



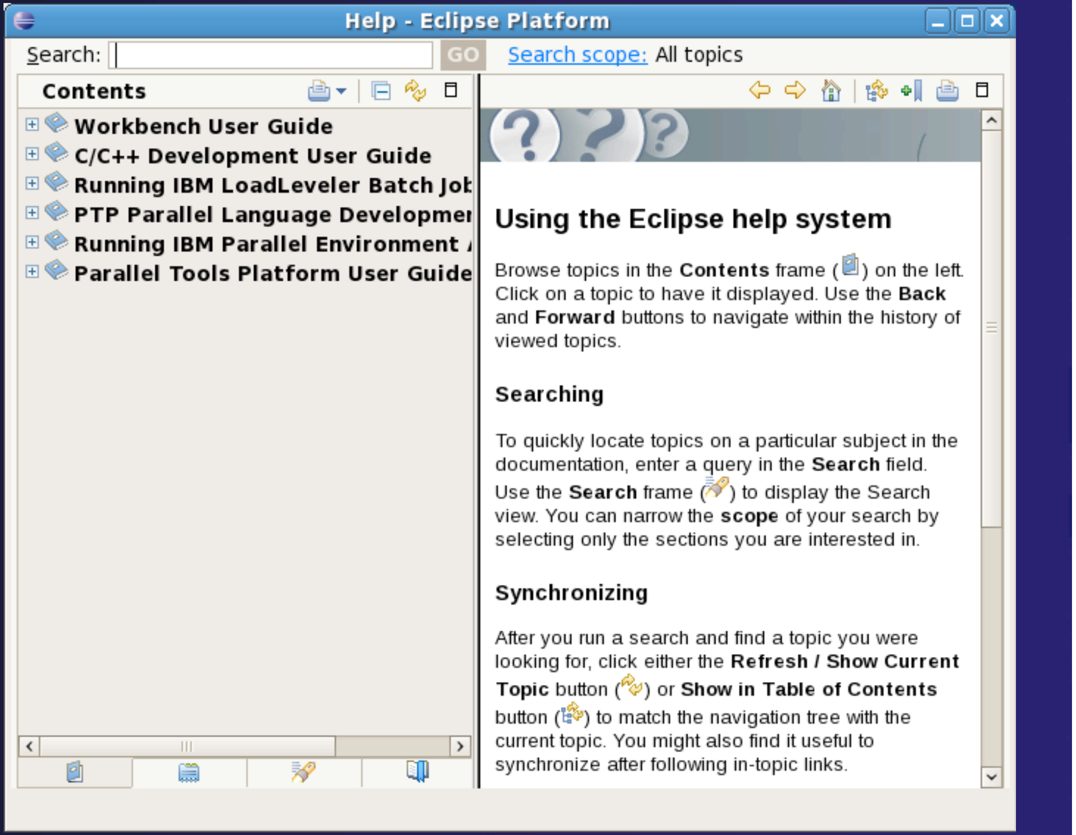
```
/*
 ============================================================
 Name        : HelloWorld.c
 Author      :
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style
 ============================================================
 */

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

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

0 items

Description	Resource	Path	Location

Help

- Access help
  - **Help▶Help Contents**
  - **Help▶Search**
  - **Help▶Dynamic Help**
- **Help Contents** provides detailed help on different Eclipse features
- **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.)



- Workbench User Guide
- C/C++ Development User Guide
- Running IBM LoadLeveler Batch Jot
- PTP Parallel Language Developmer
- Running IBM Parallel Environment /
- Parallel Tools Platform User Guide

Using the Eclipse 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

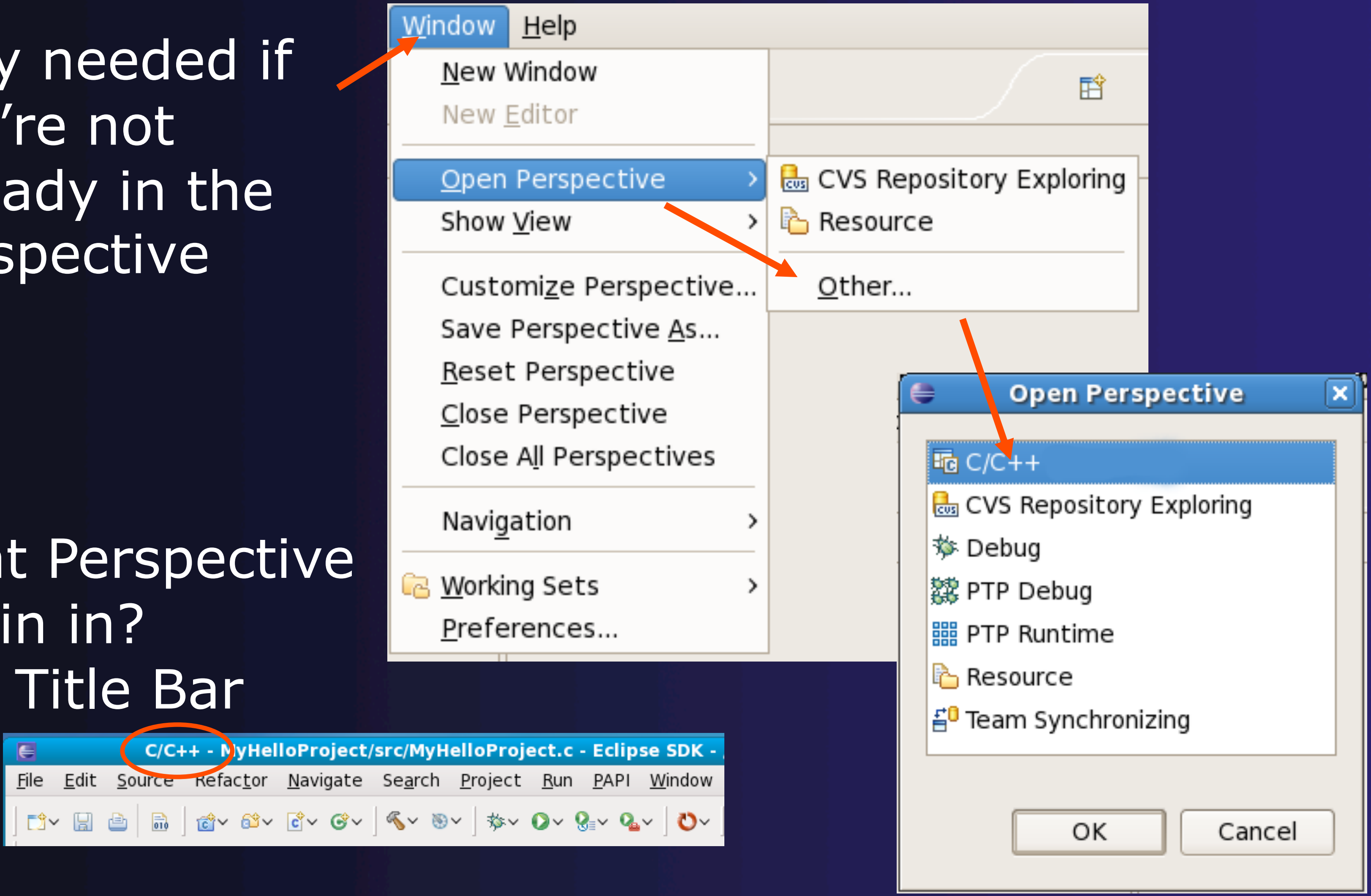
To quickly locate topics on a particular subject in the documentation, enter a query in the **Search** field. Use the **Search** frame ( ) to display the Search view. You can narrow the **scope** of your search by selecting only the sections you are interested in.

Synchronizing

After you run a search and find a topic you were looking for, click either the **Refresh / Show Current Topic** button ( ) or **Show in Table of Contents** button ( ) to match the navigation tree with the current topic. You might also find it useful to synchronize after following in-topic links.

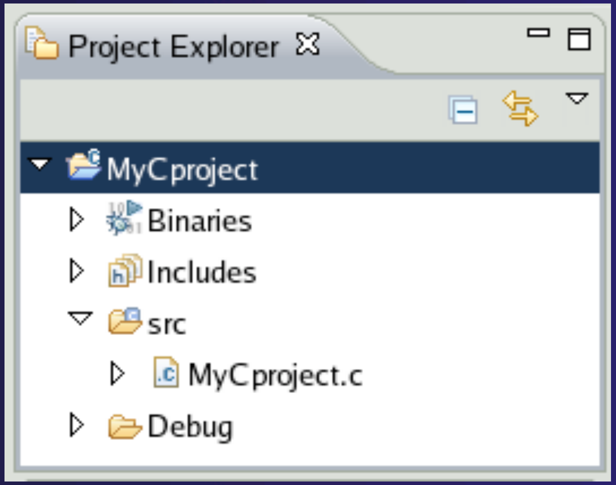
Switch to C/C++ Perspective

- Only needed if you’re not already in the perspective
- What Perspective am in in?
  See Title Bar

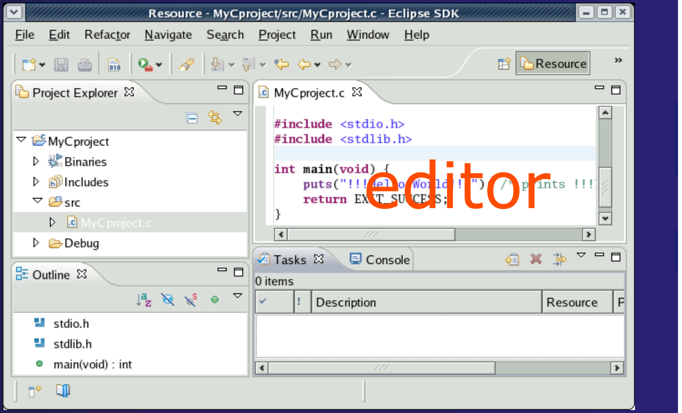


Project Explorer View

- Represents user’s data
- It is a set of user defined resources
  - Files
  - Folders
  - Projects
    - Collections of files and folders
    - Plus meta-data
- Resources are visible in the Project Explorer View

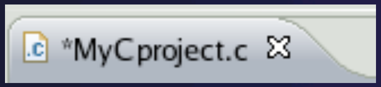


Editors



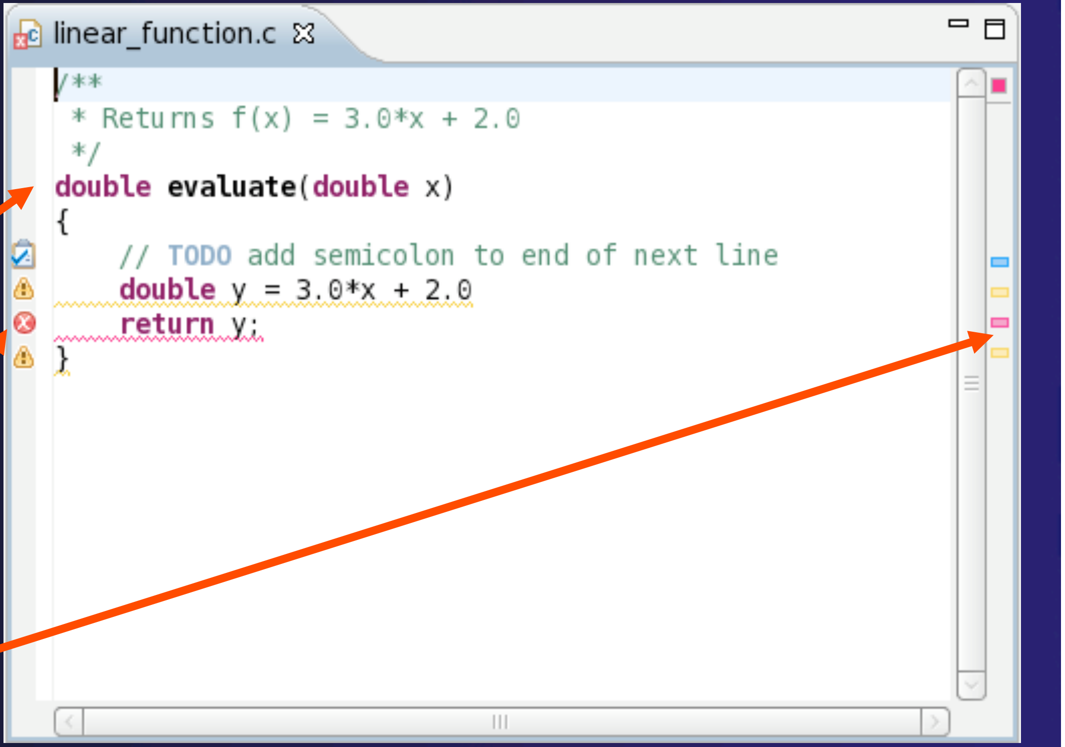
```
#include <stdio.h>
#include <stdlib.h>

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

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



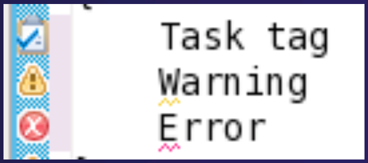
Source Code Editors

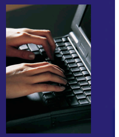
- A source code editor is a special type of editor for manipulating source code
- Language features are highlighted
- Marker bars for showing
  - Breakpoints
  - Errors/warnings
  - Tasks
- Location bar for navigating to interesting features



```
/**
 * 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;
}
```

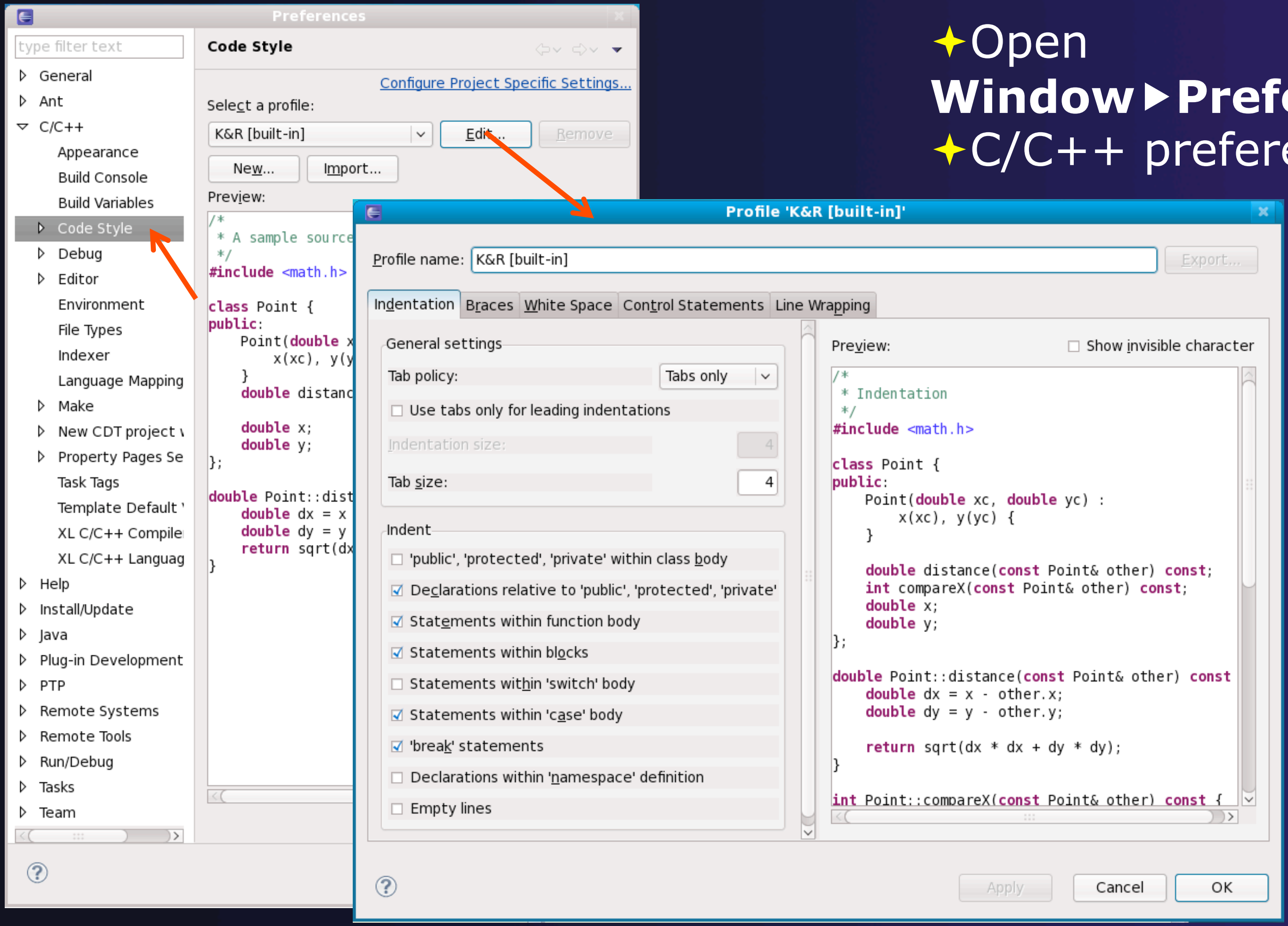
Icons:


Task tag
Warning
Error



Preferences

- Eclipse Preferences allow customization of almost everything



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

Preview:

```
/*
 * A sample source
 */
#include <math.h>

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

    double x;
    double y;
};

double Point::dist
    double dx = x
    double dy = y
    return sqrt(dx
}
```

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:

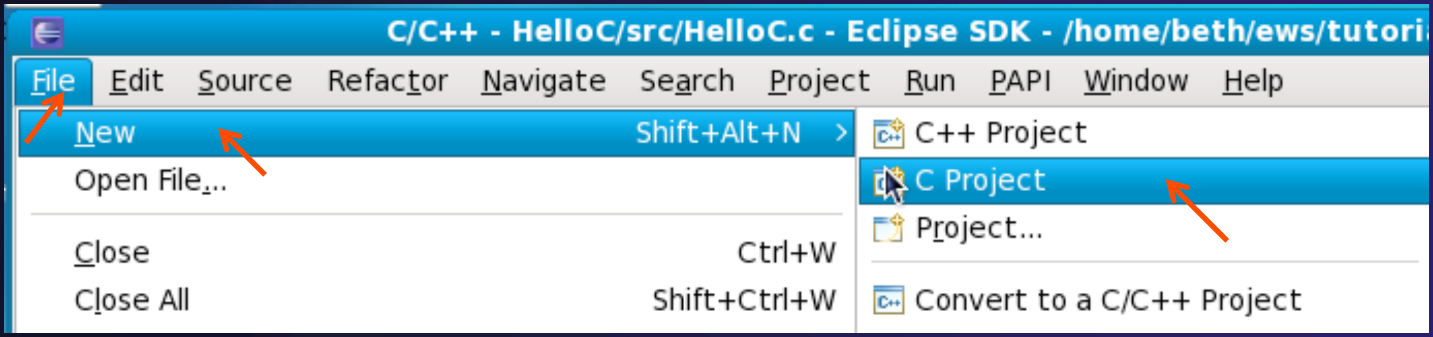
```
/*
 * Indentation
 */
#include <math.h>

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

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

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

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

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

- Open **Window ▶ Preferences...**
- C/C++ preferences allow many options
- Code formatting settings ("Code Style") shown here

Creating a C/C++ Application

Steps:

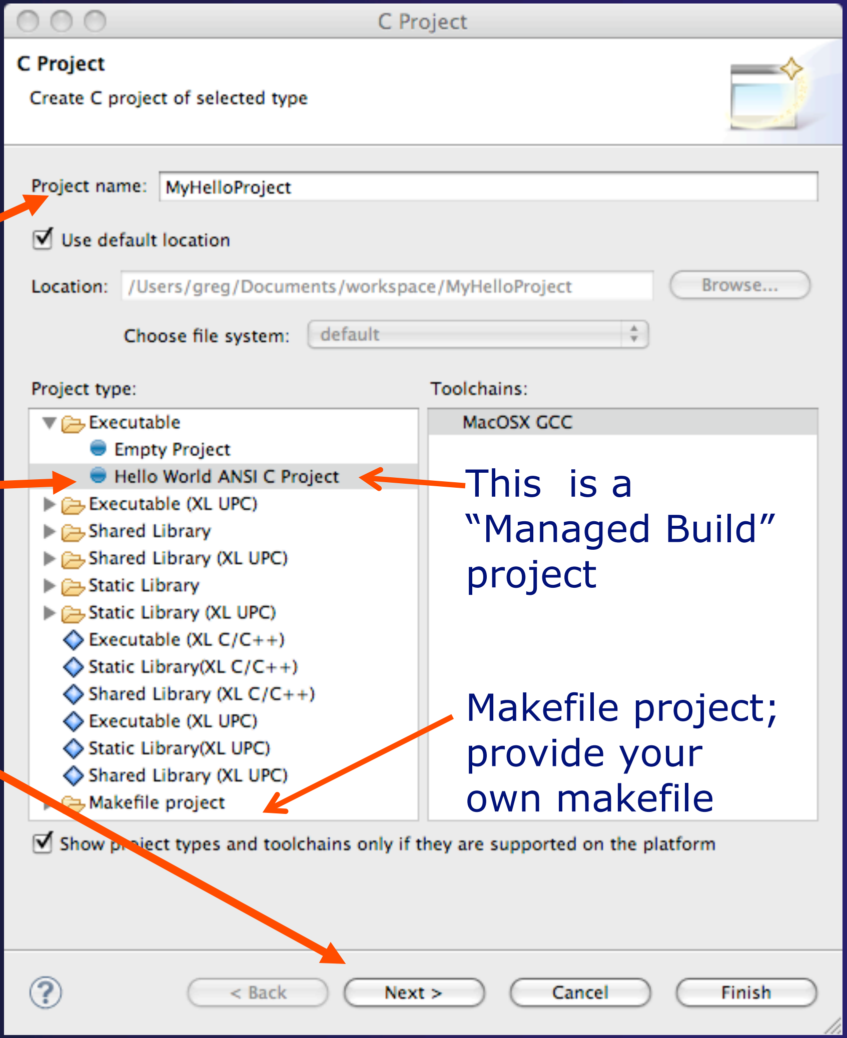
- ✦ Create a new C project
- ✦ Edit source code
- ✦ Save and build



New C Project Wizard

Create a new MPI project

- **File▸New▸C Project** (see prev. slide)
- Name the project 'MyHelloProject'
- Under Project types, under Executable, select **Hello World ANSI C Project**
  (no makefile req'd)
  and hit **Next**
- On **Basic Settings** page, fill in information for your new project (**Author name** etc.) and hit **Next**



C Project

Create C project of selected type

Project name: MyHelloProject

☑ Use default location

Location: /Users/greg/Documents/workspace/MyHelloProject Browse...

Choose file system: default

Project type:

- Executable
  - Empty Project
  - Hello World ANSI C Project
- Executable (XL UPC)
- Shared Library
- Shared Library (XL UPC)
- Static Library
- Static Library (XL UPC)
- Executable (XL C/C++)
- Static Library(XL C/C++)
- Shared Library (XL C/C++)
- Executable (XL UPC)
- Static Library(XL UPC)
- Shared Library (XL UPC)
- Makefile project

Toolchains:

MacOSX GCC

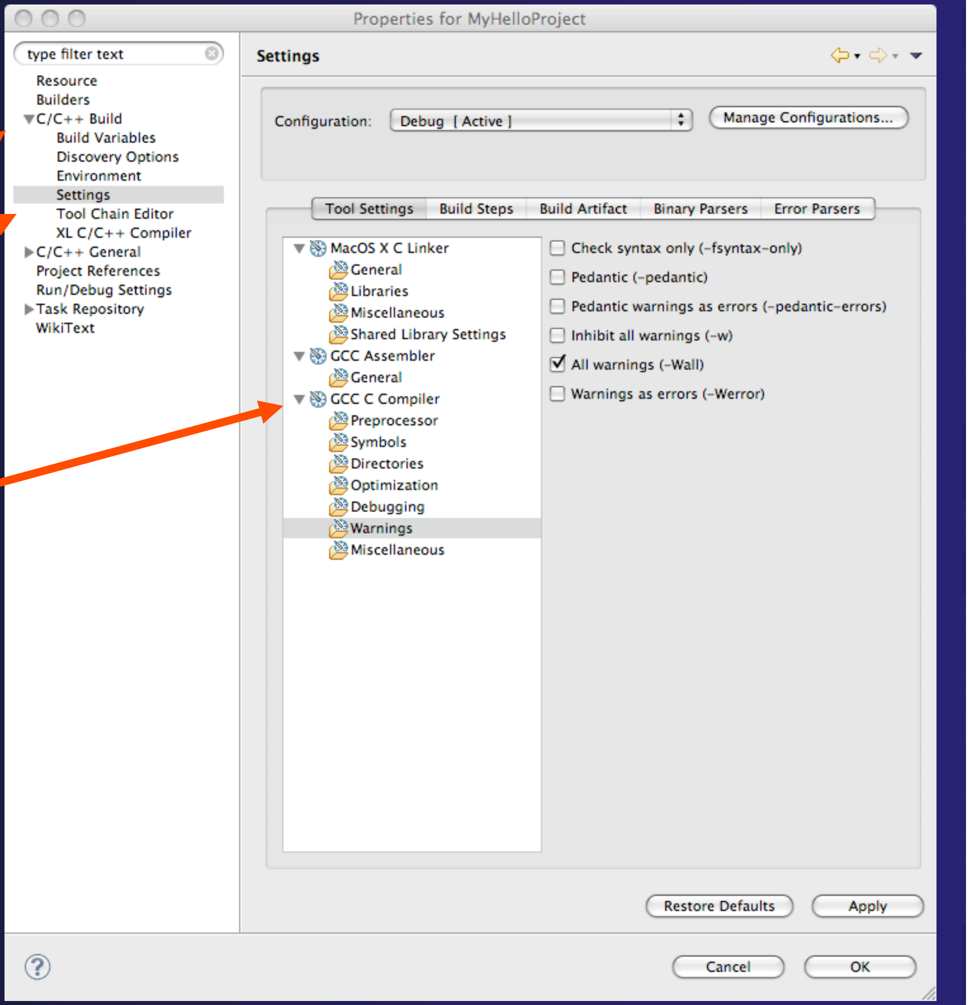
☑ Show project types and toolchains only if they are supported on the platform

This is a "Managed Build" project

Makefile project; provide your own makefile

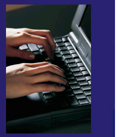
Changing the C/C++ Build Settings Manually

- Open the project properties by right-mouse clicking on project and select **Properties**
- Open **C/C++ Build**
- Select **Settings**
- Select **C Compiler** to change compiler settings
- Select **C Linker** to change linker settings
- It's also possible to change compiler/linker arguments
- Hit **OK** to close



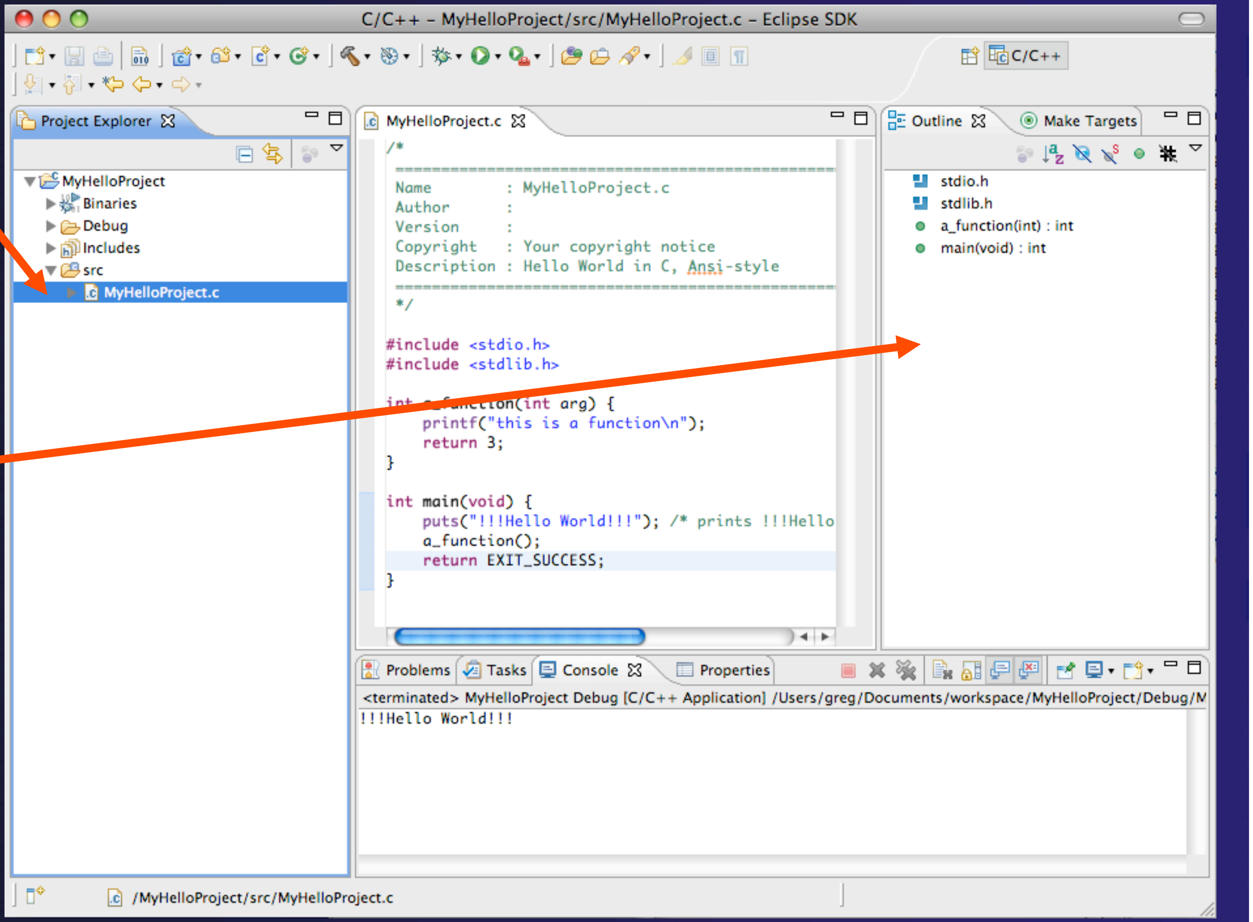
Configuration: Debug [Active]

□ Check syntax only (-fsyntax-only)
□ Pedantic (-pedantic)
□ Pedantic warnings as errors (-pedantic-errors)
□ Inhibit all warnings (-w)
☑ All warnings (-Wall)
□ Warnings as errors (-Werror)



Editor and Outline View

- Double-click on source file in the **Project Explorer** to open C editor
- Outline view is shown for file in editor

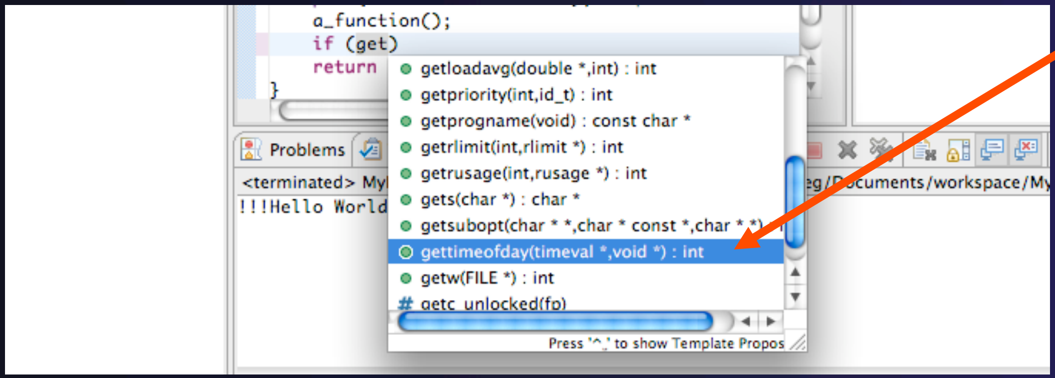


```
/*
 ============================================================
 Name        : MyHelloProject.c
 Author      :
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style
 ============================================================
 */

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

int a_function(int arg) {
    printf("this is a function\n");
    return 3;
}

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

```
<terminated> MyHelloProject Debug [C/C++ Application] /Users/greg/Documents/workspace/MyHelloProject/Debug/M
!!!Hello World!!!
```

Content Assist

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



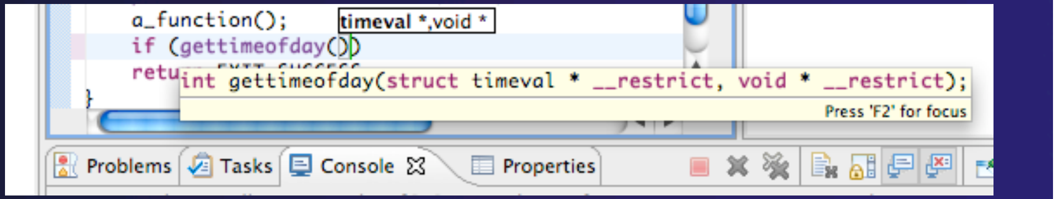
```
    a_function();
    if (get)
    return
}
```

```
getloadavg(double *,int) : int
getpriority(int,id_t) : int
getprogname(void) : const char *
getrlimit(int,rlimit *) : int
getrusage(int,rusage *) : int
gets(char *) : char *
getsubopt(char * *,char * const *,char * *)
gettimeofday(timeval *,void *) : int
getw(FILE *) : int
# getc_unlocked(fp)
```

Press '^.' to show Template Propos

```
<terminated> My
!!!Hello World
```

- Hover over a program element in the source file to see additional information

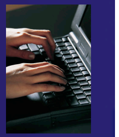


```
    a_function();
    if (gettimeofday())
    retu
}
```

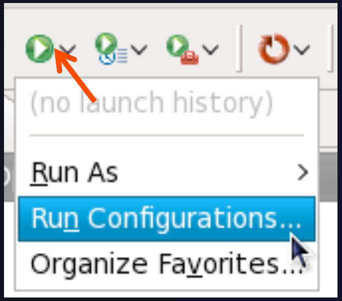
timeval *,void *

```
int gettimeofday(struct timeval * __restrict, void * __restrict);
```

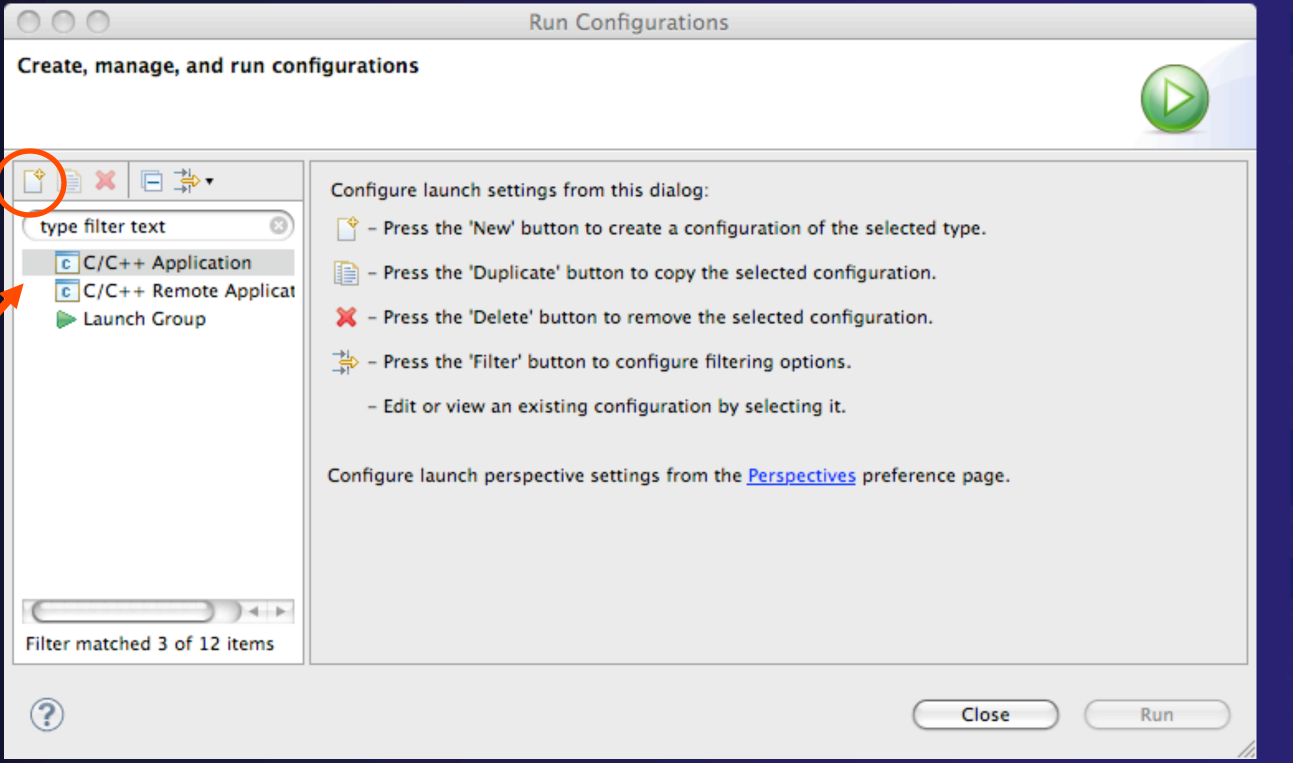
Press 'F2' for focus



Create a Launch Configuration



- ✦ Open the run configuration dialog **Run ▶ Run Configurations…**
- ✦ Select **C/C++ Application**
- ✦ Select the **New** button



Create, manage, and run configurations

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.

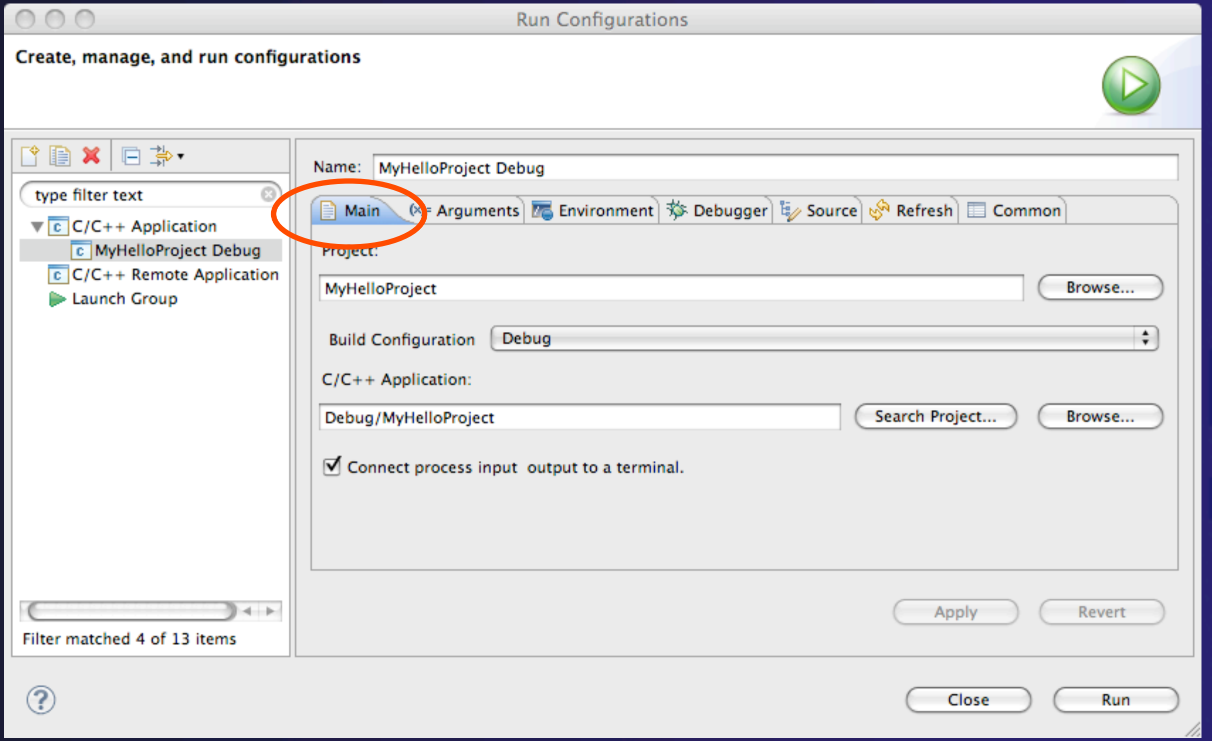
Configure launch perspective settings from the Perspectives preference page.

Filter matched 3 of 12 items

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

Complete the Main Tab

- Ensure that the correct project is selected
- Select the **C/C++ Application** (executable) if necessary
  - **Search Project…** will search just within the project
  - **Browse** will search anywhere on the local file system
- Select **Connect process input/output to a terminal** if desired



Run Configurations

Create, manage, and run configurations

type filter text

- C/C++ Application
  - MyHelloProject Debug
- C/C++ Remote Application
- Launch Group

Filter matched 4 of 13 items

Name: MyHelloProject Debug

Project:

MyHelloProject

Build Configuration: Debug

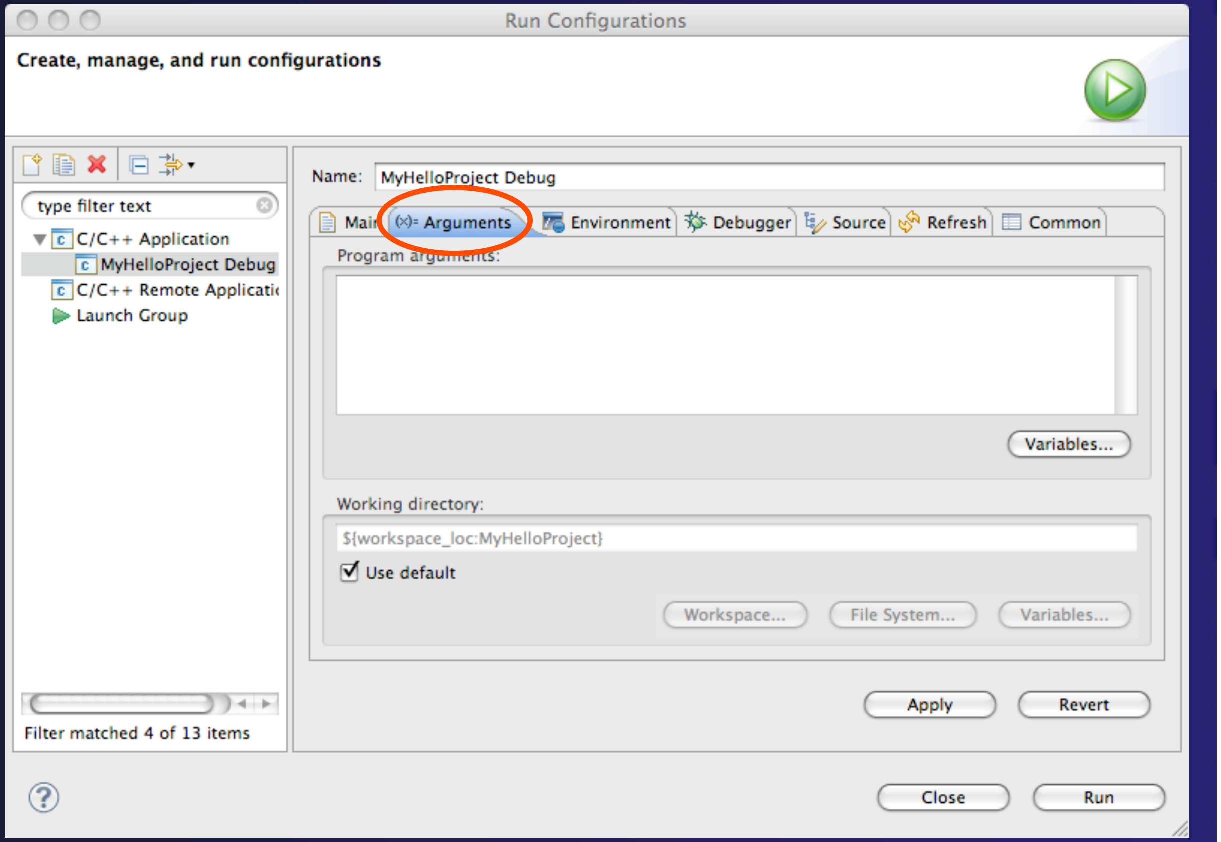
C/C++ Application:

Debug/MyHelloProject

☑ Connect process input output to a terminal.

Complete the Arguments Tab

- Enter any program arguments into the text box
- Eclipse variables can also be passed using the **Variables…** button
- Select a different working directory if desired



Run Configurations

Create, manage, and run configurations

Name: MyHelloProject Debug

Program arguments:

Working directory: ${workspace_loc:MyHelloProject}

☑ Use default

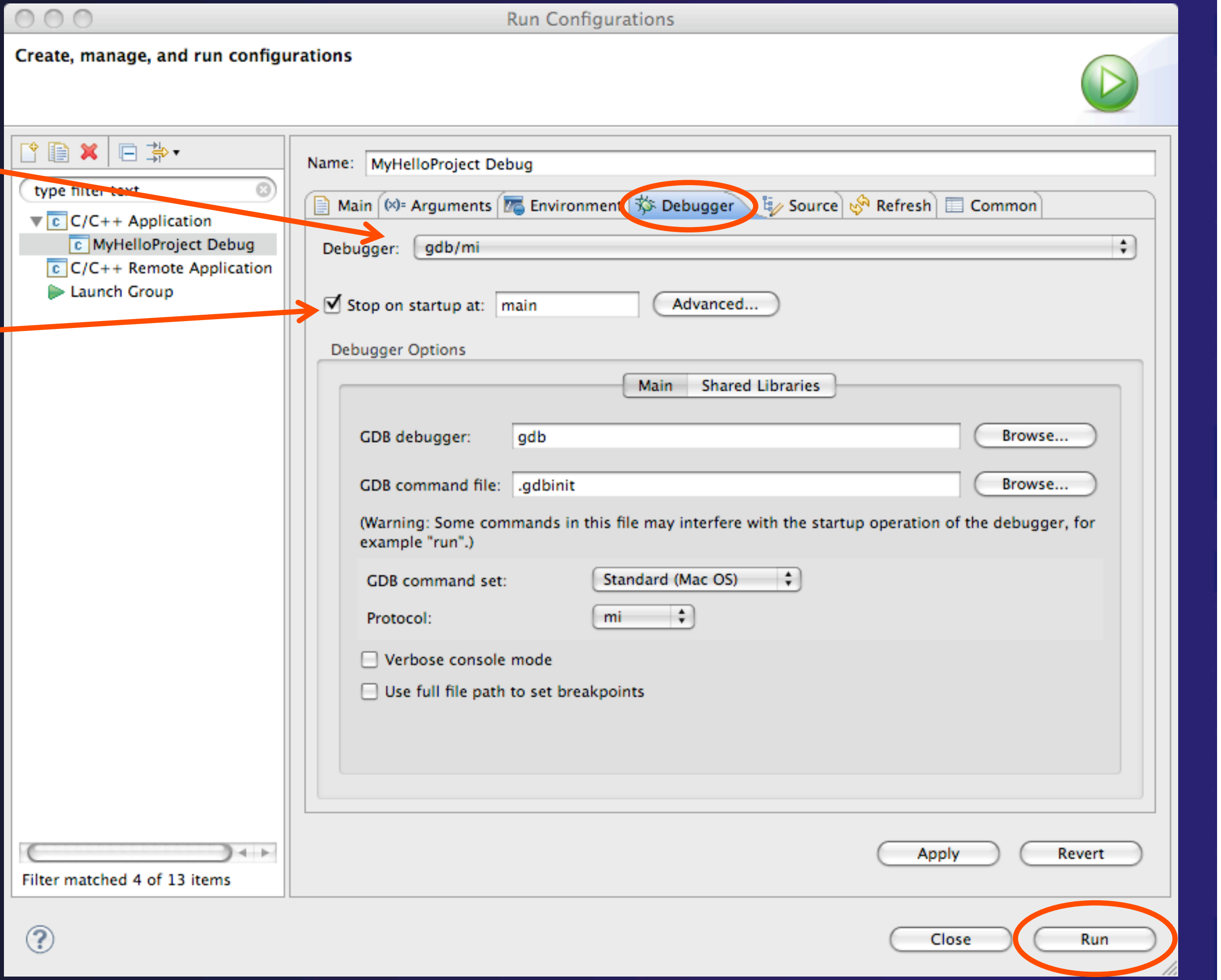
Filter matched 4 of 13 items

Complete the Debugger Tab

- Select **Debugger** tab
- Make sure **gdb/mi** is selected
- Change where the program should stop if desired
- Change any gdb-specific options if desired (advanced users only)

The information on the debugger tab will only be used for a debug launch

- Hit the Run button to launch your program



Create, manage, and run configurations

Name: MyHelloProject Debug

Debugger: gdb/mi

☑ Stop on startup at: main

Debugger Options

GDB debugger: gdb

GDB command file: .gdbinit

(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

GDB command set: Standard (Mac OS)

Protocol: mi

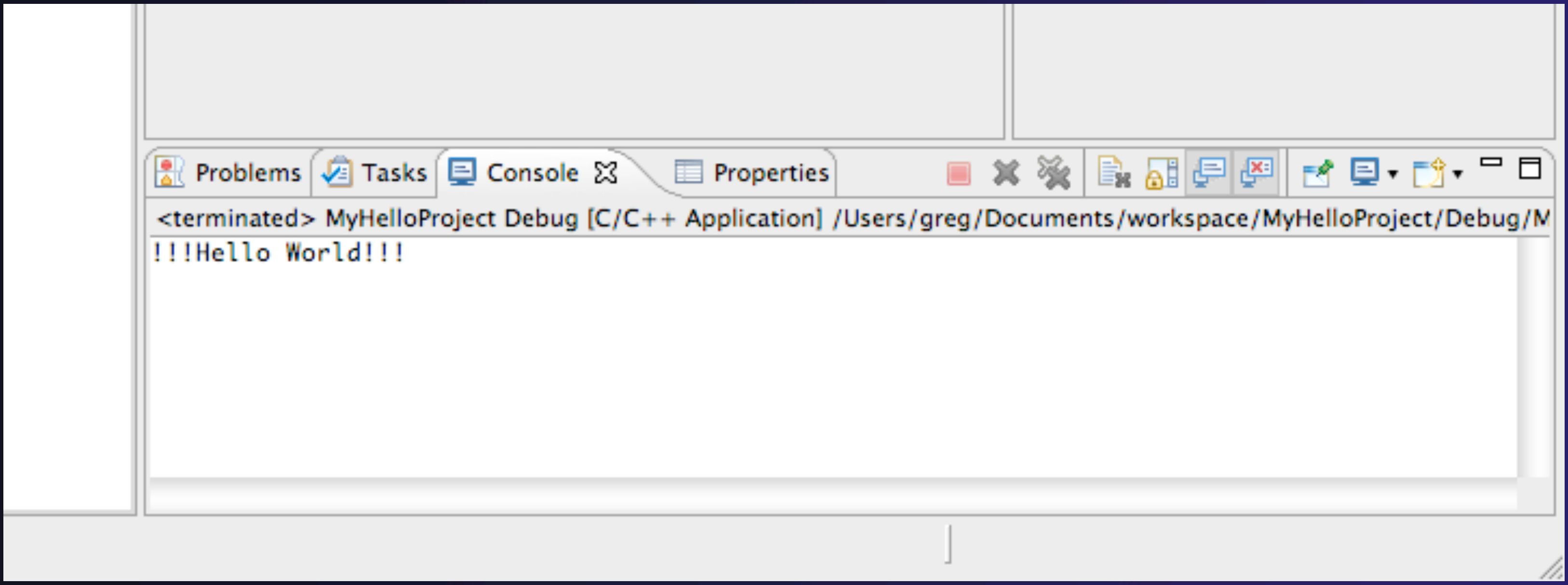
☐ Verbose console mode

☐ Use full file path to set breakpoints

Filter matched 4 of 13 items

Viewing Program Output

- When the program runs, the **Console** view should automatically become active
- Any output will be displayed in this view (stderr in red)



```
<terminated> MyHelloProject Debug [C/C++ Application] /Users/greg/Documents/workspace/MyHelloProject/Debug/M
!!!Hello World!!!
```

Module 4: Working with MPI

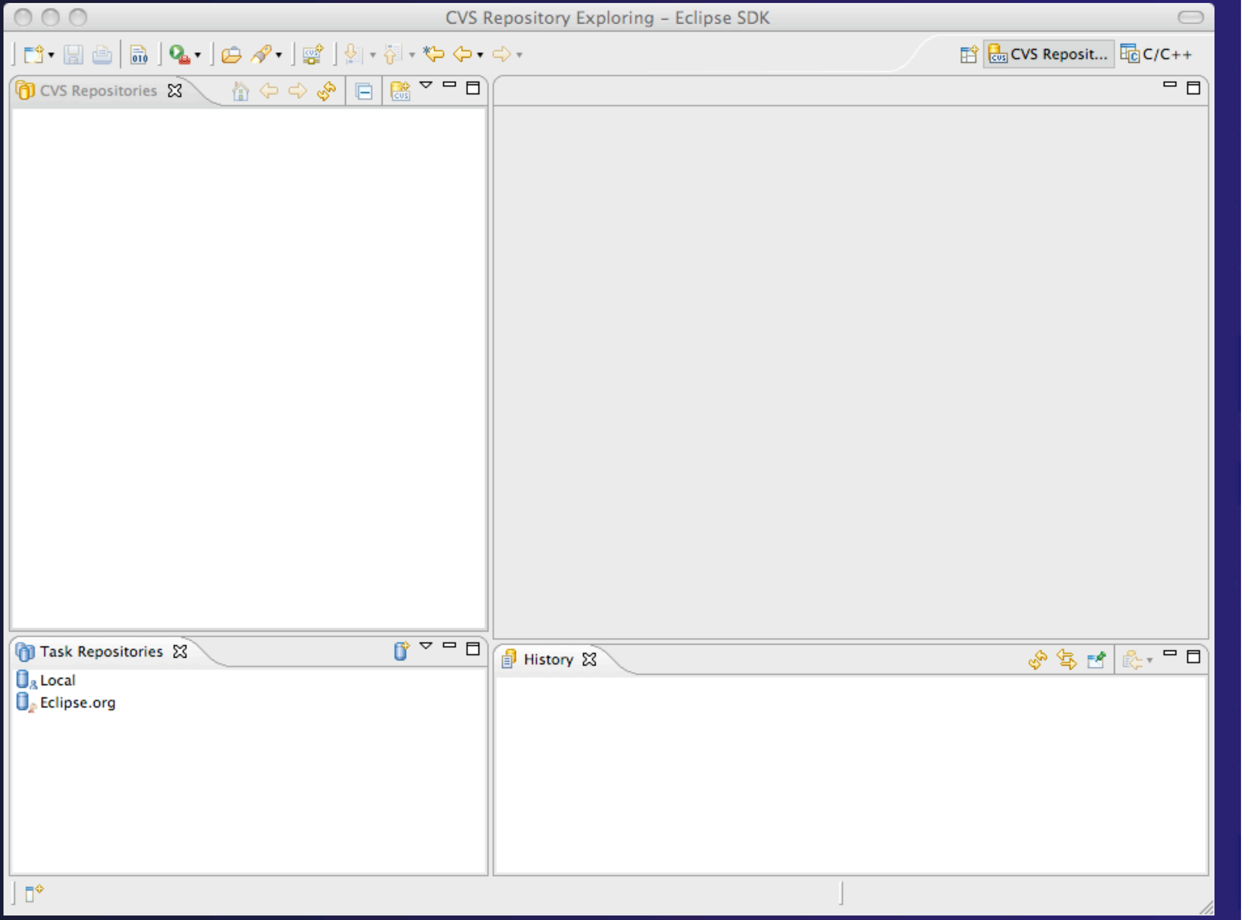
- Objective
  - Learn how to build and launch an MPI program
  - Explore some of the features to aid MPI programming
- Contents
  - Using a version control system (CVS)
  - Building with Makefiles and autoconf
  - MPI assistance features
  - Working with resource managers
  - Launching a parallel application

Creating the Project

- ✦ Configuring version control
- ✦ Checking out the source code
- ✦ Team support

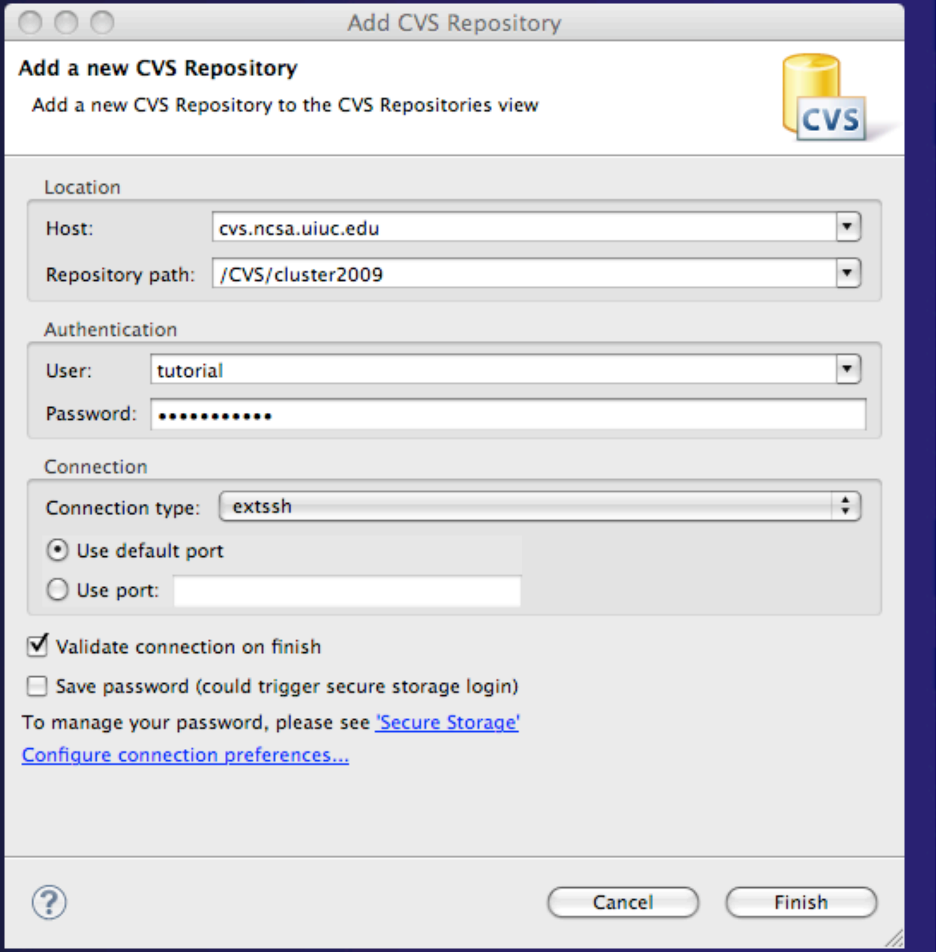
Connecting to a Repository

- Select **Window▶Open Perspective▶Other…**
- Select **CVS Repository Exploring** then **OK**



Specify Repository Location

- Right-click in the **CVS Repositories** view, then select **New▶Repository Location…**
- Set **Host** to the hostname of remote machine
- Set **Repository path** to the CVS repository path
- Fill in **Username** and **Password**
- Set **Connection type** to **extssh** to use an ssh connection
- Check **Save password** if you wish to save the password
- Select **Finish**


Add CVS Repository

Add a new CVS Repository

Add a new CVS Repository to the CVS Repositories view

Location

Host: cvs.ncsa.uiuc.edu

Repository path: /CVS/cluster2009

Authentication

User: tutorial

Password: •••••••••••

Connection

Connection type: extssh

● Use default port

○ Use port:

☑ Validate connection on finish

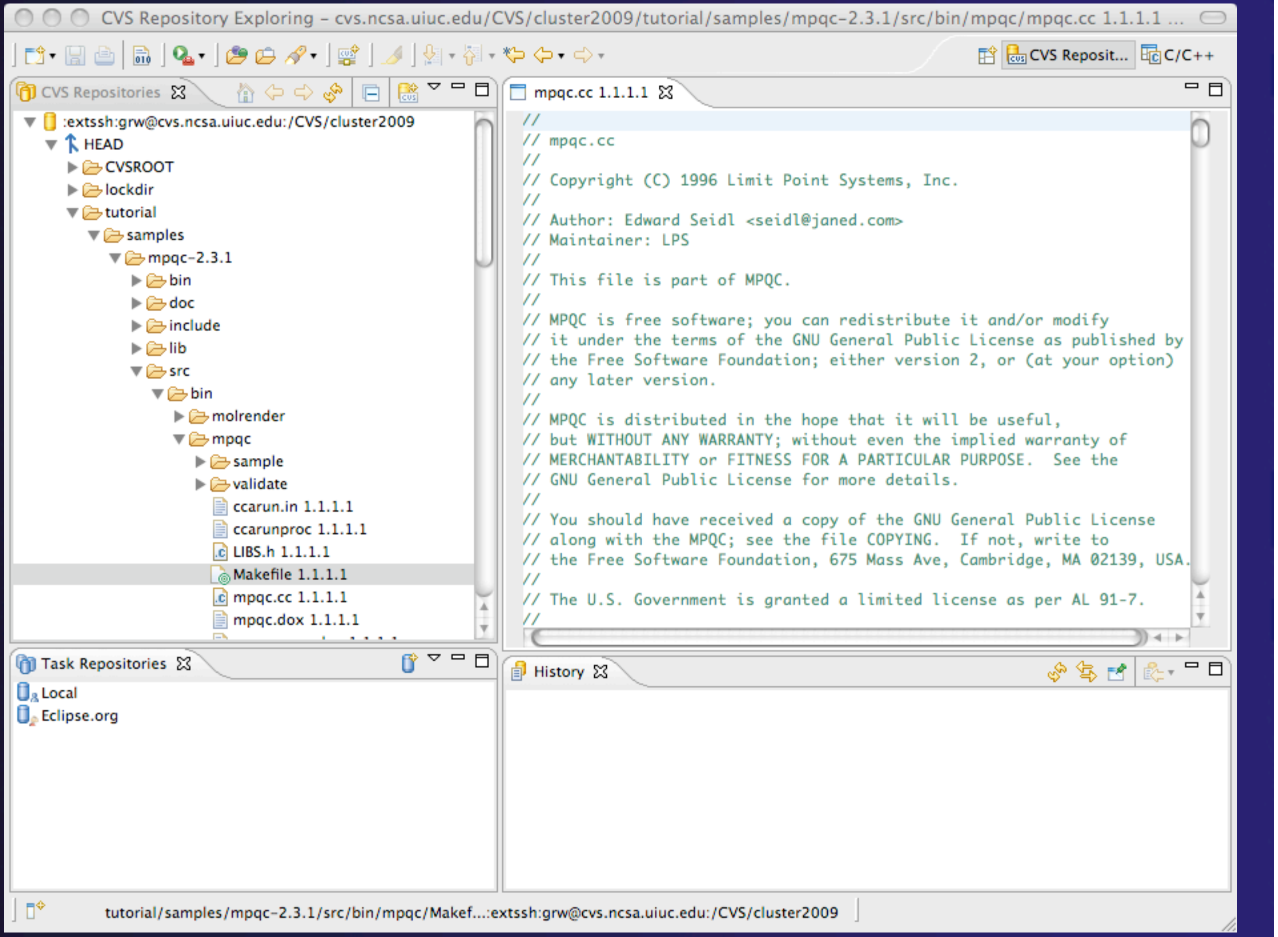
□ Save password (could trigger secure storage login)

To manage your password, please see 'Secure Storage'

Configure connection preferences...

CVS Repository Exploring

- Open the repository in the **CVS Repository** view
- Open **HEAD** to view files and folders in the CVS head
- Open **Branches** or **Versions** to view CVS branches or versions respectively
- Right-click on the repository and select **Refresh Branches…** to see all branches and versions


CVS Repository Exploring – cvs.ncsa.uiuc.edu/CVS/cluster2009/tutorial/samples/mpqc-2.3.1/src/bin/mpqc/mpqc.cc 1.1.1.1 ...
CVS Reposit...
C/C++
CVS Repositories
mpqc.cc 1.1.1.1
Task Repositories
History

```
:extssh:grw@cvs.ncsa.uiuc.edu:/CVS/cluster2009
  HEAD
    CVSROOT
    lockdir
    tutorial
      samples
        mpqc-2.3.1
          bin
          doc
          include
          lib
          src
            bin
              molrender
              mpqc
                sample
                validate
                ccarun.in 1.1.1.1
                ccarunproc 1.1.1.1
                LIBS.h 1.1.1.1
                Makefile 1.1.1.1
                mpqc.cc 1.1.1.1
                mpqc.dox 1.1.1.1
```

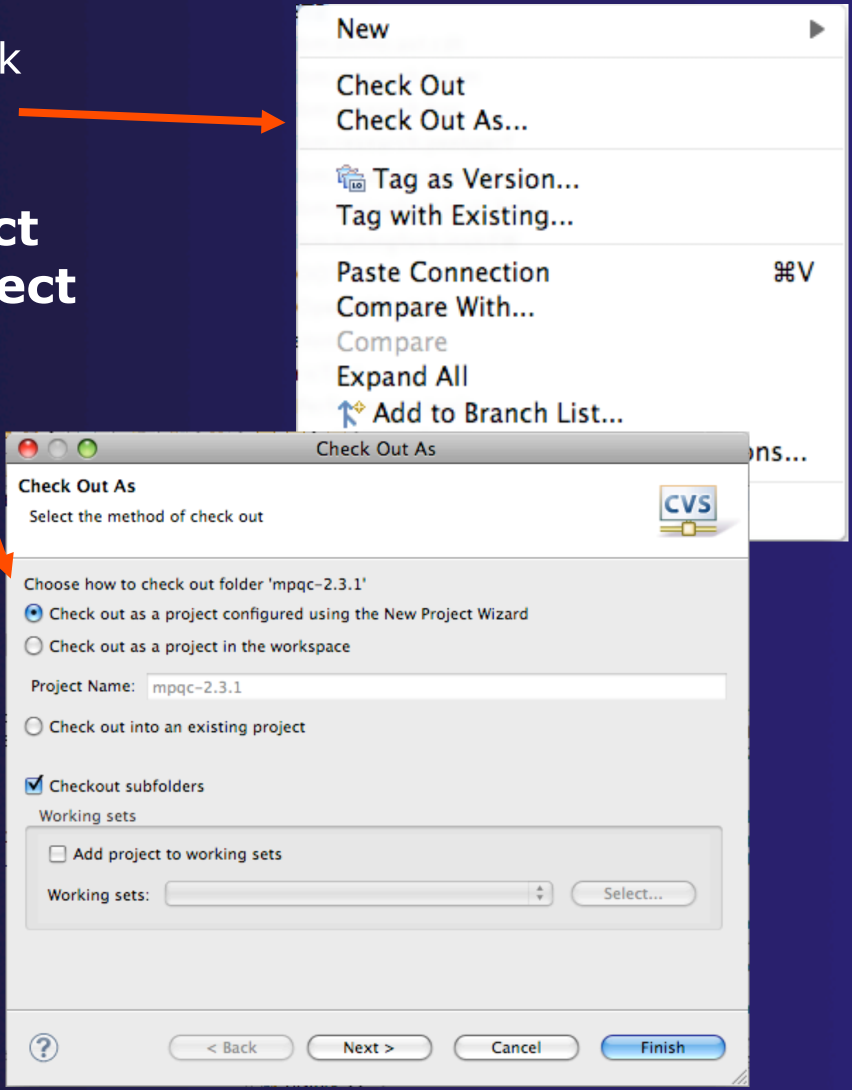
```
//
// mpqc.cc
//
// Copyright (C) 1996 Limit Point Systems, Inc.
//
// Author: Edward Seidl <seidl@janed.com>
// Maintainer: LPS
//
// This file is part of MPQC.
//
// MPQC is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation; either version 2, or (at your option)
// any later version.
//
// MPQC is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with the MPQC; see the file COPYING.  If not, write to
// the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
//
// The U.S. Government is granted a limited license as per AL 91-7.
//
```

Local
Eclipse.org

tutorial/samples/mpqc-2.3.1/src/bin/mpqc/Makef...:extssh:grw@cvs.ncsa.uiuc.edu:/CVS/cluster2009

Check out as an Eclipse Project

- In CVS Repositories view, right-click on project and select **Project▶Check out As…**
- Make sure **Check out as a project configured using the New Project Wizard** is selected
- Leave **Checkout subfolders** checked
- Select **Finish**



Check Out As

Check Out As

Select the method of check out

Choose how to check out folder 'mpqc-2.3.1'

● Check out as a project configured using the New Project Wizard

○ Check out as a project in the workspace

Project Name: mpqc-2.3.1

○ Check out into an existing project

☑ Checkout subfolders

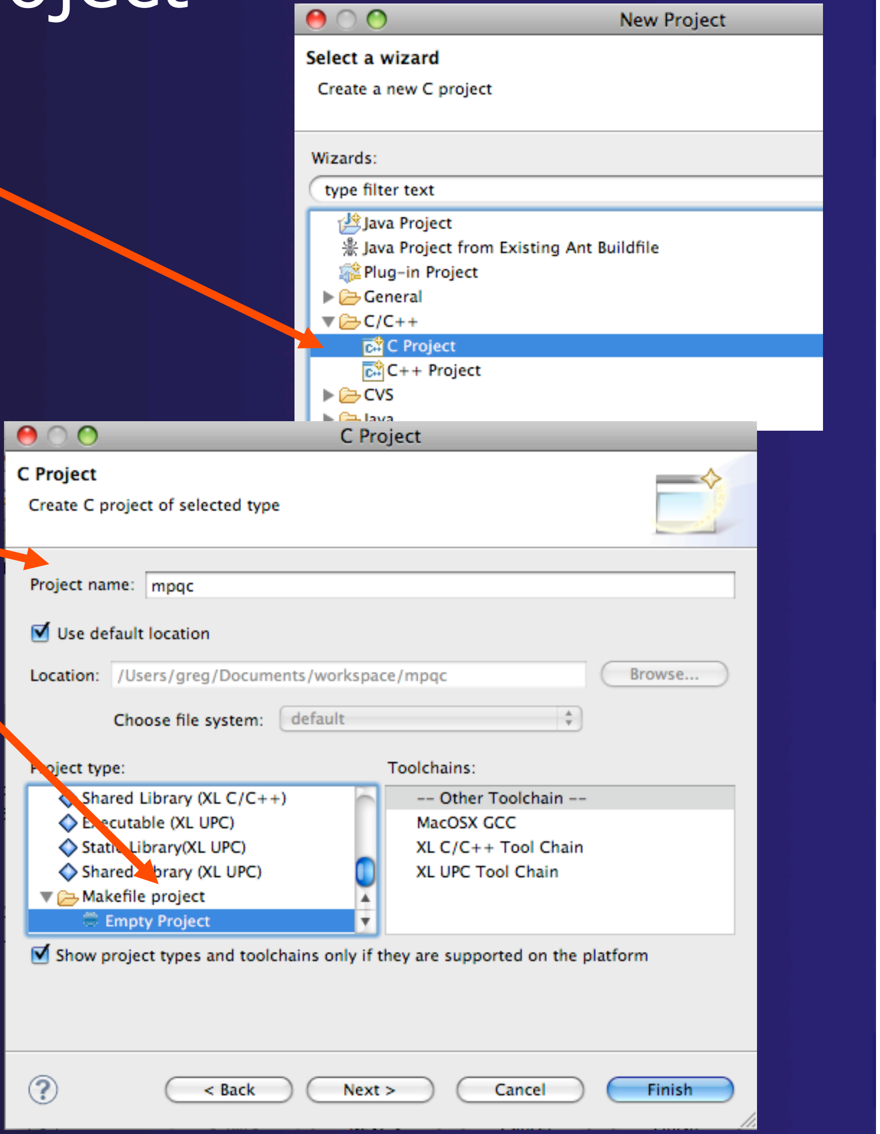
Working sets

□ Add project to working sets

Working sets:

Create a C Project

- The **New Project Wizard** is used to create a C project
- Enter **Project name**
- Under **Project Types**, select **Makefile project▶Empty Project**
  - Ensures that CDT will use existing makefiles
- Select **Finish**
- When prompted to switch to the **C/C++ Perspective**, select **Yes**



New Project

Select a wizard

Create a new C project

Wizards:

type filter text

C Project

C Project

Create C project of selected type

Project name: mpqc

☑ Use default location

Location: /Users/greg/Documents/workspace/mpqc

Choose file system: default

Project type:

Toolchains:

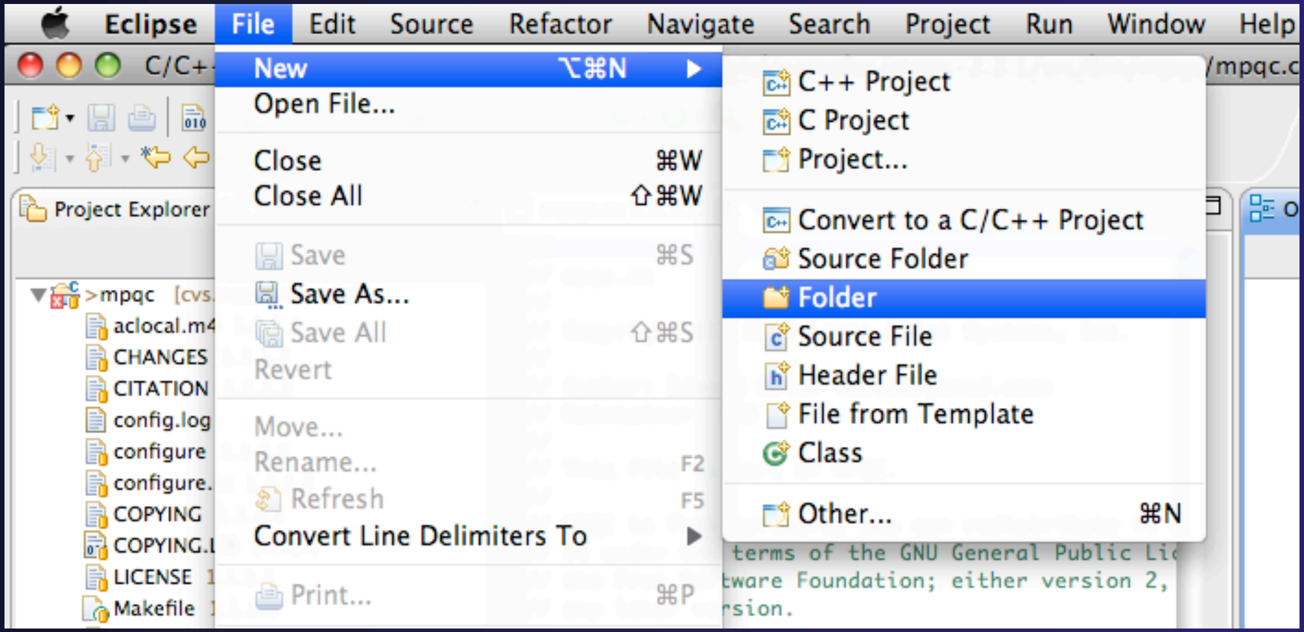
☑ Show project types and toolchains only if they are supported on the platform

Building the Application

- Configuring the project build directory
- Generating Makefiles
- Creating a Make Target
- Running the build

Create a **build** directory

- This program requires a separate build directory
- Select the project in the **Project Explorer** view
- From the **File** menu, select elect **New▶Folder…**
- Make sure the parent folder is correct
- Enter "build" as the folder name
- Click **Finish**

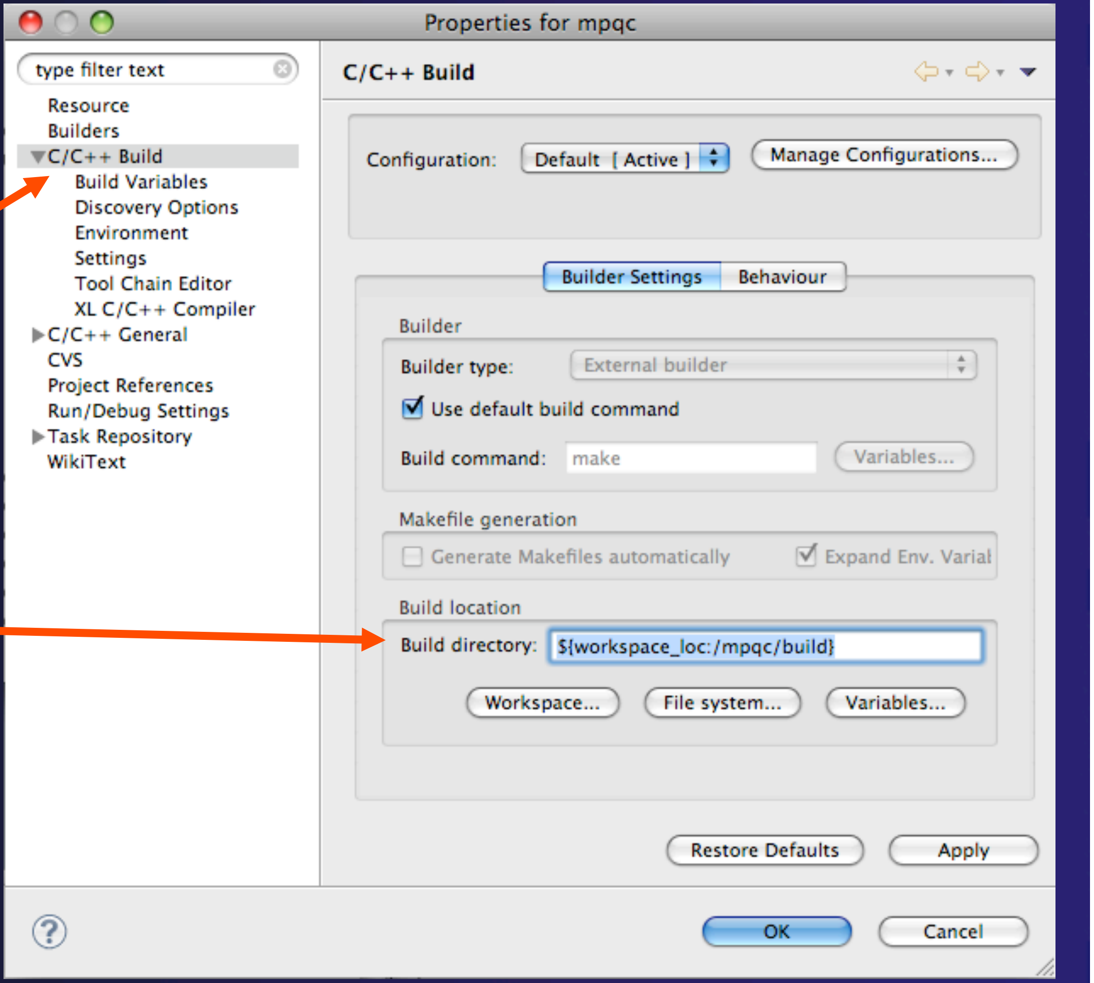


Makefile Project

- Similar to managed project, but uses custom Makefile (or other script) to control build
- User can specify command that will be used to initiate build
- Can also specify the directory in which the build will take place
- “Make targets” are used to control type of build
- Can switch between managed and un-managed project

Makefile Project Properties

- Right click on project in **Project Explorer** to bring up properties
- Click on **C/C++ Build** for the build settings
- Can change build command if desired
- Change the **Build location** to the **build** directory in the project



Configuration: Default [Active]

Builder type: External builder

☑ Use default build command

Build command: make

□ Generate Makefiles automatically ☑ Expand Env. Varial

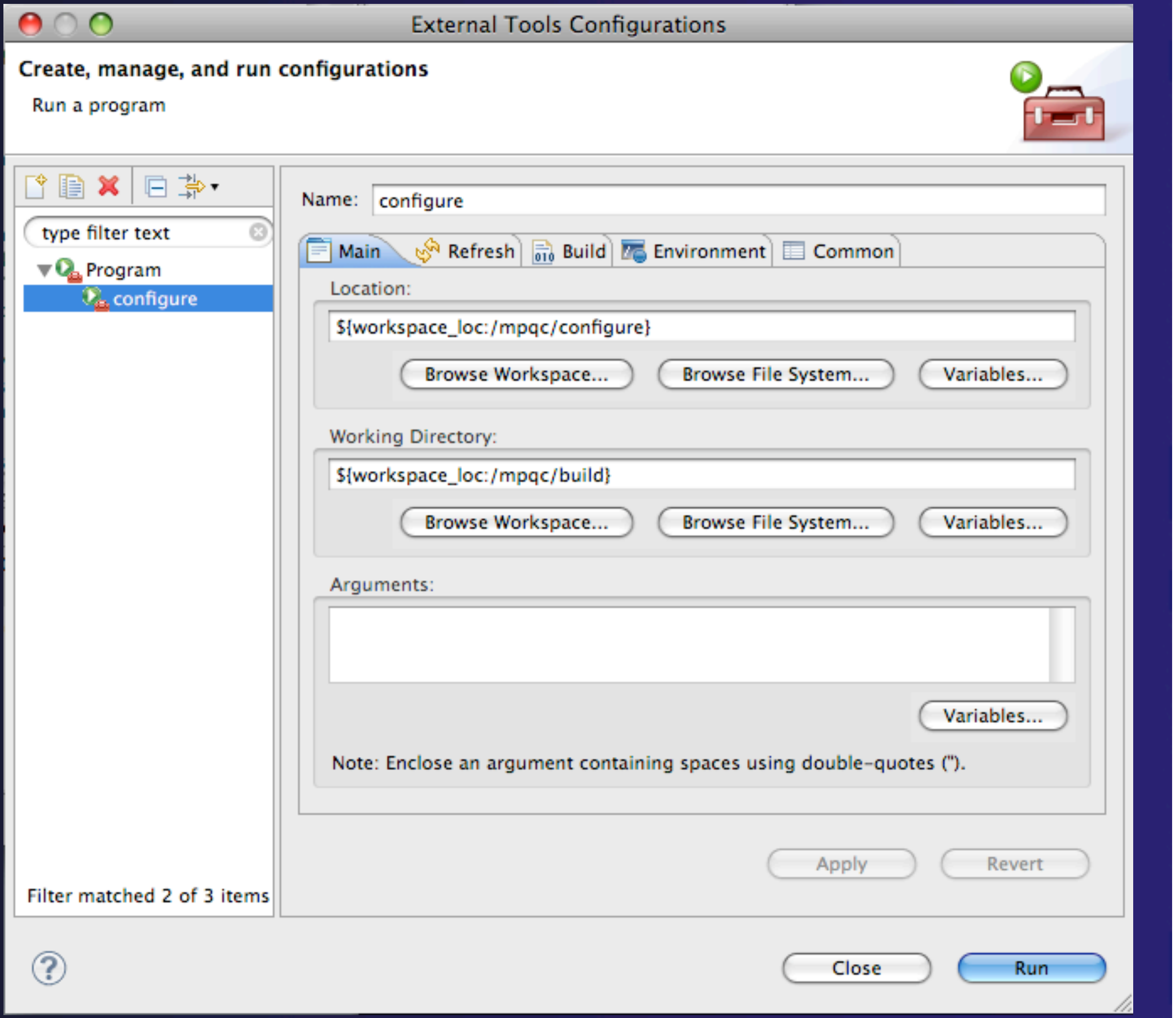
Build directory: ${workspace_loc:/mpqc/build}

About Makefiles and autoconf

- `Autoconf` is a GNU utility often used to create Makefiles for open source projects
  - Used to generate a `configure` script
  - `Configure` is run to generate a Makefile that suits a particular system configuration
  - Normally only needs to be run once, unless the build process needs to be changed
- Run `configure` using two methods:
  - Manually from an external shell
  - By creating an **External Tools Launch Configuration**
- Must refresh **Project Explorer** whenever file system is modified outside of Eclipse, such as after running `configure`

Generate the Makefiles

- From the **Run** menu, select **External Tools▶External Tools Configurations…**
- Create a new **Program**
- For **Location**, click **Browse Workspace…** and find the configure script
- For **Working Directory**, click **Browse Workspace…** and select the **build** directory in the project
- Click **Run** and you should see output in the **Console** view
- In **Project Explorer**, right-click and select **Refresh** to see the new files that have been created



External Tools Configurations

Create, manage, and run configurations

Run a program

Name: configure

Location: ${workspace_loc:/mpqc/configure}

Working Directory: ${workspace_loc:/mpqc/build}

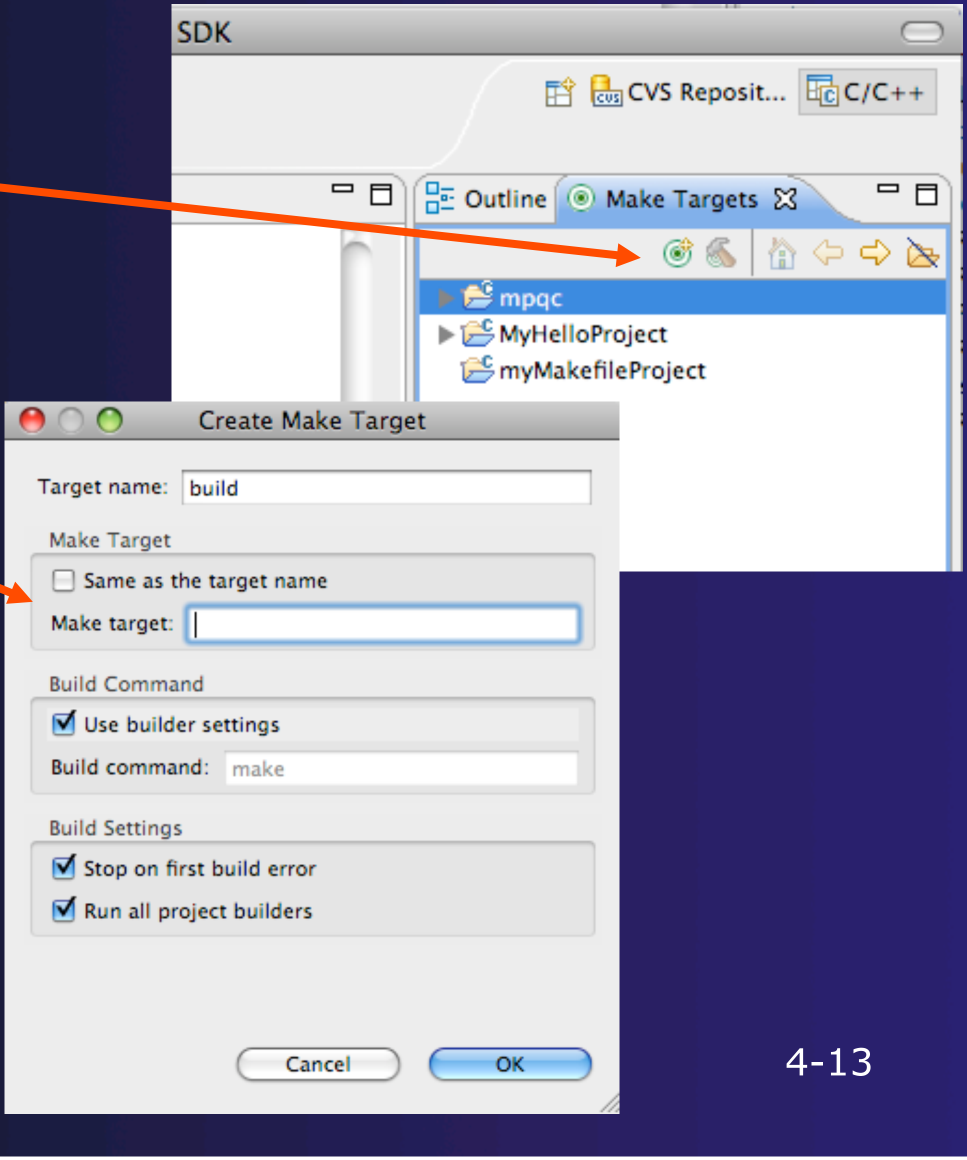
Arguments:

Note: Enclose an argument containing spaces using double-quotes (").

Filter matched 2 of 3 items

Create a Make Target

- Select the project in **Make Targets** view
- Click on **New Make Target** icon
- Enter the desired name of the target
- Unselect Same as the target name and delete "build"
  - This will run the "make" command with no arguments
- Select **OK**



Create Make Target

Target name: build

Make Target

☐ Same as the target name

Make target:

Build Command

☑ Use builder settings

Build command: make

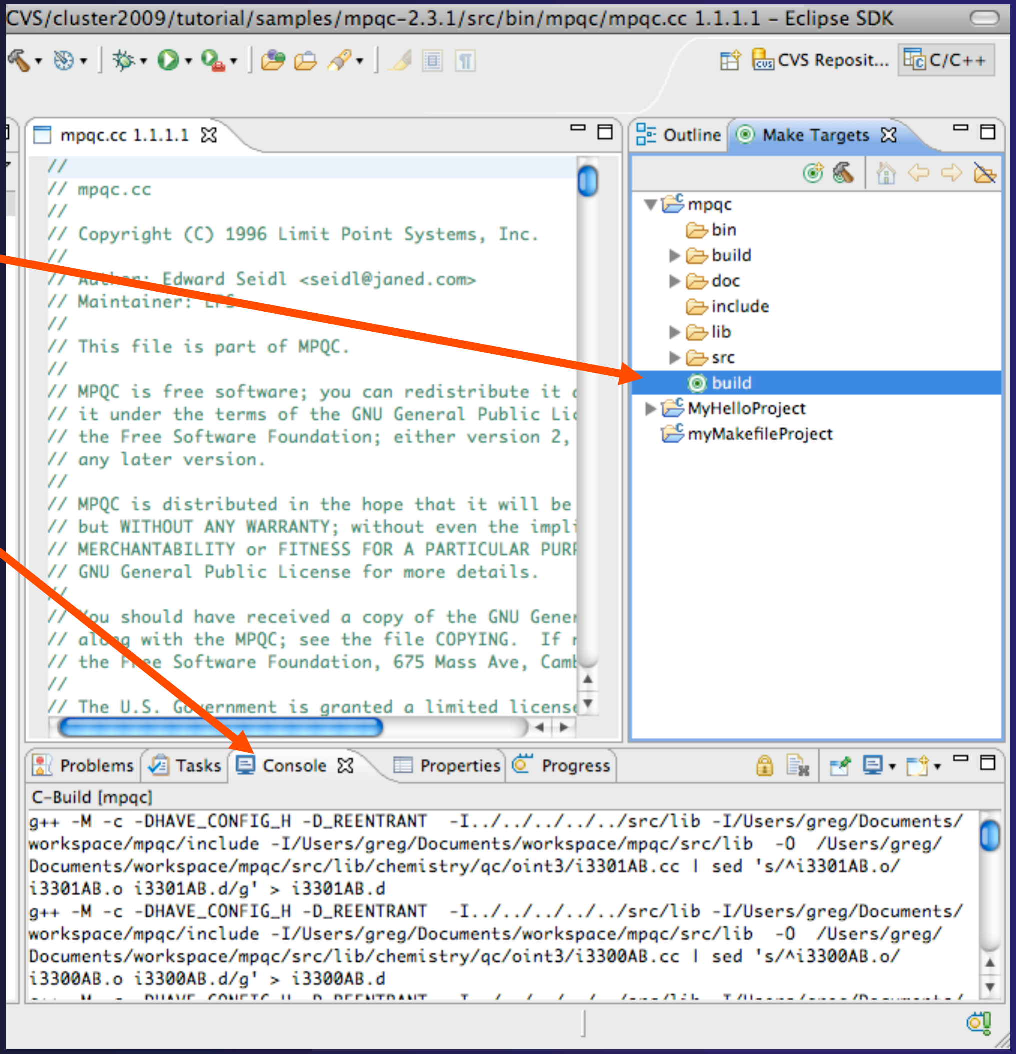
Build Settings

☑ Stop on first build error

☑ Run all project builders

Running the Build

- Open the project in the **Make Targets** view to see the **build** target
- Double-click on the **build** target to initiate the build
- Output from the build will be visible in the **Console** view



```
//
// mpqc.cc
//
// Copyright (C) 1996 Limit Point Systems, Inc.
//
// Author: Edward Seidl <seidl@janed.com>
// Maintainer: LPS
//
// This file is part of MPQC.
//
// MPQC is free software; you can redistribute it a
// it under the terms of the GNU General Public Lic
// the Free Software Foundation; either version 2,
// any later version.
//
// MPQC is distributed in the hope that it will be
// but WITHOUT ANY WARRANTY; without even the impli
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURP
// GNU General Public License for more details.
//
// You should have received a copy of the GNU Gener
// along with the MPQC; see the file COPYING.  If n
// the Free Software Foundation, 675 Mass Ave, Camb
//
// The U.S. Government is granted a limited license
```

C-Build [mpqc]

```
g++ -M -c -DHAVE_CONFIG_H -D_REENTRANT  -I../../../../../src/lib -I/Users/greg/Documents/
workspace/mpqc/include -I/Users/greg/Documents/workspace/mpqc/src/lib  -O  /Users/greg/
Documents/workspace/mpqc/src/lib/chemistry/qc/oint3/i3301AB.cc | sed 's/^i3301AB.o/
i3301AB.o i3301AB.d/g' > i3301AB.d
g++ -M -c -DHAVE_CONFIG_H -D_REENTRANT  -I../../../../../src/lib -I/Users/greg/Documents/
workspace/mpqc/include -I/Users/greg/Documents/workspace/mpqc/src/lib  -O  /Users/greg/
Documents/workspace/mpqc/src/lib/chemistry/qc/oint3/i3300AB.cc | sed 's/^i3300AB.o/
i3300AB.o i3300AB.d/g' > i3300AB.d
```

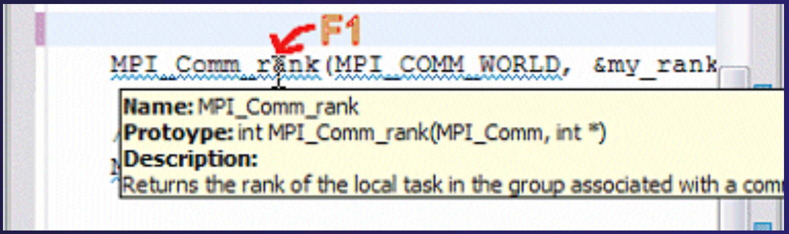
MPI Assistance Tools

Added by PLDT (Parallel Lang. Dev. Tools) feature of PTP

- MPI Context sensitive help
- MPI artifact locations
- MPI barrier analysis
- MPI templates

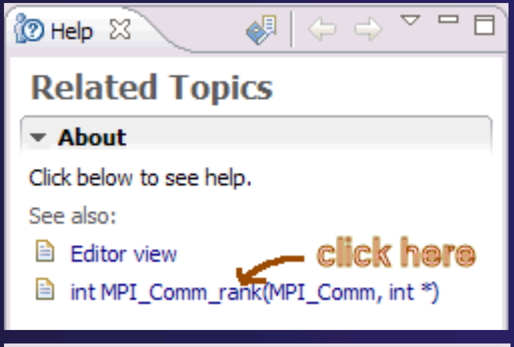
Context Sensitive Help

- 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



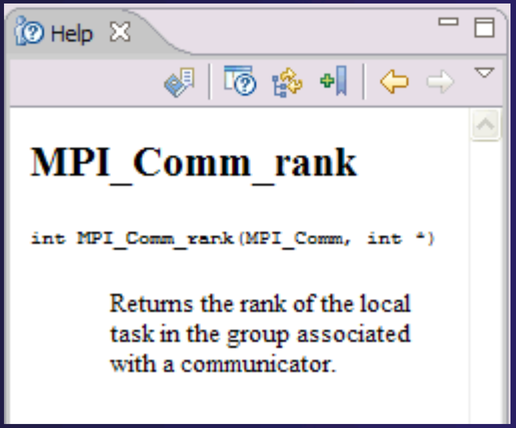
Related Topics

About

Click below to see help.

See also:

- Editor view
- int MPI_Comm_rank(MPI_Comm, int *)



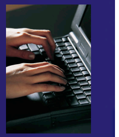
MPI_Comm_rank

```
int MPI_Comm_rank(MPI_Comm, int *)
```

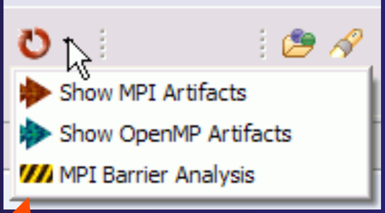
Returns the rank of the local task in the group associated with a communicator.

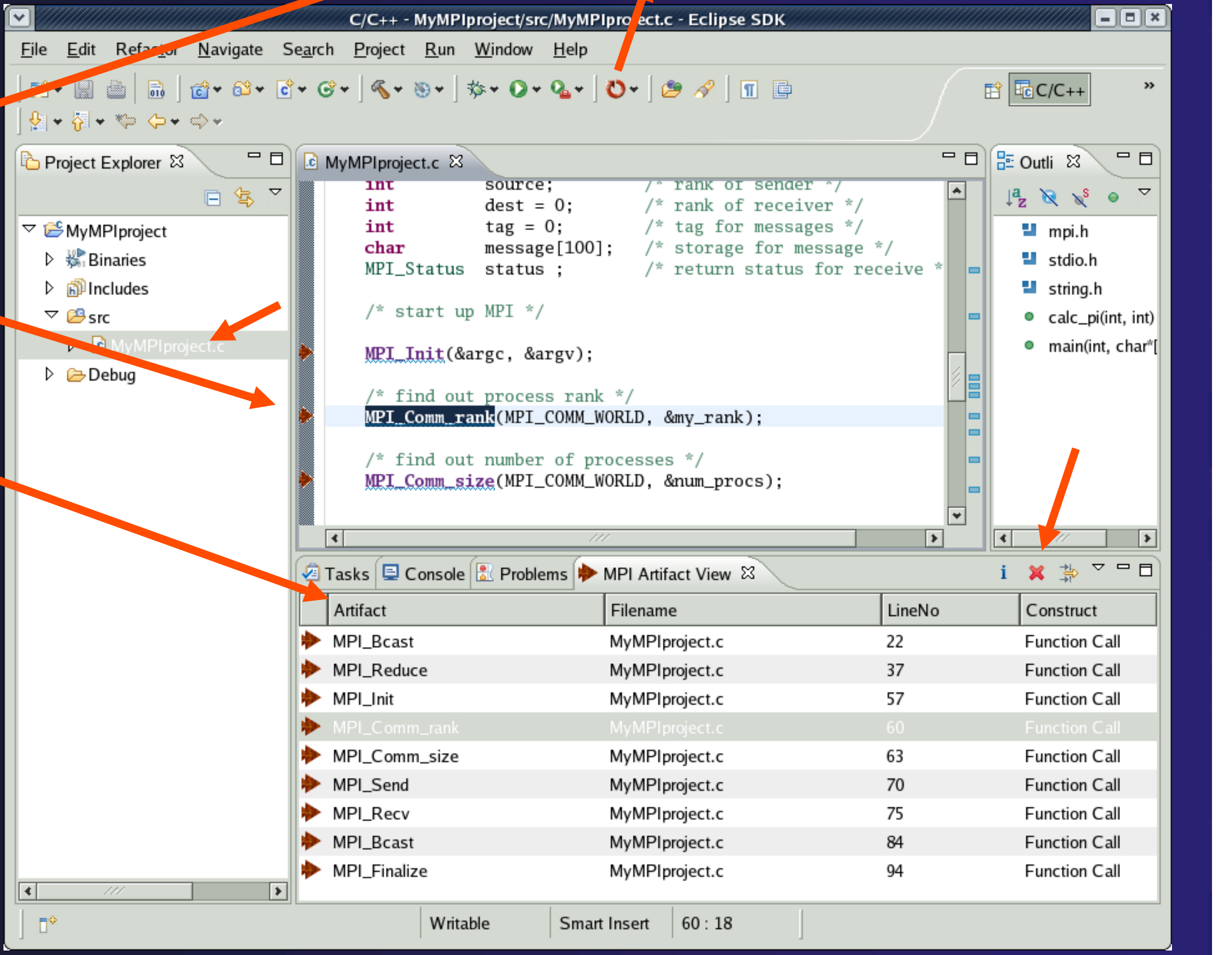
Some special info has been added for MPI APIs.

Show MPI Artifacts



- ✦ Select source file; Run analysis by clicking on drop-down menu next to the analysis button and selecting **Show MPI Artifacts**
- ✦ Markers indicate the location of artifacts in editor
- ✦ In **MPI Artifact View** sort by any column (click on col. heading)
- ✦ Navigate to source code line by double-clicking on the artifact
- ✦ Run the analysis on another file and its markers will be added to the view
- ✦ Remove markers via ✖

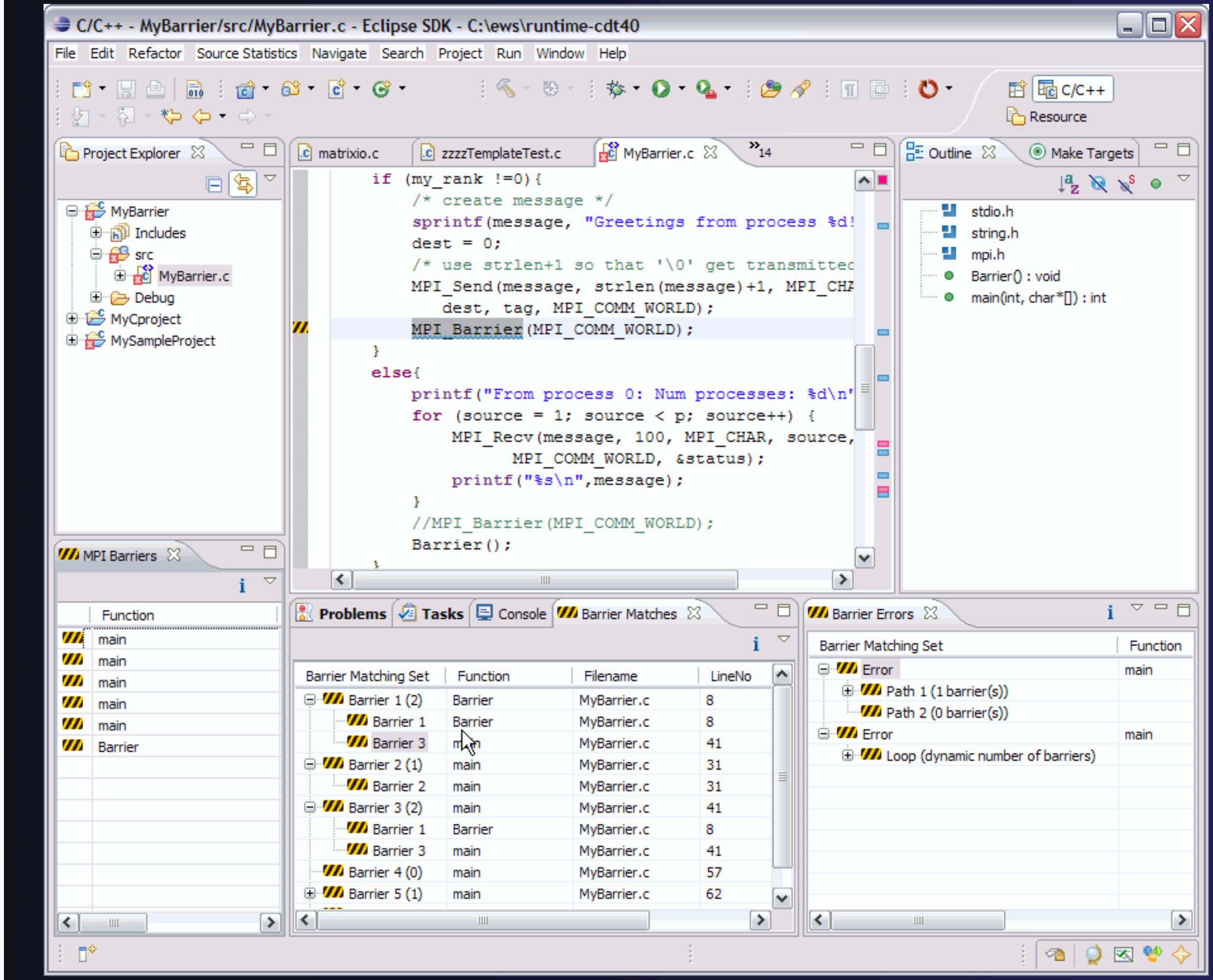




```
int         source;        /* rank of sender */
int         dest = 0;      /* rank of receiver */
int         tag = 0;       /* tag for messages */
char        message[100];  /* storage for message */
MPI_Status  status ;       /* return status for receive *

/* start up MPI */

MPI_Init(&argc, &argv);

/* find out process rank */
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

/* find out number of processes */
MPI_Comm_size(MPI_COMM_WORLD, &num_procs);
```

Artifact	Filename	LineNo	Construct
MPI_Bcast	MyMPIproject.c	22	Function Call
MPI_Reduce	MyMPIproject.c	37	Function Call
MPI_Init	MyMPIproject.c	57	Function Call
MPI_Comm_rank	MyMPIproject.c	60	Function Call
MPI_Comm_size	MyMPIproject.c	63	Function Call
MPI_Send	MyMPIproject.c	70	Function Call
MPI_Recv	MyMPIproject.c	75	Function Call
MPI_Bcast	MyMPIproject.c	84	Function Call
MPI_Finalize	MyMPIproject.c	94	Function Call

MPI Barrier Analysis



```
    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

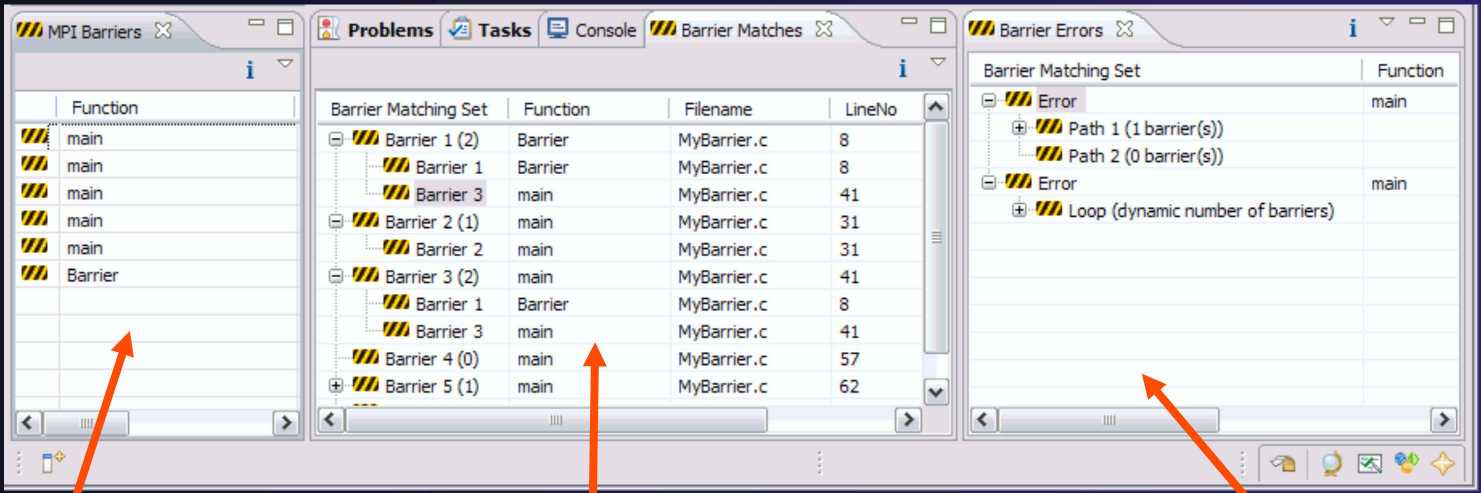
Barrier Matching Set	Function
⊟ Error	main
⊞ Path 1 (1 barrier(s))	
Path 2 (0 barrier(s))	
⊟ Error	main
⊞ Loop (dynamic number of barriers)	

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 - views



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)	

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

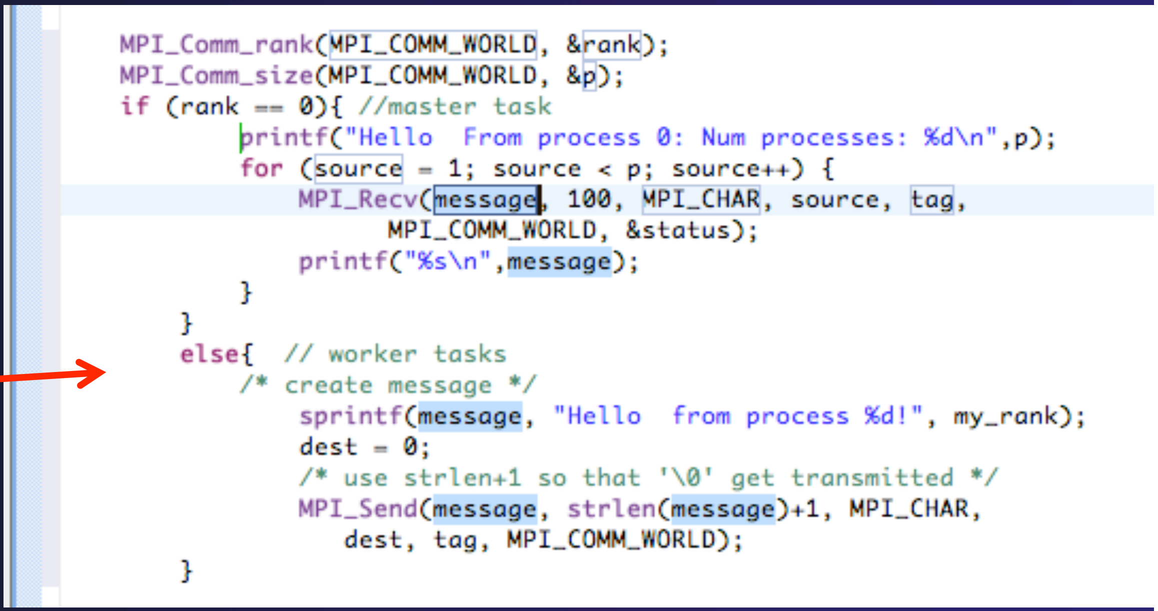
Groups barriers that match together in a barrier set – all processes must go through a barrier in the set to prevent a deadlock

Barrier Errors view

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

MPI Templates

- ✦Allows quick entry of common patterns in MPI programming
- ✦Example: MPI send-receive
- ✦Enter: `mpisr` <ctrl-space>
- ✦Expands to



```
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);
    }
```

- ✦Eclipse preferences: add more!
    - ✦C/C++ > Editor > Templates
- ✦Extend to other common patterns

Running the Program

- Terminology
- PTP Runtime Perspective
- Resource Managers
- Launch Configurations

Terminology

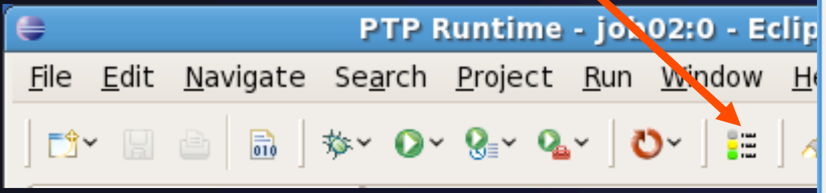
- The **PTP Runtime** perspective is provided for monitoring and controlling applications
- Some terminology
  - **Resource manager** - Corresponds to an instance of a resource management system (e.g. a job scheduler). You can have multiple resource mangers connected to different machines.
  - **Queue** - A queue of pending jobs
  - **Job** – A single run of a parallel application
  - **Machine** - A parallel computer system
  - **Node** - Some form of computational resource
  - **Process** - An execution unit (may be multiple threads of execution)

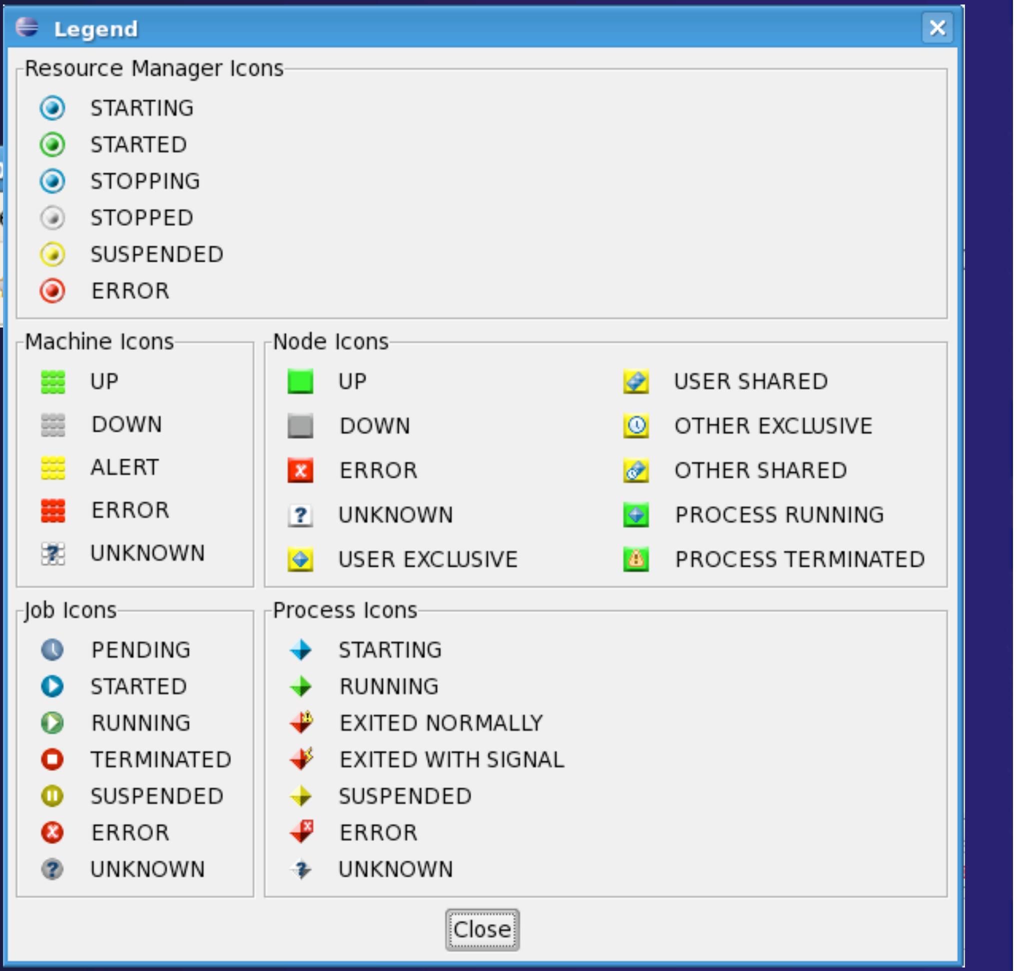
Resource Managers

- PTP uses the term “resource manager” to refer to any subsystem that controls the resources required for launching a parallel job.
- Examples:
  - Job scheduler (e.g. LoadLeveler)
  - Open MPI Runtime Environment (ORTE)
- Each resource manager controls one target system
- Resource Managers can be local or remote

About PTP Icons

- Open using legend icon in toolbar





Resource Manager Icons

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

Machine Icons

- UP
- DOWN
- ALERT
- ERROR
- UNKNOWN

Node Icons

- UP
- DOWN
- ERROR
- UNKNOWN
- USER EXCLUSIVE
- USER SHARED
- OTHER EXCLUSIVE
- OTHER SHARED
- PROCESS RUNNING
- PROCESS TERMINATED

Job Icons

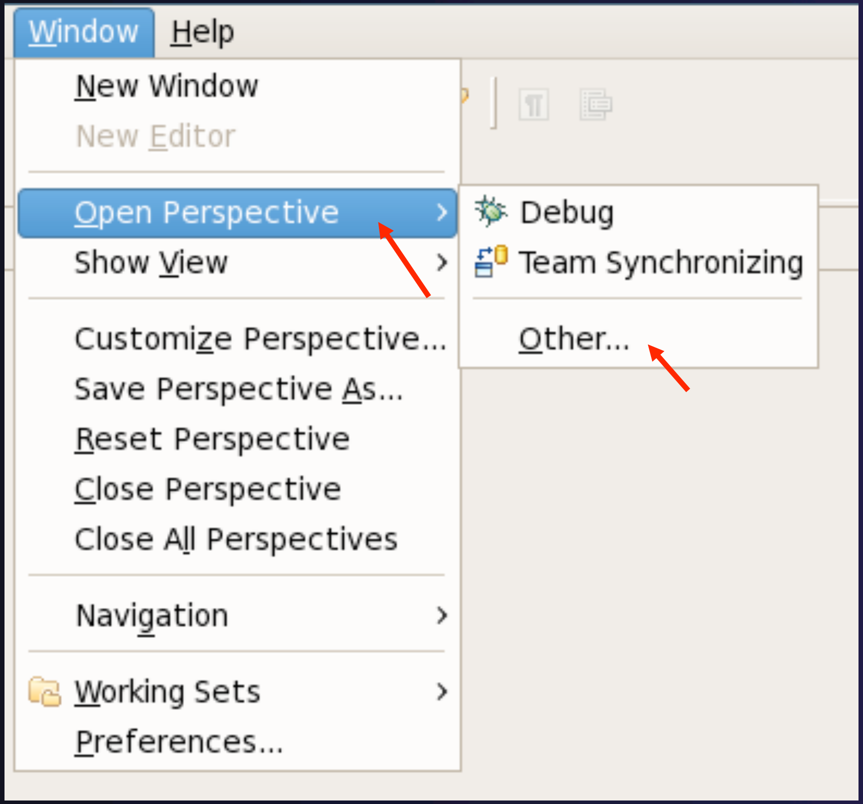
- PENDING
- STARTED
- RUNNING
- TERMINATED
- SUSPENDED
- ERROR
- UNKNOWN

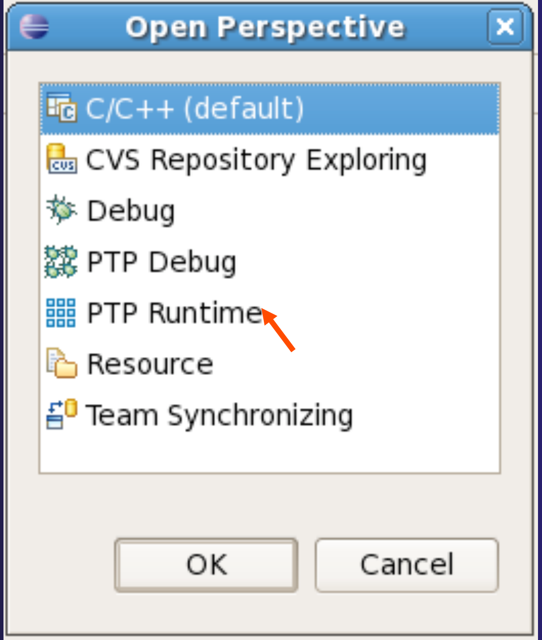
Process Icons

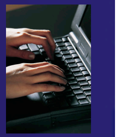
- STARTING
- RUNNING
- EXITED NORMALLY
- EXITED WITH SIGNAL
- SUSPENDED
- ERROR
- UNKNOWN

Open PTP Runtime Perspective

Window > Open Perspective > Other…

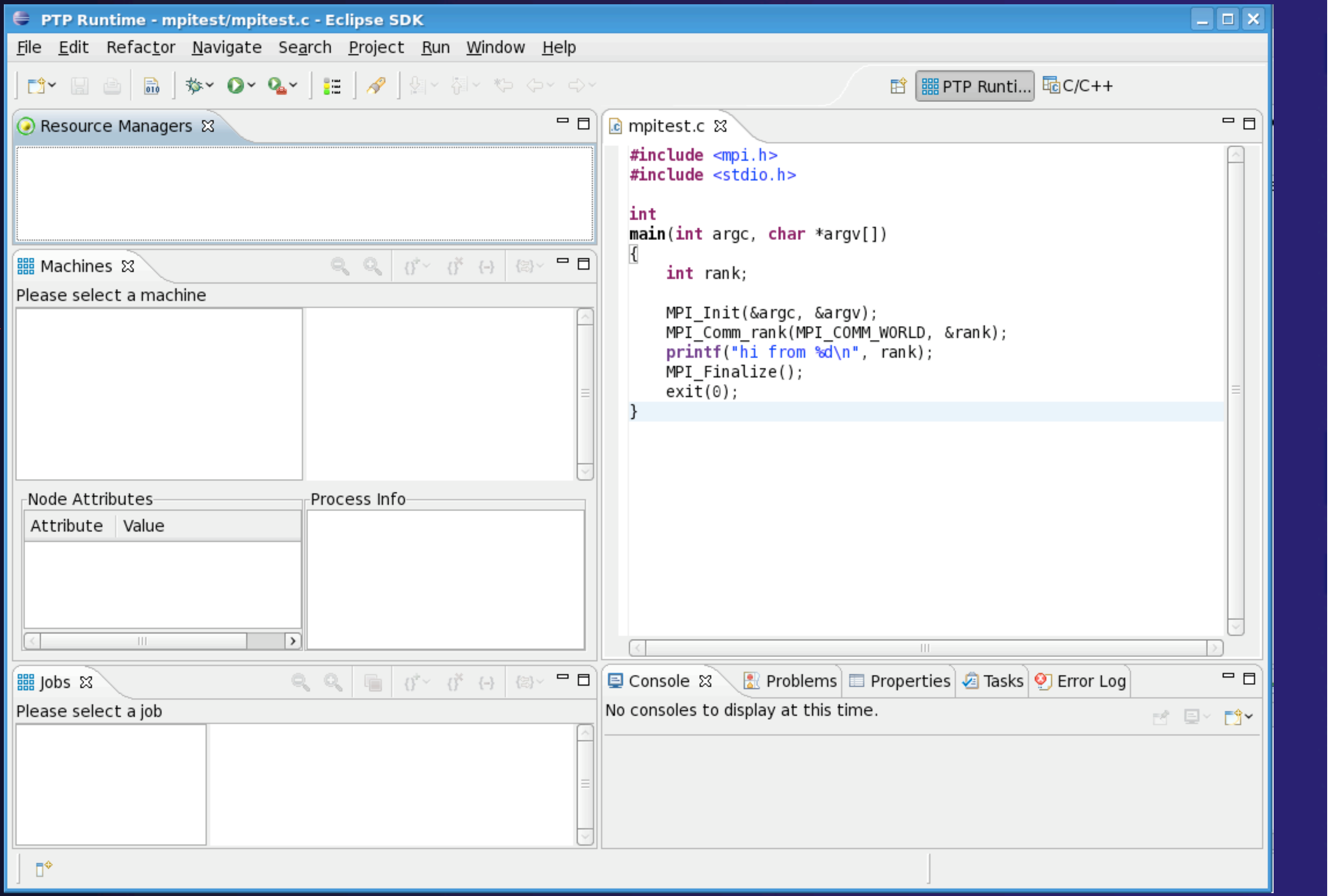






PTP Runtime Perspective

- Resource managers view
- Machines view
- Node details view
- Jobs view

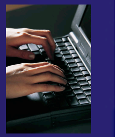


Please select a machine

Please select a job

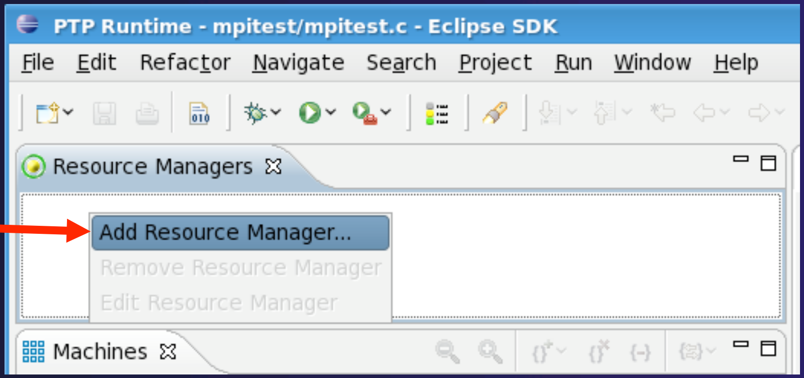
```
#include <mpi.h>
#include <stdio.h>

int
main(int argc, char *argv[])
{
    int rank;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    printf("hi from %d\n", rank);
    MPI_Finalize();
    exit(0);
}
```

No consoles to display at this time.



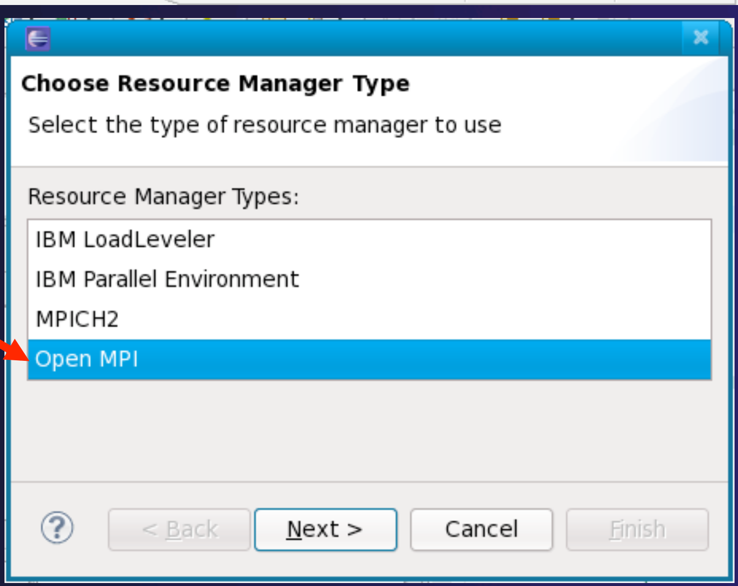
Adding a Resource Manager

- Right-click in Resource Managers view and select **Add Resource Manager**



- Choose the **Open MPI Resource Manager Type**

- Select **Next>**



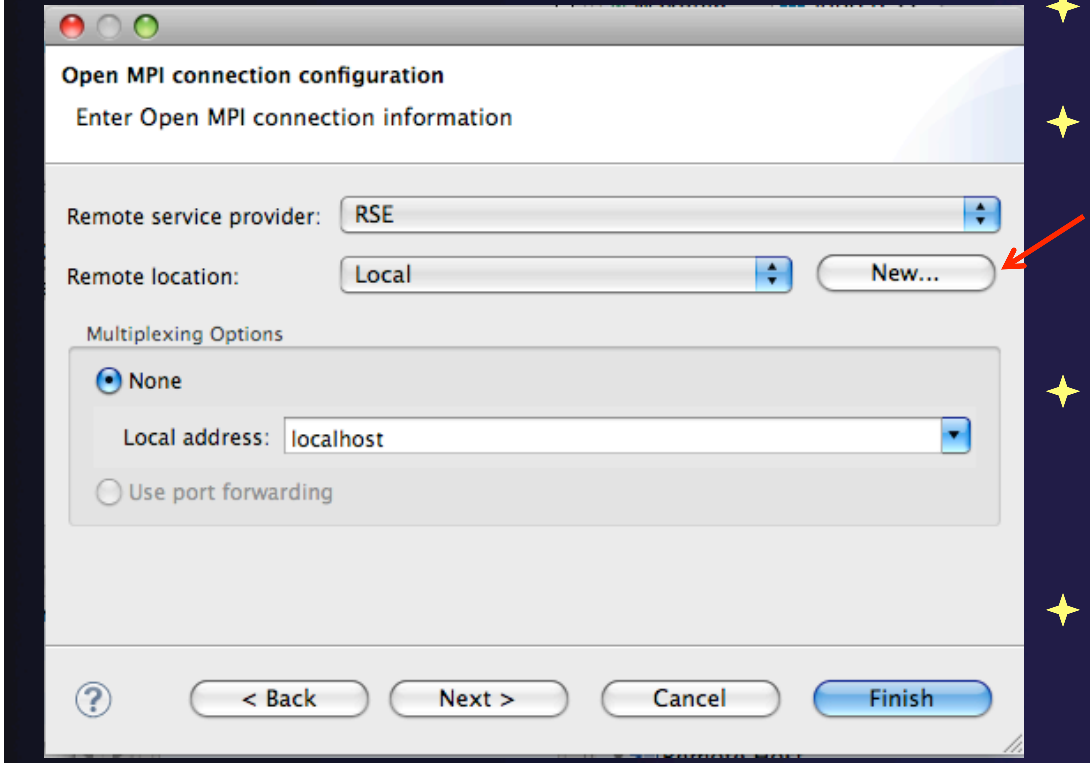
Choose Resource Manager Type

Select the type of resource manager to use

Resource Manager Types:

- IBM LoadLeveler
- IBM Parallel Environment
- MPICH2
- Open MPI

Configure the Remote Location



Open MPI connection configuration

Enter Open MPI connection information

Remote service provider: RSE

Remote location: Local

Multiplexing Options

● None

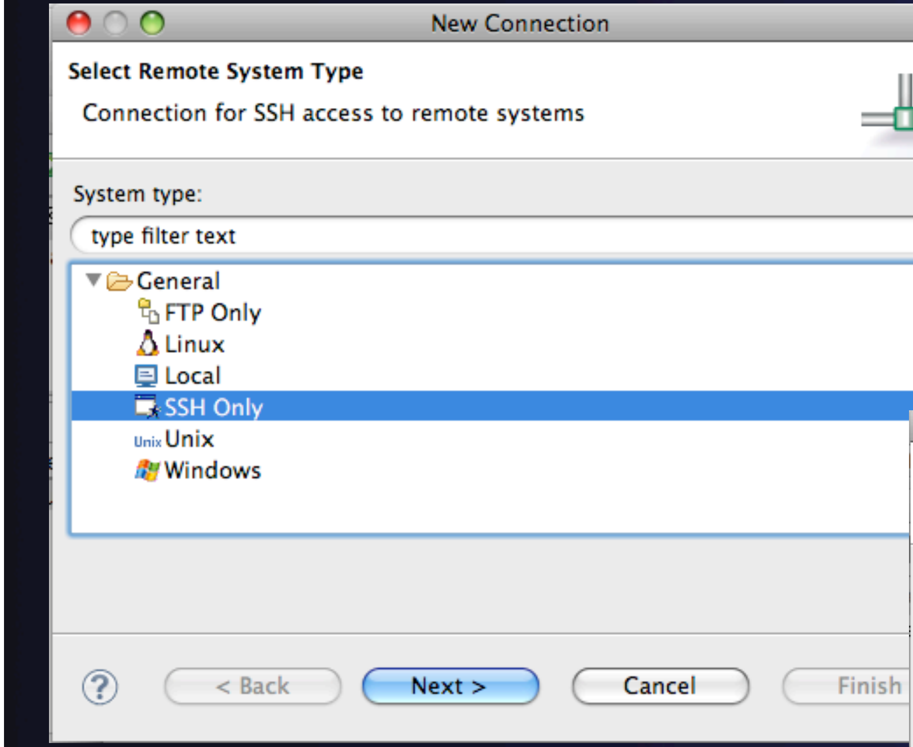
Local address: localhost

○ Use port forwarding

- Choose **RSE** for **Remote service provider**
- Choose **Remote location** or click **New...** to create a new location
  - **Local** can be used to run applications locally
- Some resource managers support tunneling over ssh connections (e.g. Remote Tools)
- The port forwarding option would be enabled this if it was available

Create a New Location (RSE)

- Choose **SSH Only** for this connection
- Click **Next>**
- Enter **Host name** of remote system
- Click **Finish**



New Connection

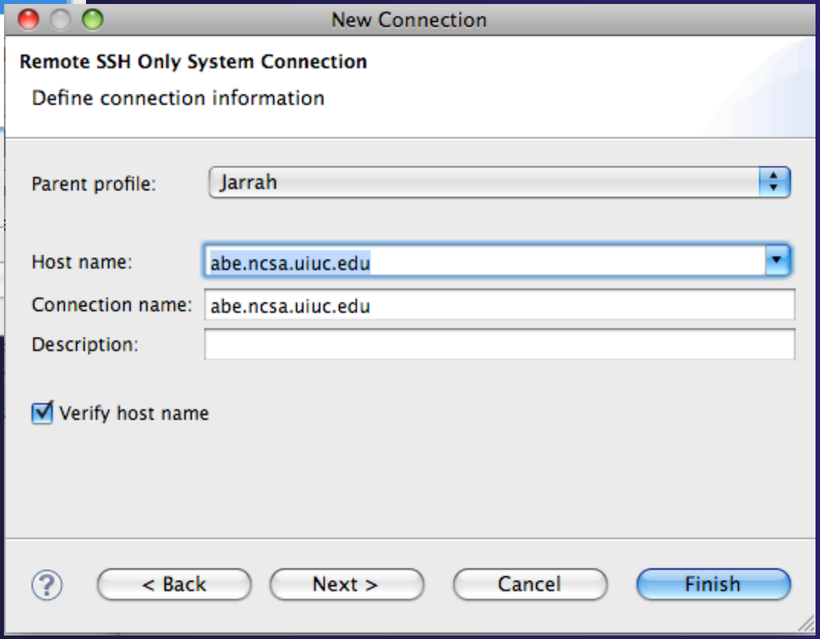
Select Remote System Type

Connection for SSH access to remote systems

System type:

type filter text

- General
  - FTP Only
  - Linux
  - Local
  - SSH Only
  - Unix
  - Windows



New Connection

Remote SSH Only System Connection

Define connection information

Parent profile: Jarrah

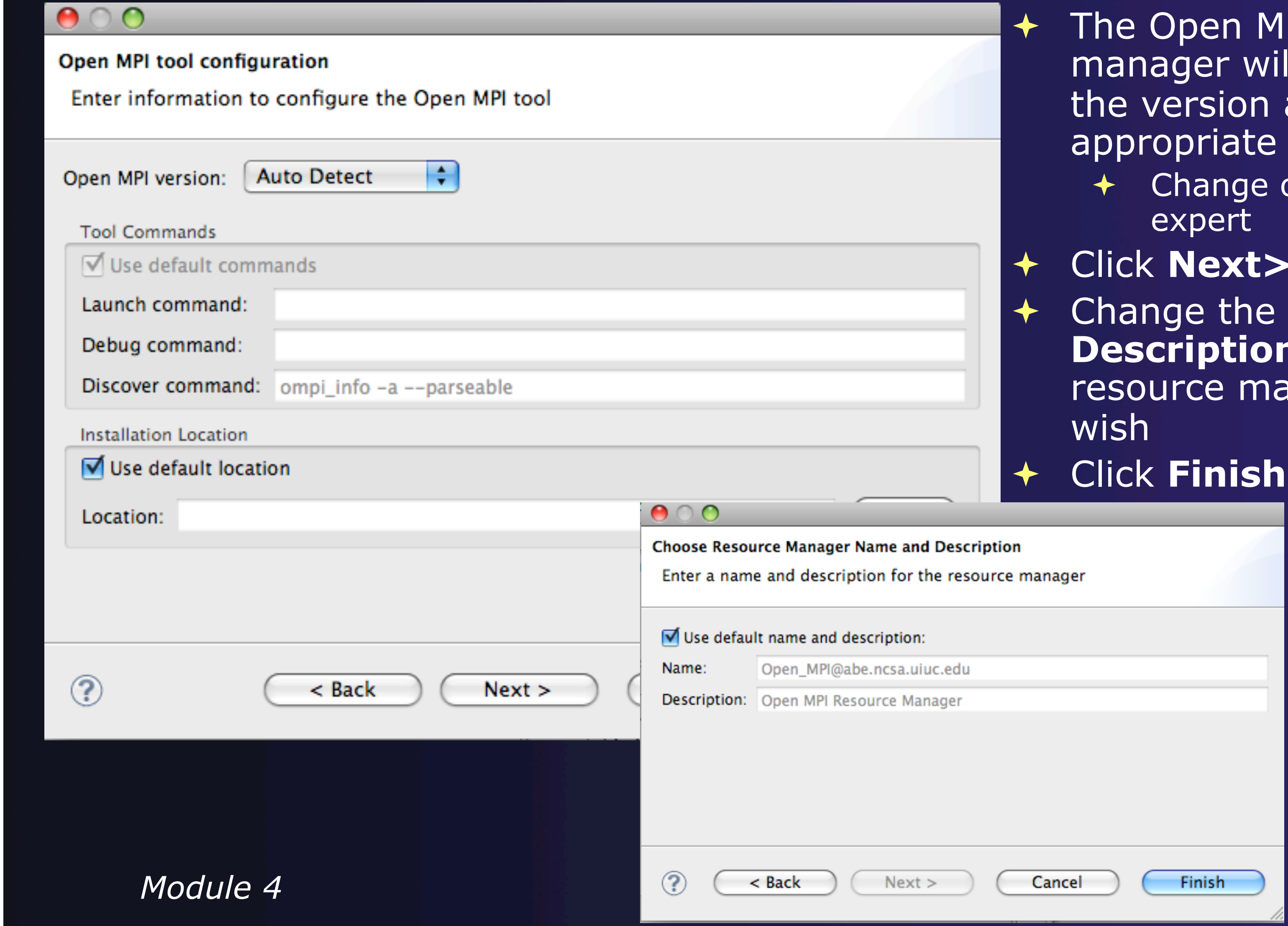
Host name: abe.ncsa.uiuc.edu

Connection name: abe.ncsa.uiuc.edu

Description:

☑ Verify host name

Configure the Resource Manager



Open MPI tool configuration

Enter information to configure the Open MPI tool

Open MPI version: Auto Detect

Tool Commands

☑ Use default commands

Launch command:

Debug command:

Discover command: ompi_info -a --parseable

Installation Location

☑ Use default location

Location:

Choose Resource Manager Name and Description

Enter a name and description for the resource manager

☑ Use default name and description:

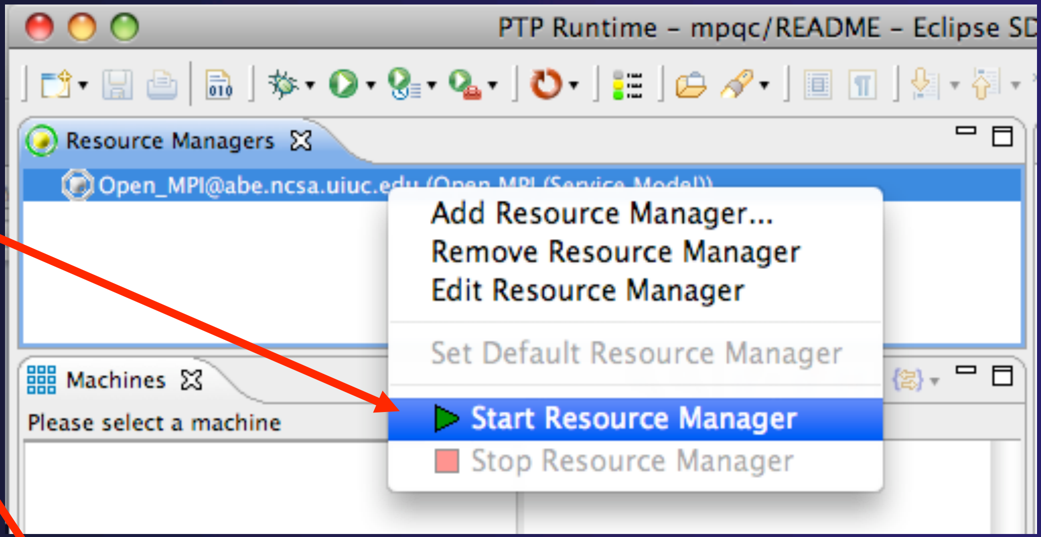
Name: Open_MPI@abe.ncsa.uiuc.edu

Description: Open MPI Resource Manager

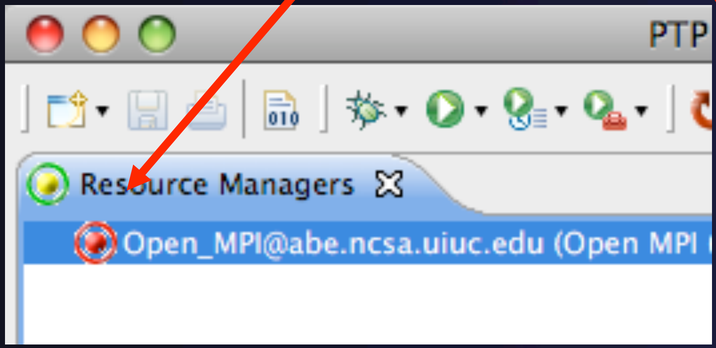
- The Open MPI resource manager will auto detect the version and use the appropriate commands
  - Change only if you’re an expert
- Click **Next>**
- Change the **Name** or **Description** of the resource manager if you wish
- Click **Finish**

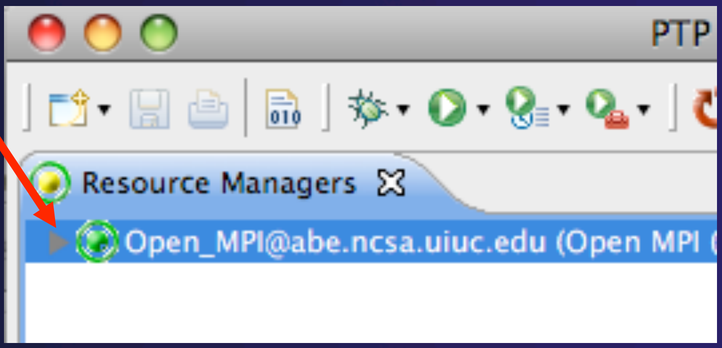
Starting the Resource Manager

- 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



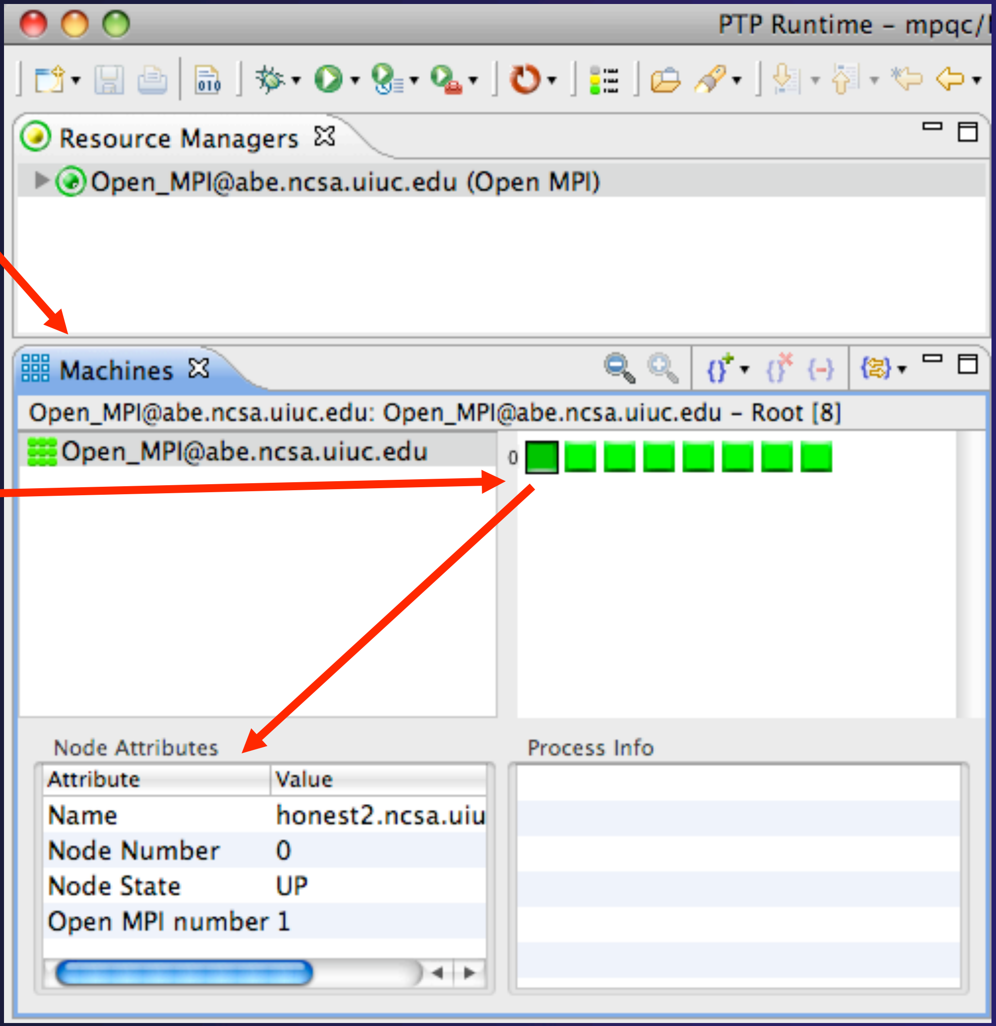
Please select a machine





System Monitoring

- Machine status shown in **Machines** view
- Node status also shown **Machines** view
- Hover over node to see node name
- Double-click on node to show attributes



PTP Runtime – mpqc/

Resource Managers

Open_MPI@abe.ncsa.uiuc.edu (Open MPI)

Machines

Open_MPI@abe.ncsa.uiuc.edu: Open_MPI@abe.ncsa.uiuc.edu – Root [8]

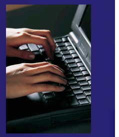
Open_MPI@abe.ncsa.uiuc.edu

0 🟩🟩🟩🟩🟩🟩🟩🟩

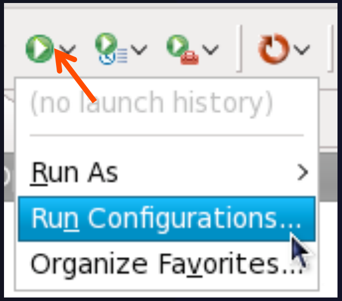
Node Attributes

Attribute	Value
Name	honest2.ncsa.uiu
Node Number	0
Node State	UP
Open MPI number	1

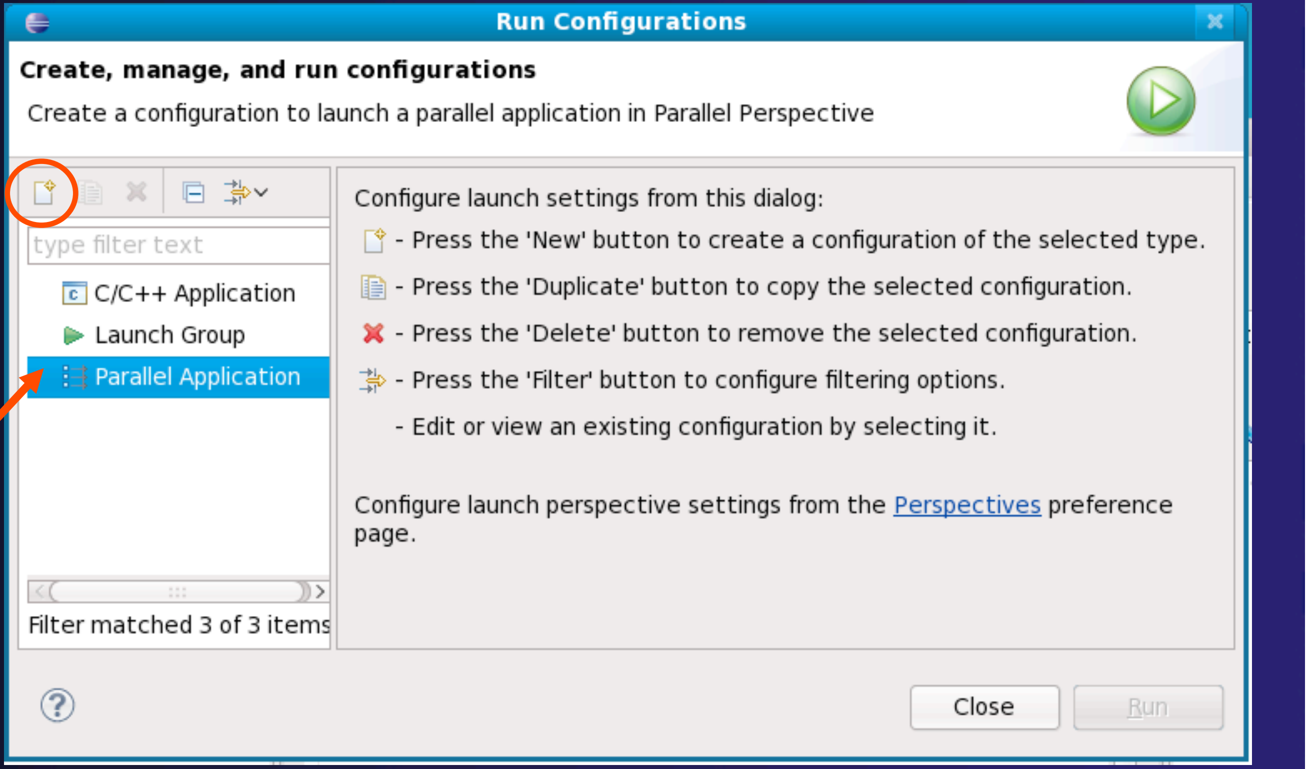
Process Info



Create a Launch Configuration



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



Create, manage, and run configurations

Create a configuration to launch a parallel application in Parallel Perspective

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.

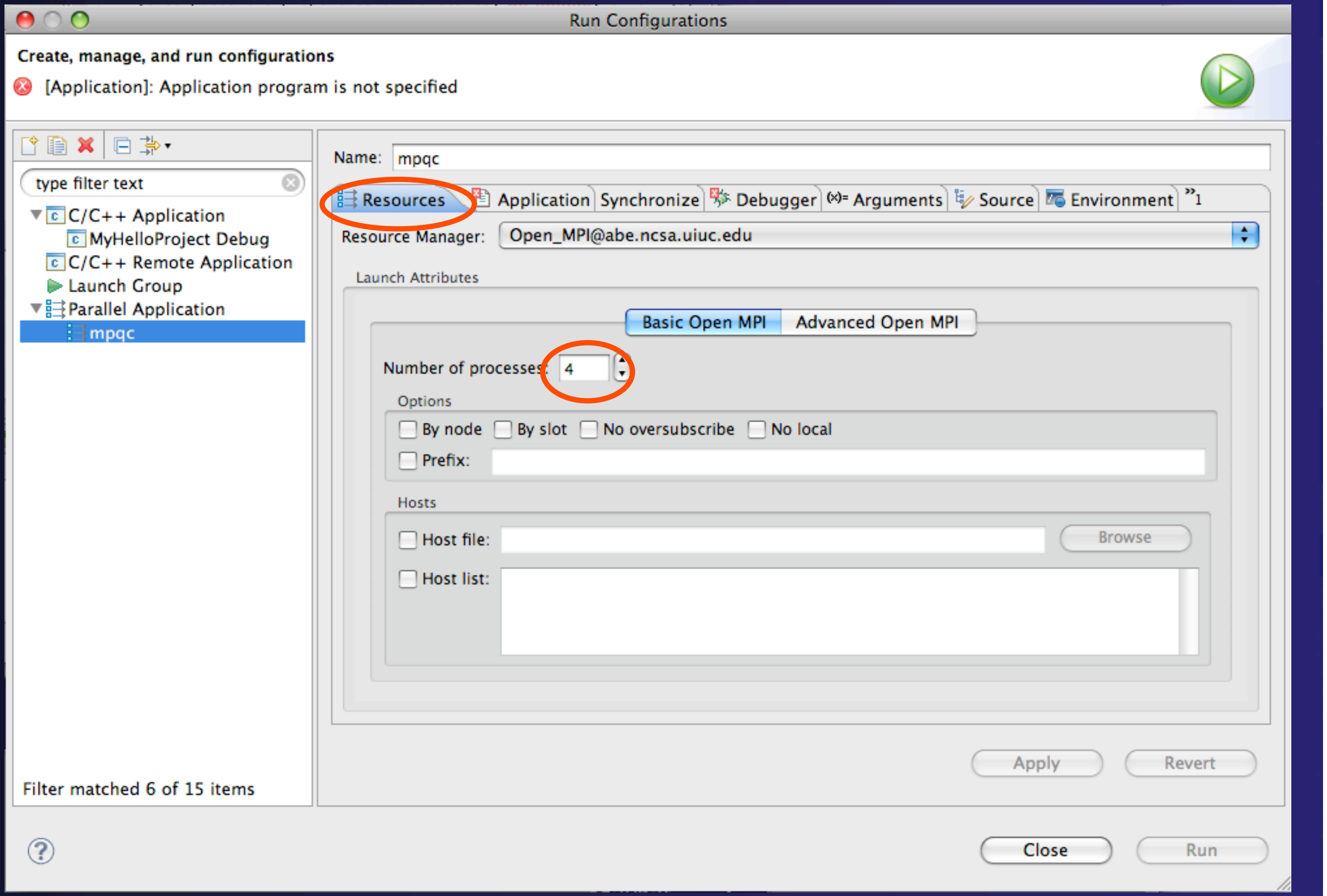
Configure launch perspective settings from the Perspectives preference page.

Filter matched 3 of 3 items

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

Complete the Resources Tab

- In **Resources** tab, select the resource manager you want to use to launch this job
- Enter a value in the **Number of processes** field
- Other fields can be used to specify resource manager-specific information



Run Configurations

Create, manage, and run configurations

[Application]: Application program is not specified

Name: mpqc

Resource Manager: Open_MPI@abe.ncsa.uiuc.edu

Number of processes: 4

□ By node □ By slot □ No oversubscribe □ No local

□ Prefix:

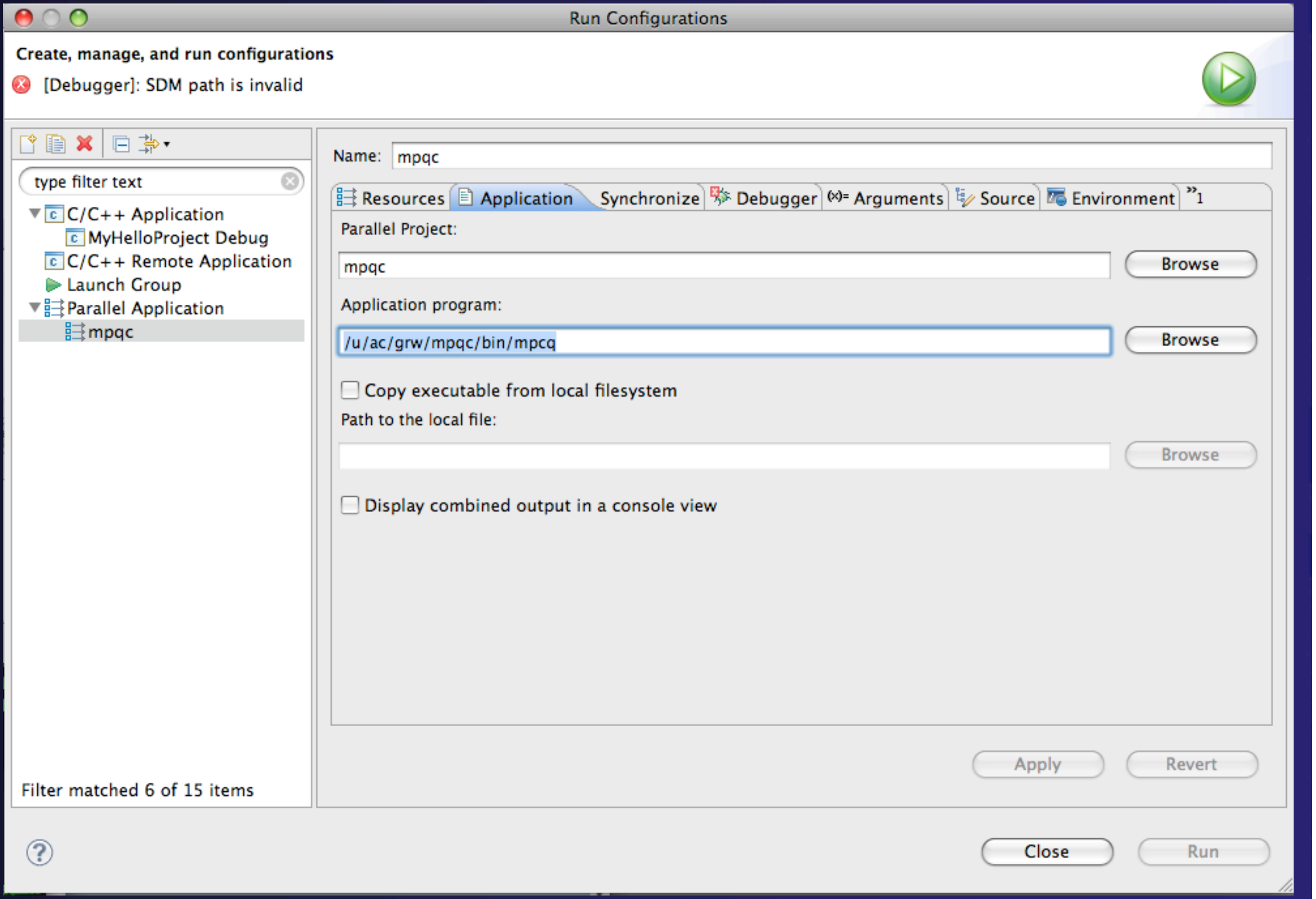
□ Host file:

□ Host list:

Filter matched 6 of 15 items

Complete the Application Tab

- Select the **Application** tab
- Choose the **Application program** (executable) by clicking the **Browse** button
  - Local program: executable is under Debug folder in the project
  - Remote program: must copy to remote machine; navigate to its location on the remote machine here
- Select **Display combined output in a console view** if desired



Run Configurations

Create, manage, and run configurations

[Debugger]: SDM path is invalid

Name: mpqc

Parallel Project: mpqc

Application program: /u/ac/grw/mpqc/bin/mpcq

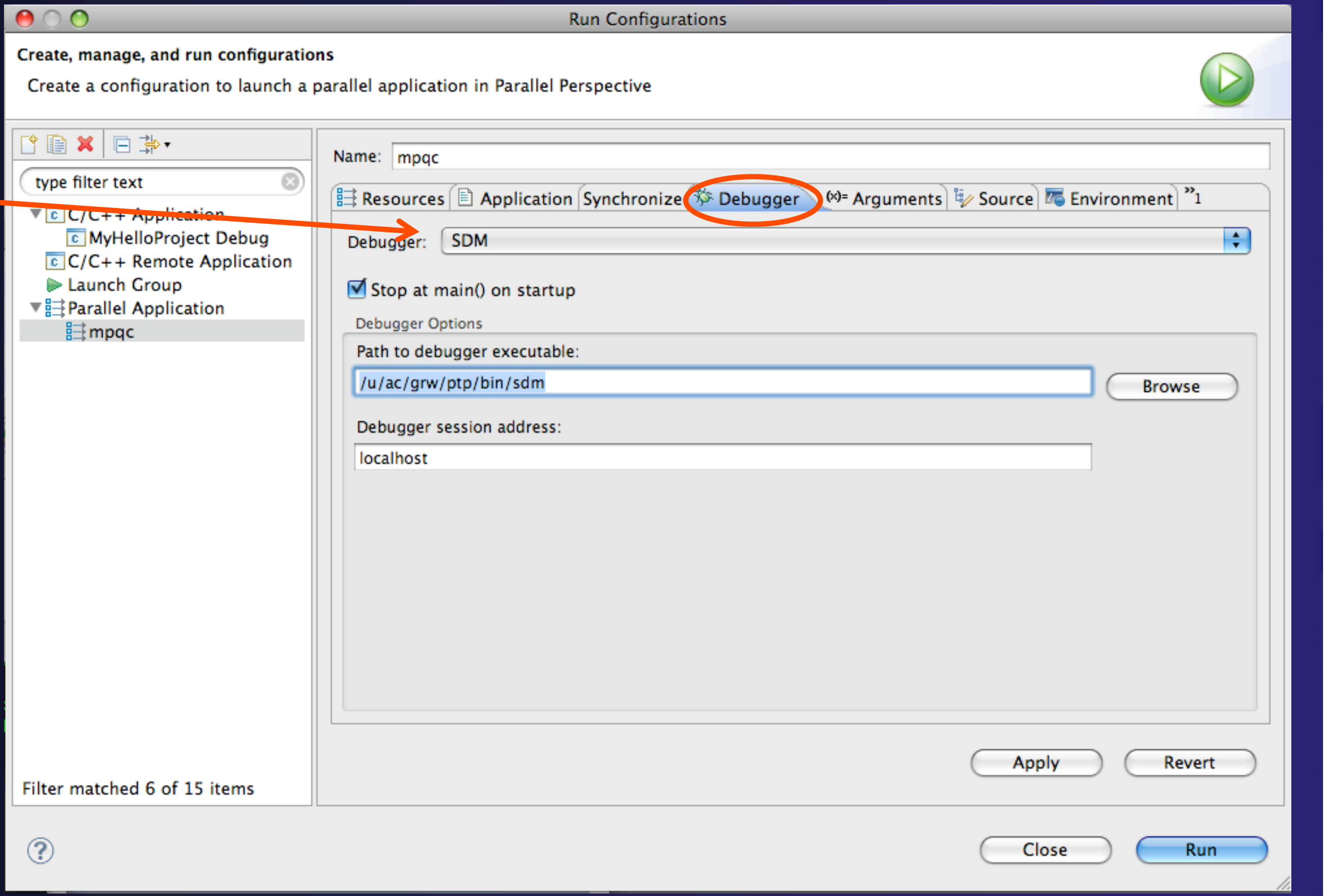
☐ Copy executable from local filesystem

Path to the local file:

☐ Display combined output in a console view

Complete the Debugger Tab

- Select **Debugger** tab
- Choose **SDM** from the **Debugger** dropdown
- Use the **Browse** button to select the debugger executable
  - If launching remotely, the debugger executable must also be located remotely
- Set debugger session address (covered later)
- Click on **Run** to launch the program


Run Configurations

Create, manage, and run configurations

Create a configuration to launch a parallel application in Parallel Perspective

Name: mpqc

Debugger: SDM

☑ Stop at main() on startup

Debugger Options

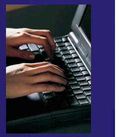
Path to debugger executable: /u/ac/grw/ptp/bin/sdm

Debugger session address: localhost

Filter matched 6 of 15 items

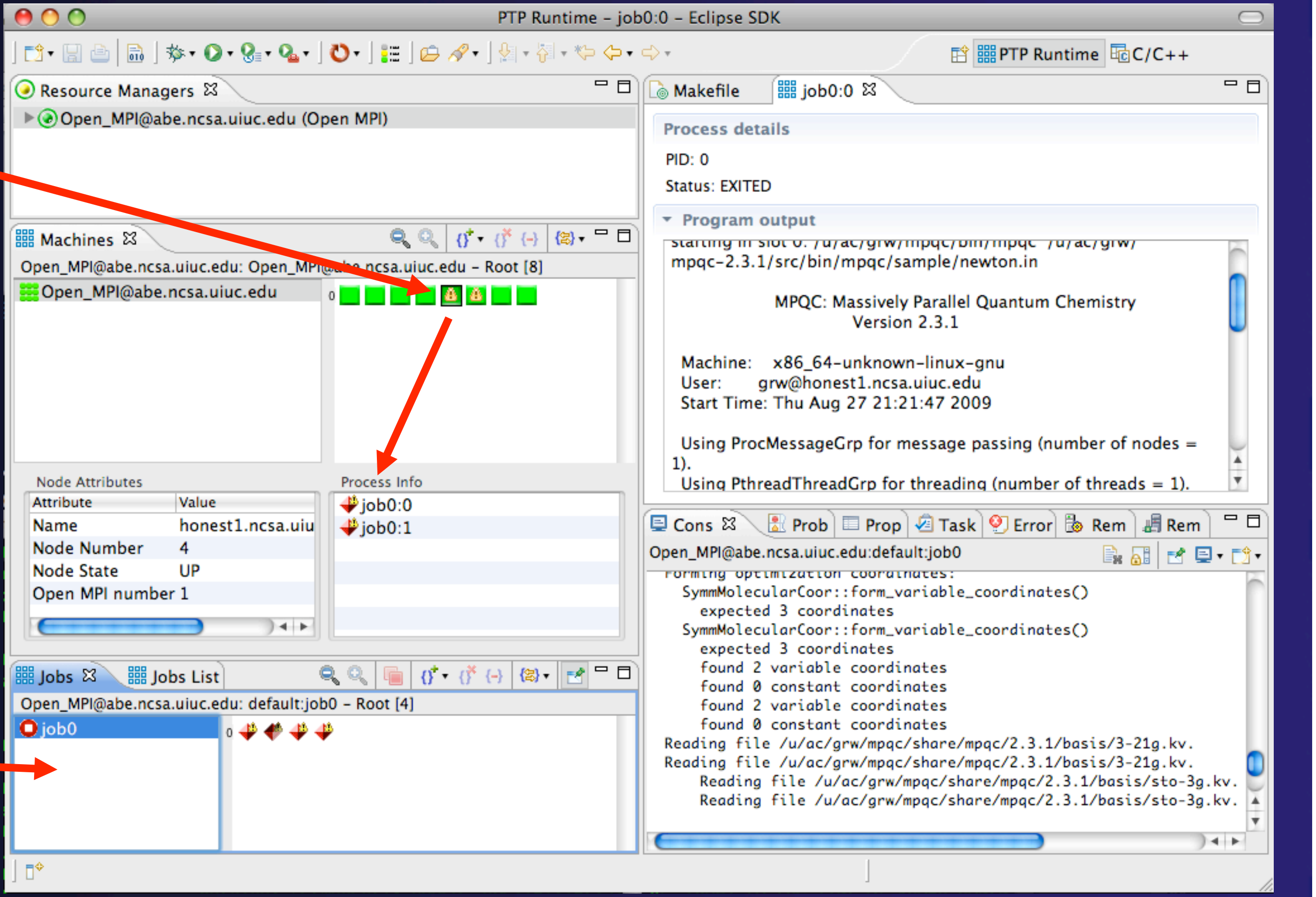
The debugger settings will not be used until the application is launched under the debugger

This will be covered in more detail in Module 6



Viewing The Run

- Double-click a node in machines view to see which processes ran on the node
- Hover over a process for tooltip popup
- Job and processes shown in jobs view



PTP Runtime – job0:0 – Eclipse SDK
PTP Runtime
C/C++
Resource Managers
Open_MPI@abe.ncsa.uiuc.edu (Open MPI)
Machines
Open_MPI@abe.ncsa.uiuc.edu: Open_MPI@abe.ncsa.uiuc.edu – Root [8]
Open_MPI@abe.ncsa.uiuc.edu
0
Node Attributes
Process Info
job0:0
job0:1
Jobs
Jobs List
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]
job0
0
Makefile
job0:0
Process details
Program output
Cons
Prob
Prop
Task
Error
Rem
Rem
Open_MPI@abe.ncsa.uiuc.edu:default:job0

Attribute	Value
Name	honest1.ncsa.uiu
Node Number	4
Node State	UP
Open MPI number	1

PID: 0

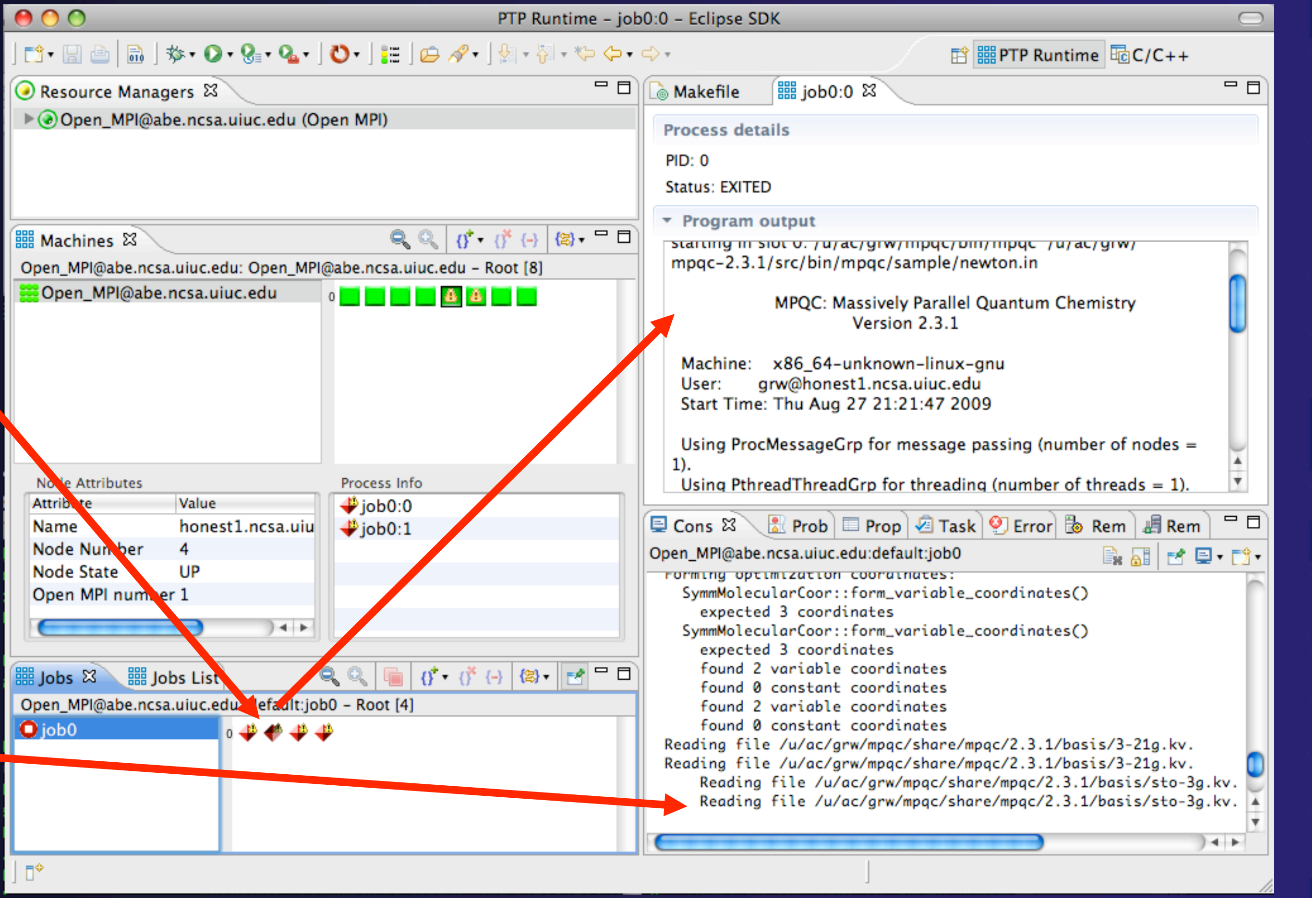
Status: EXITED

```
starting in slot 0: /u/ac/grw/mpqc/bin/mpqc /u/ac/grw/
mpqc-2.3.1/src/bin/mpqc/sample/newton.in

               MPQC: Massively Parallel Quantum Chemistry
                           Version 2.3.1

  Machine:    x86_64-unknown-linux-gnu
  User:       grw@honest1.ncsa.uiuc.edu
  Start Time: Thu Aug 27 21:21:47 2009

  Using ProcMessageGrp for message passing (number of nodes =
1).
  Using PthreadThreadGrp for threading (number of threads = 1).
```

```
  Forming optimization coordinates:
    SymmMolecularCoor::form_variable_coordinates()
      expected 3 coordinates
    SymmMolecularCoor::form_variable_coordinates()
      expected 3 coordinates
      found 2 variable coordinates
      found 0 constant coordinates
      found 2 variable coordinates
      found 0 constant coordinates
  Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/3-21g.kv.
  Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/3-21g.kv.
      Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/sto-3g.kv.
      Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/sto-3g.kv.
```

Viewing Program Output

- Double-click a process to see process detail and standard output from the process
- Console displays combined output from all processes


PTP Runtime – job0:0 – Eclipse SDK
PTP Runtime
C/C++
Resource Managers
Open_MPI@abe.ncsa.uiuc.edu (Open MPI)
Machines
Open_MPI@abe.ncsa.uiuc.edu: Open_MPI@abe.ncsa.uiuc.edu – Root [8]
Open_MPI@abe.ncsa.uiuc.edu
0
Node Attributes
Process Info
job0:0
job0:1
Jobs
Jobs List
Open_MPI@abe.ncsa.uiuc.edu:default:job0 – Root [4]
job0
0
Makefile
job0:0
Process details
Program output
Cons
Prob
Prop
Task
Error
Rem
Rem
Open_MPI@abe.ncsa.uiuc.edu:default:job0

Attribute	Value
Name	honest1.ncsa.uiu
Node Number	4
Node State	UP
Open MPI number	1

PID: 0

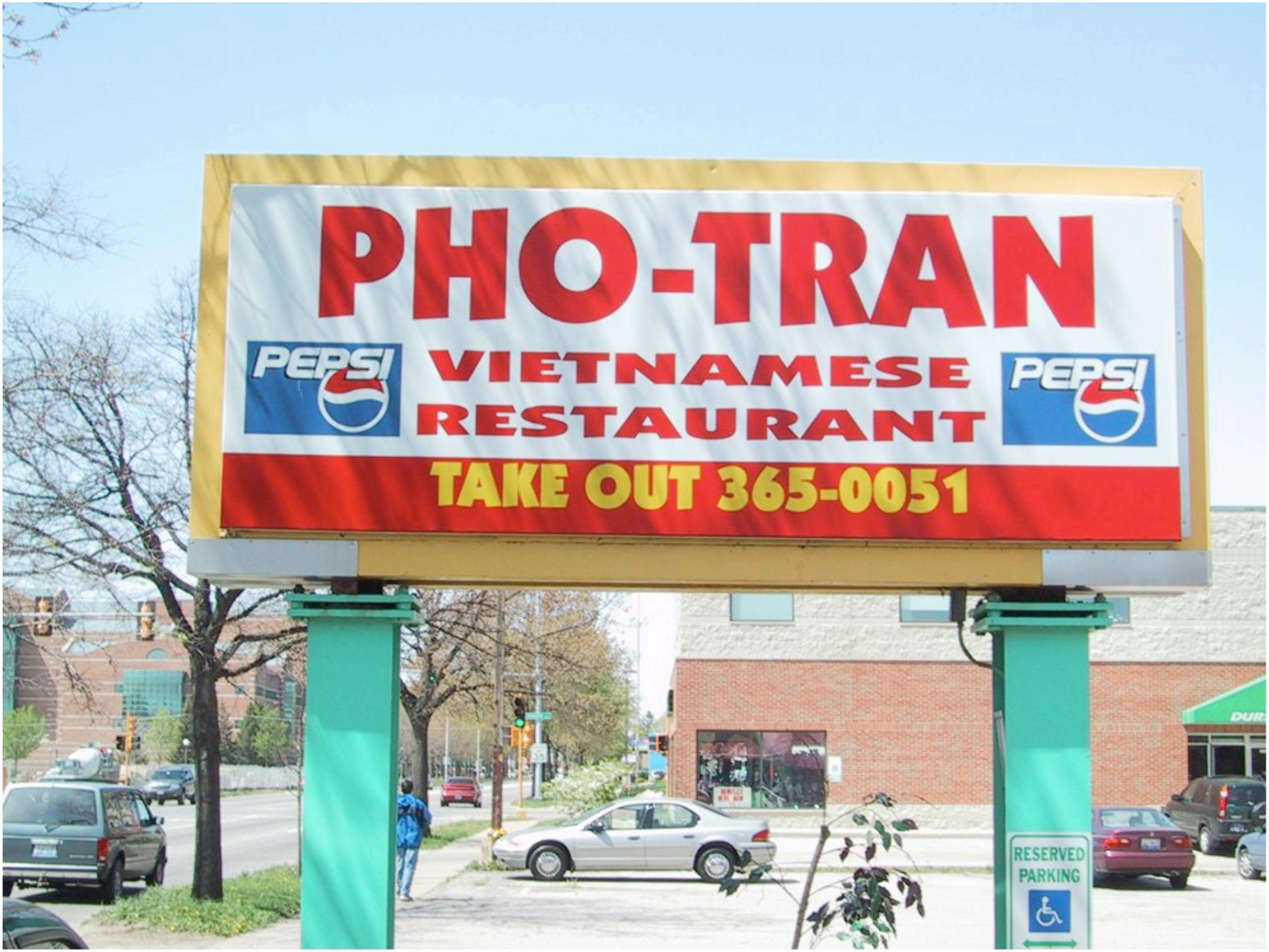
Status: EXITED

```
starting in slot 0: /u/ac/grw/mpqc/bin/mpqc /u/ac/grw/
mpqc-2.3.1/src/bin/mpqc/sample/newton.in

                MPQC: Massively Parallel Quantum Chemistry
                              Version 2.3.1

  Machine:    x86_64-unknown-linux-gnu
  User:       grw@honest1.ncsa.uiuc.edu
  Start Time: Thu Aug 27 21:21:47 2009

  Using ProcMessageGrp for message passing (number of nodes =
1).
  Using PthreadThreadGrp for threading (number of threads = 1).
```

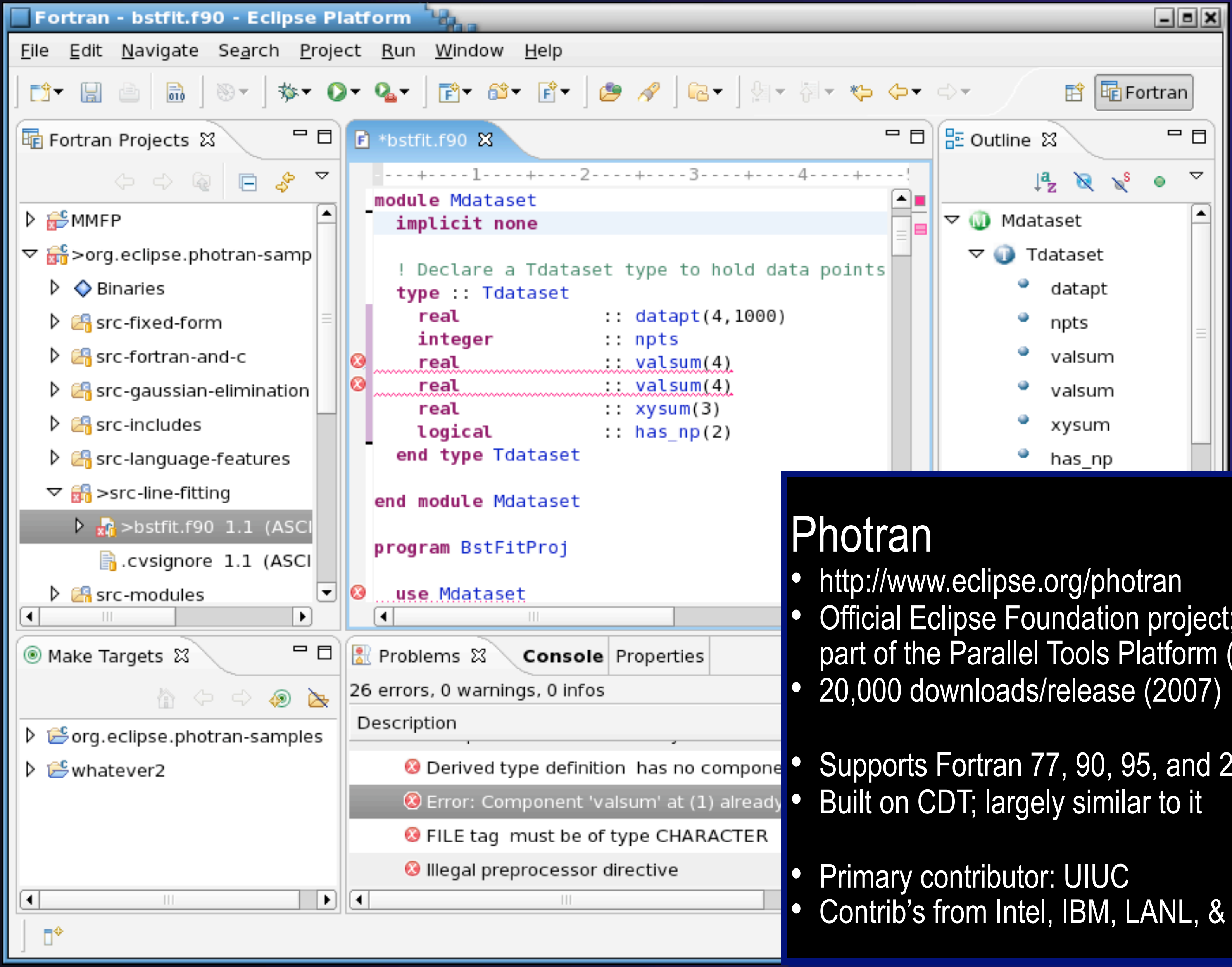
```
Forming optimization coordinates:
  SymmMolecularCoor::form_variable_coordinates()
    expected 3 coordinates
  SymmMolecularCoor::form_variable_coordinates()
    expected 3 coordinates
    found 2 variable coordinates
    found 0 constant coordinates
    found 2 variable coordinates
    found 0 constant coordinates
Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/3-21g.kv.
Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/3-21g.kv.
    Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/sto-3g.kv.
    Reading file /u/ac/grw/mpqc/share/mpqc/2.3.1/basis/sto-3g.kv.
```

Module 5: Fortran

- Objective
  - Learn what Photran is and how it compares to CDT
  - Learn how to create a Fortran MPI application
- Contents
  - Overview of Photran
  - Module 3 redux (in Fortran)
  - Differences between Photran and CDT
  - Pointers to online documentation for Photran


PHO-TRAN
PEPSI
VIETNAMESE
RESTAURANT
PEPSI
TAKE OUT 365-0051
RESERVED
PARKING

PHOTRAN
eclipse
IDE for Fortran
eclipse
TAKE OUT 365-0051
RESERVED PARKING



```
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

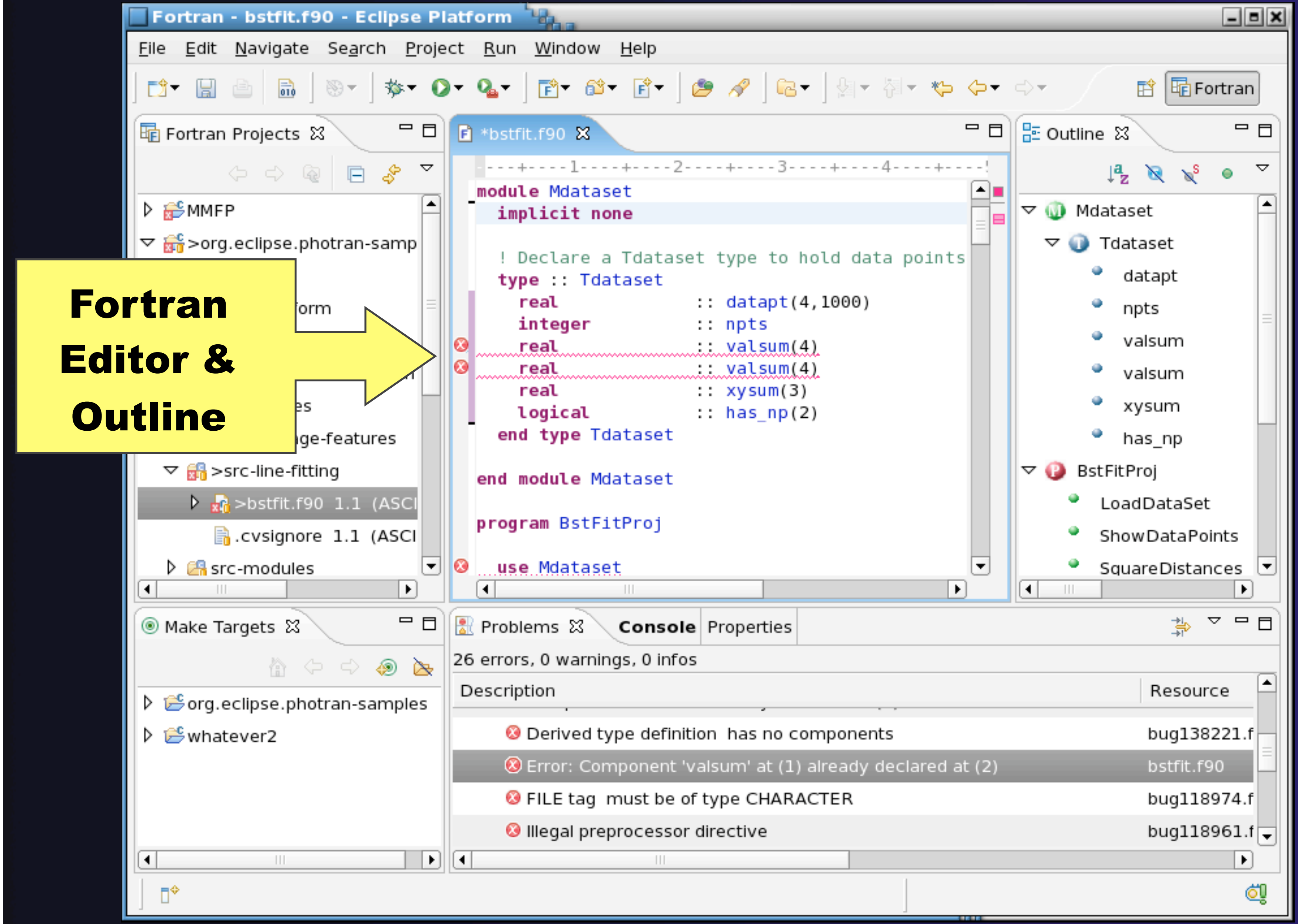
- Derived type definition has no compone
- Error: Component 'valsum' at (1) already
- FILE tag must be of type CHARACTER
- Illegal preprocessor directive

Photran

- http://www.eclipse.org/photran
- Official Eclipse Foundation project; part of the Parallel Tools Platform (PTP)
- 20,000 downloads/release (2007)

- Supports Fortran 77, 90, 95, and 2003
- Built on CDT; largely similar to it

- Primary contributor: UIUC
- Contrib’s from Intel, IBM, LANL, & others

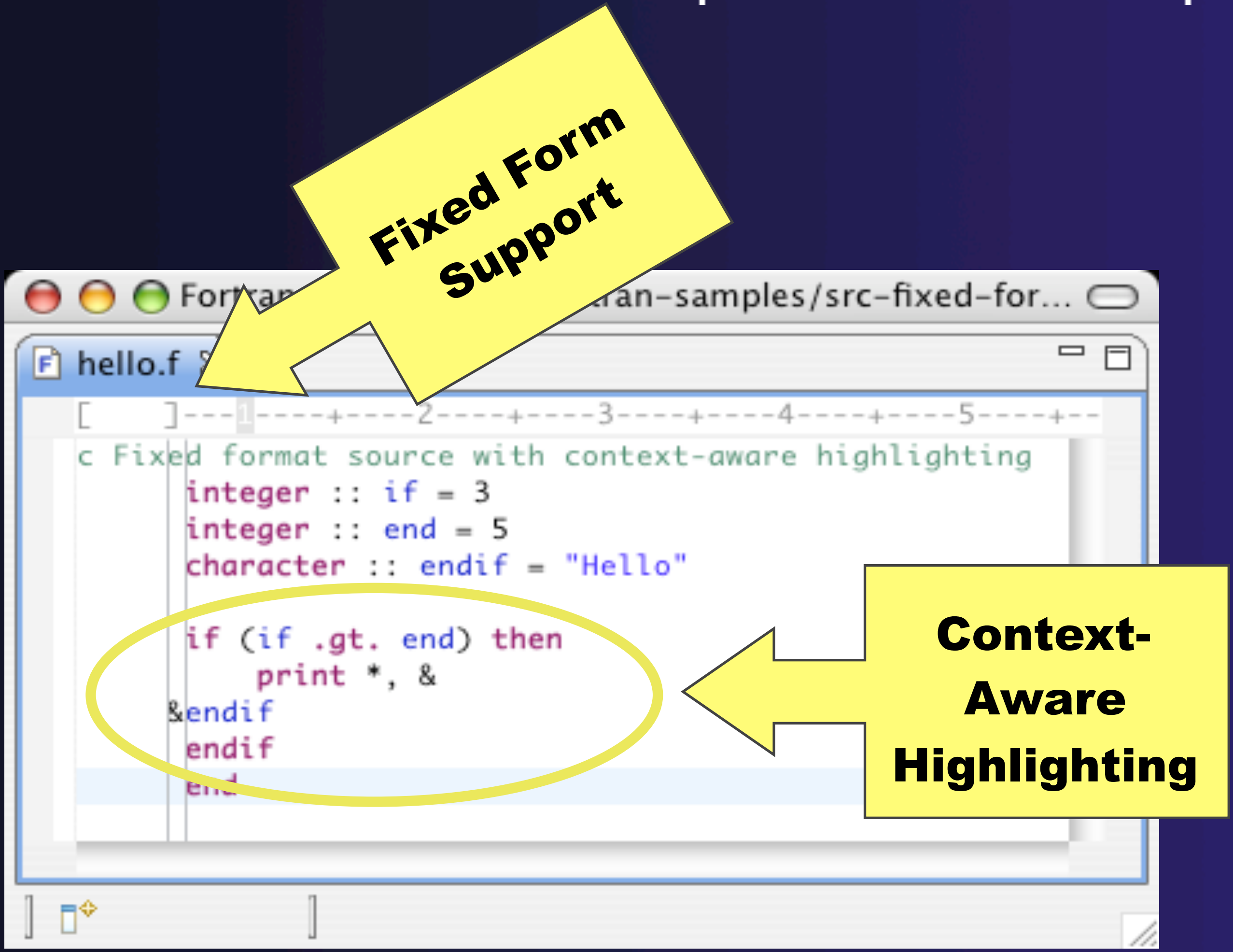


Fortran Editor & Outline

```
----+----1----+----2----+----3----+----4----+----
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

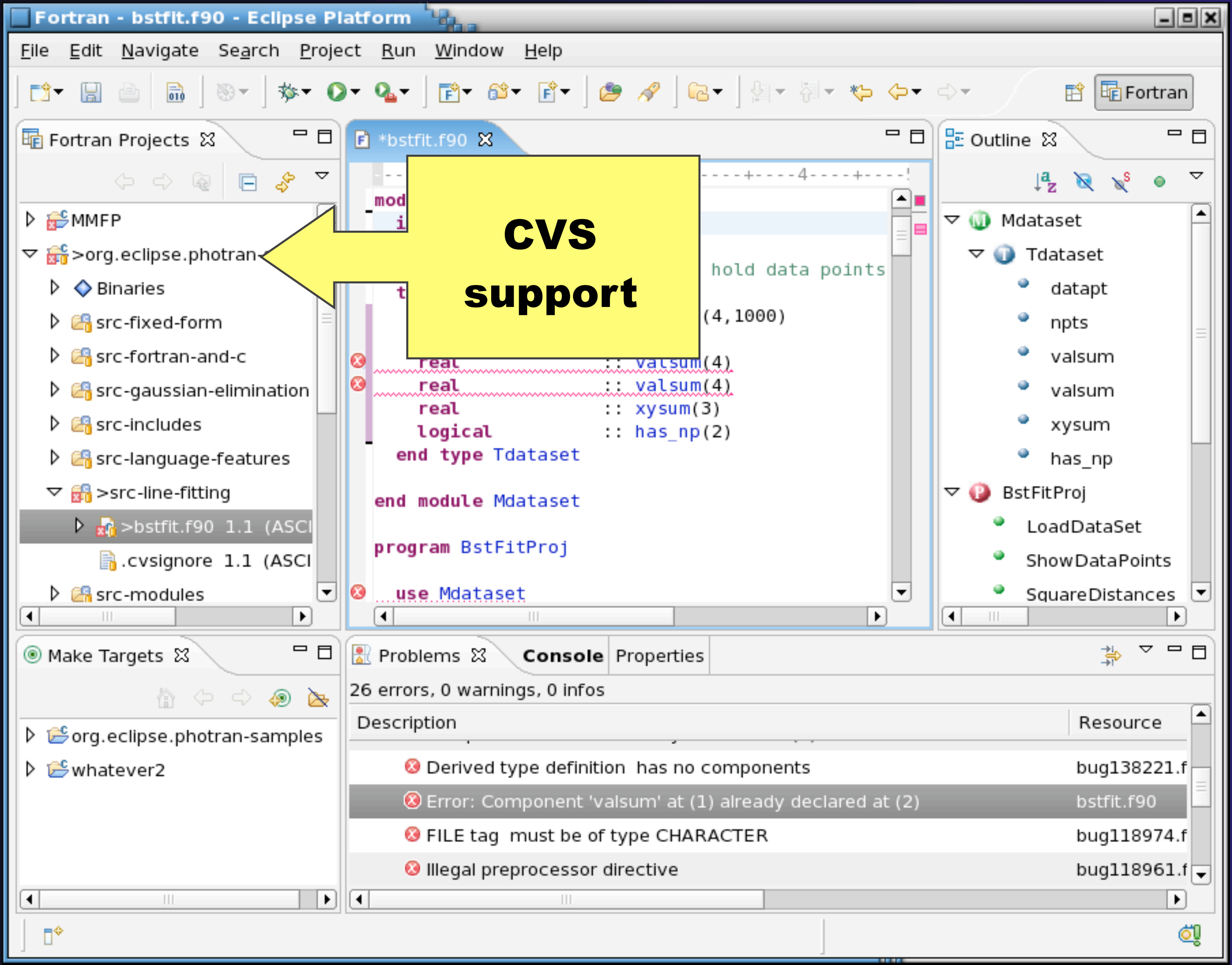
Description	Resource
Derived type definition has no components	bug138221.f
Error: Component 'valsum' at (1) already declared at (2)	bstfit.f90
FILE tag must be of type CHARACTER	bug118974.f
Illegal preprocessor directive	bug118961.f



```
c Fixed format source with context-aware highlighting
      integer :: if = 3
      integer :: end = 5
      character :: endif = "Hello"

      if (if .gt. end) then
          print *, &
     &endif
      endif
      end
```

Fixed Form Support

Context-Aware Highlighting

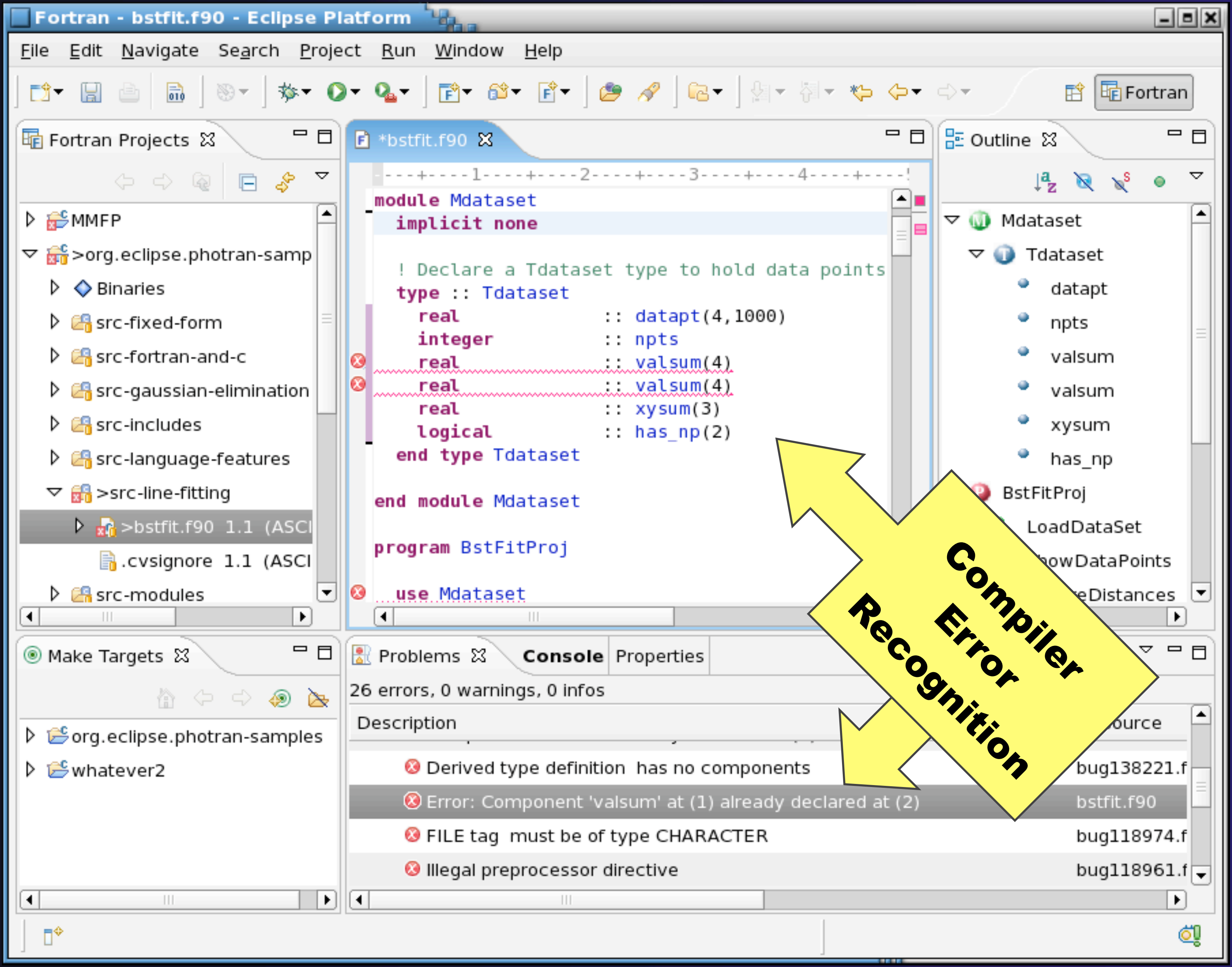


CVS support

```
mod
  i
                                    hold data points
  t                                 (4,1000)
     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

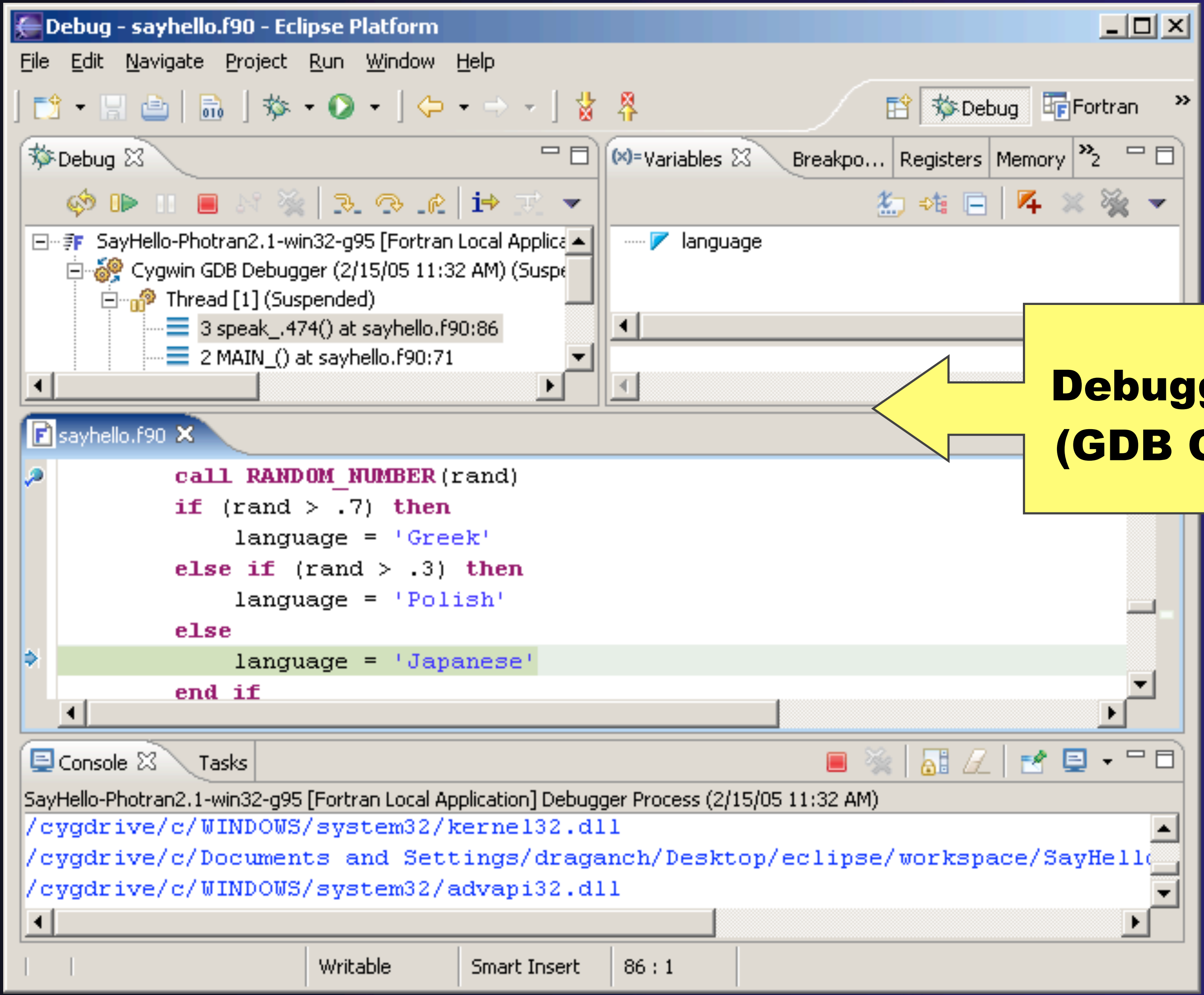


```
module Mdataset
  implicit none

  ! 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	ource
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

Compiler Error Recognition



```
SayHello-Photran2.1-win32-g95 [Fortran Local Applica
  Cygwin GDB Debugger (2/15/05 11:32 AM) (Suspe
    Thread [1] (Suspended)
      3 speak_.474() at sayhello.f90:86
      2 MAIN_() at sayhello.f90:71
```

```
language
```

```fortran
        call RANDOM_NUMBER(rand)
        if (rand > .7) then
            language = 'Greek'
        else if (rand > .3) then
            language = 'Polish'
        else
            language = 'Japanese'
        end if
```

```
SayHello-Photran2.1-win32-g95 [Fortran Local Application] Debugger Process (2/15/05 11:32 AM)
/cygdrive/c/WINDOWS/system32/kernel32.dll
/cygdrive/c/Documents and Settings/draganch/Desktop/eclipse/workspace/SayHello
/cygdrive/c/WINDOWS/system32/advapi32.dll
```

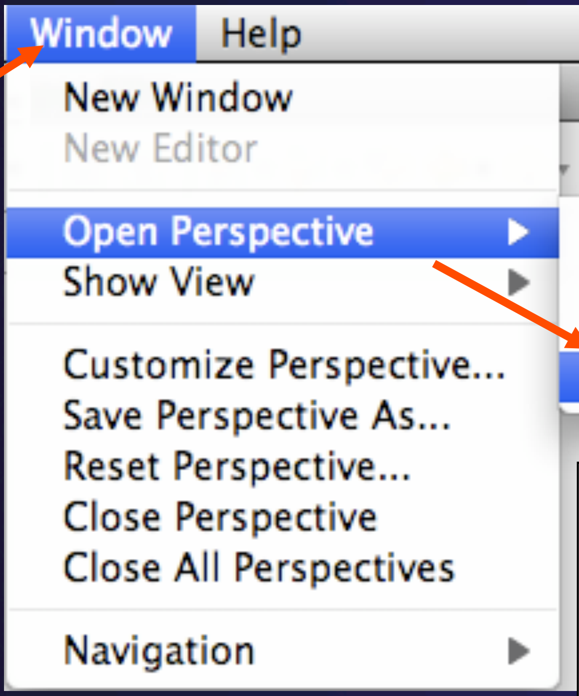
Debugging (GDB GUI)

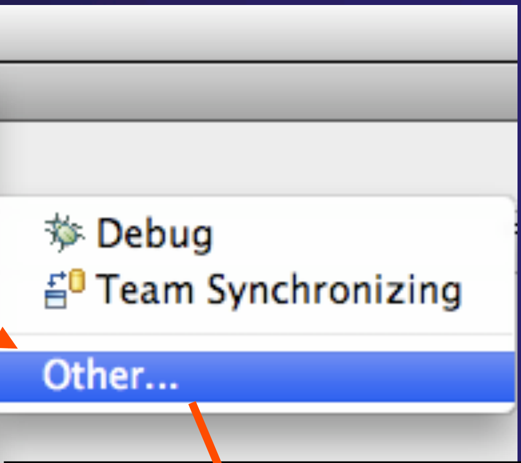
Using Photran

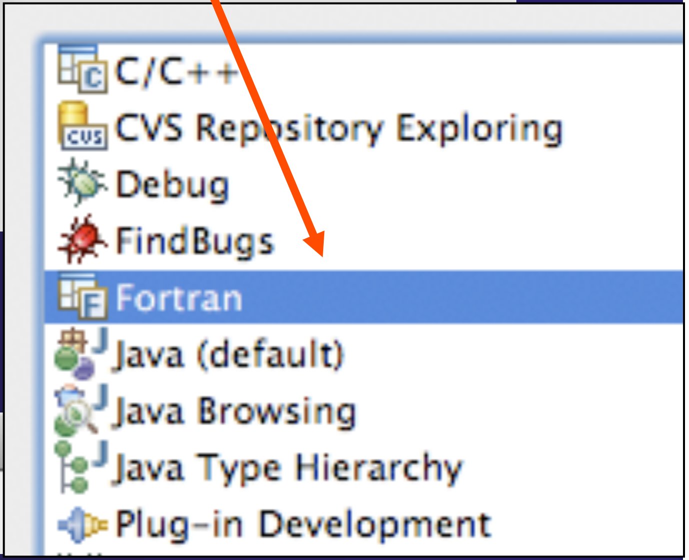
- It’s just like using CDT…
  - Similar New Project wizards
  - Similar build procedure
  - Similar launch/debug procedure
- …but not exactly
  - Configuring fixed vs. free form file extensions
  - Different editor features
  - Different advanced features (Module 7)

Switch to ~~C/C++~~ Fortran Perspective

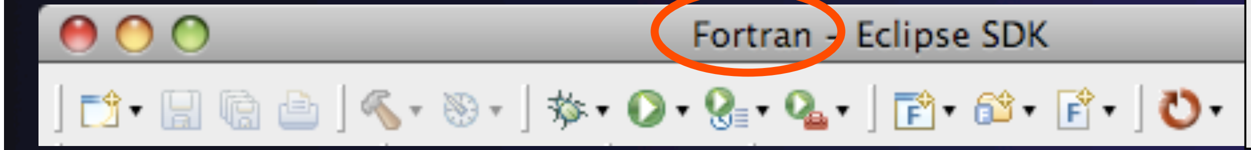
- Only needed if you're not already in the perspective







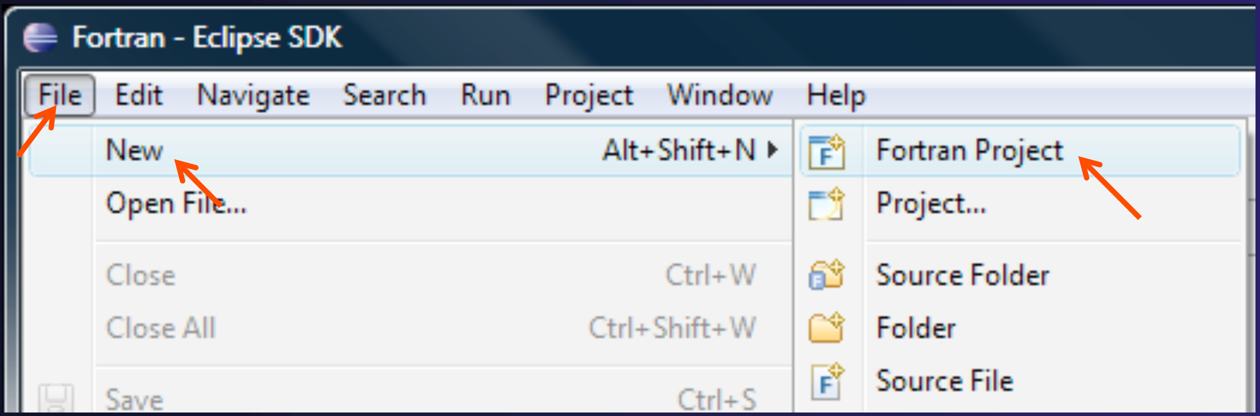
- What Perspective am in in?
  See Title Bar



Creating a ~~C/C++~~ Fortran Application

Steps:

- Create a new ~~C~~ Fortran project
- Edit source code
- Save and build

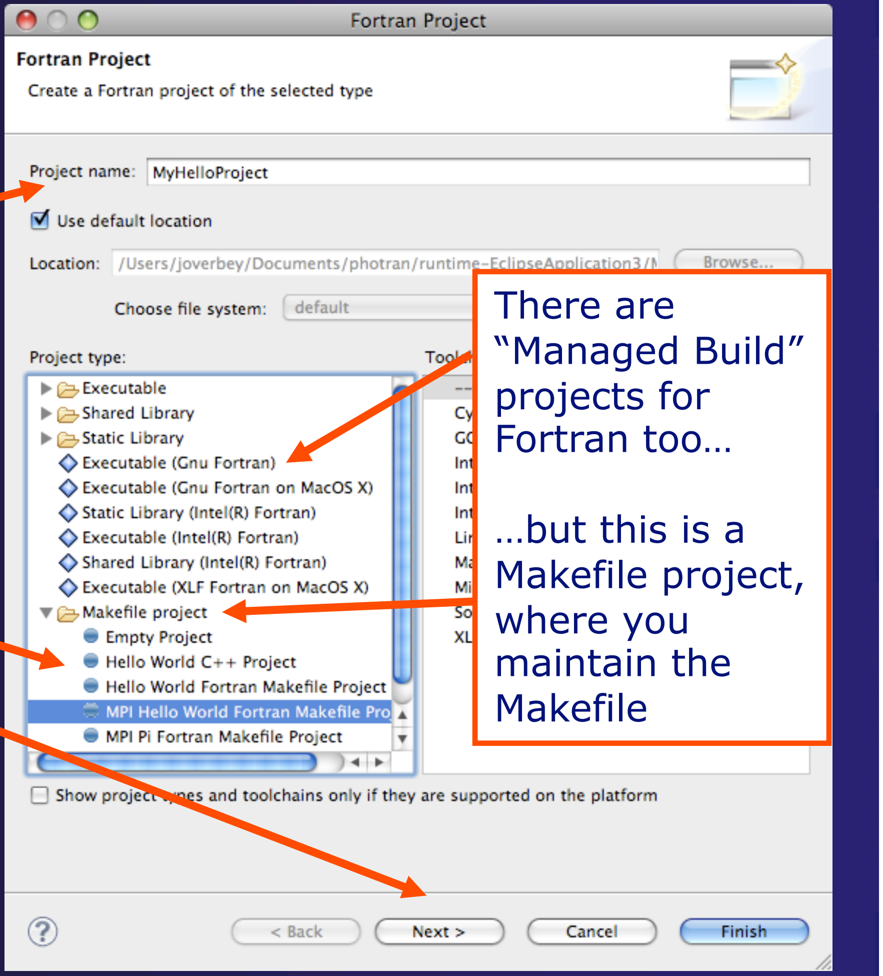


Fortran

New ~~C~~ Project Wizard

Create a new MPI project

- File ▶ New ▶ ~~C Project~~ **Fortran Project** (see prev. slide)
- Name the project 'MyHelloProject'
- Under Project types, under ~~Executable, select **MPI Hello World Project**~~ Makefile Project, select **MPI Hello World Fortran Project** and hit **Next**
- On **Basic Settings** page, fill in information for your new project (**Author name** etc.) and hit ~~Next~~ **Finish**



Fortran Project

Fortran Project

Create a Fortran project of the selected type

Project name: MyHelloProject

☑ Use default location

Location: /Users/joverbey/Documents/photran/runtime-EclipseApplication3/N

Choose file system: default

Project type:

- ▶ Executable
- ▶ Shared Library
- ▶ Static Library
- Executable (Gnu Fortran)
- Executable (Gnu Fortran on MacOS X)
- Static Library (Intel(R) Fortran)
- Executable (Intel(R) Fortran)
- Shared Library (Intel(R) Fortran)
- Executable (XLF Fortran on MacOS X)
- ▼ Makefile project
  - Empty Project
  - Hello World C++ Project
  - Hello World Fortran Makefile Project
  - MPI Hello World Fortran Makefile Pro
  - MPI Pi Fortran Makefile Project

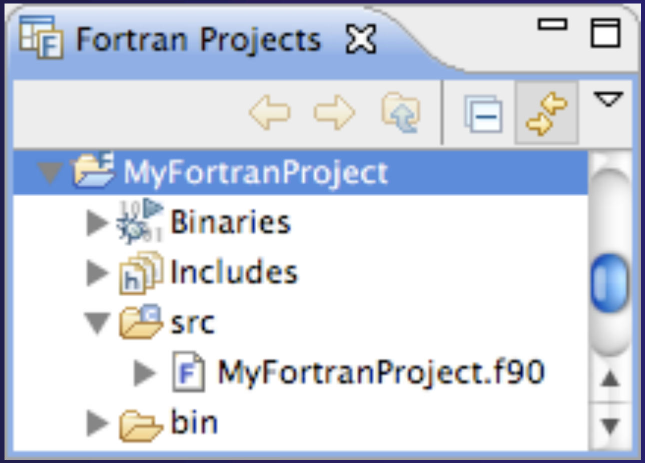
☐ Show project types and toolchains only if they are supported on the platform

There are "Managed Build" projects for Fortran too…

…but this is a Makefile project, where you maintain the Makefile

Fortran Projects
~~Project Explorer~~ View

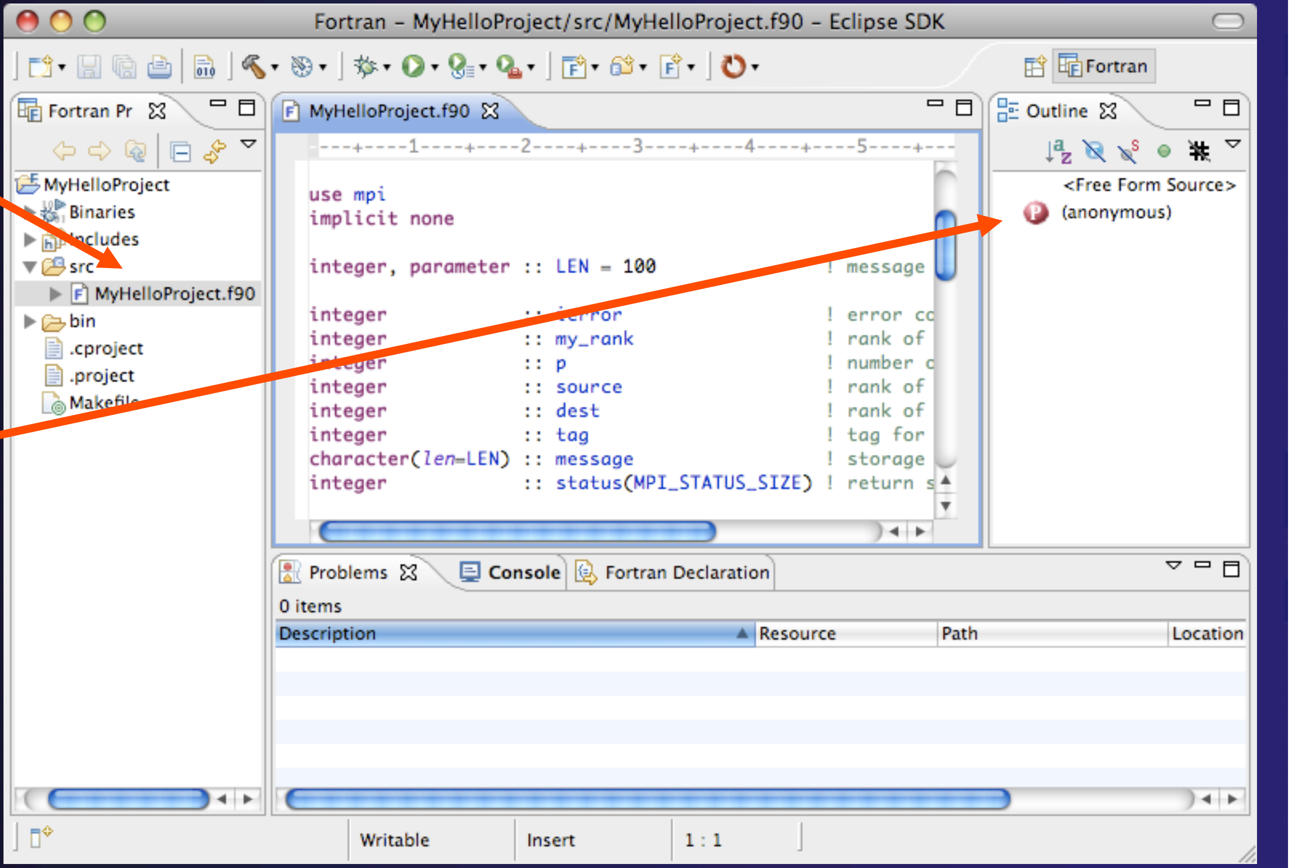
- Represents user's data
- It is a set of user defined resources
  - Files
  - Folders
  - Projects
    - Collections of files and folders
    - Plus meta-data
- Resources are visible in the ~~Project Explorer~~ View
  Fortran Projects



Editor and Outline View

- Double-click on source file to open editor Fortran

- Outline view is shown for file in editor



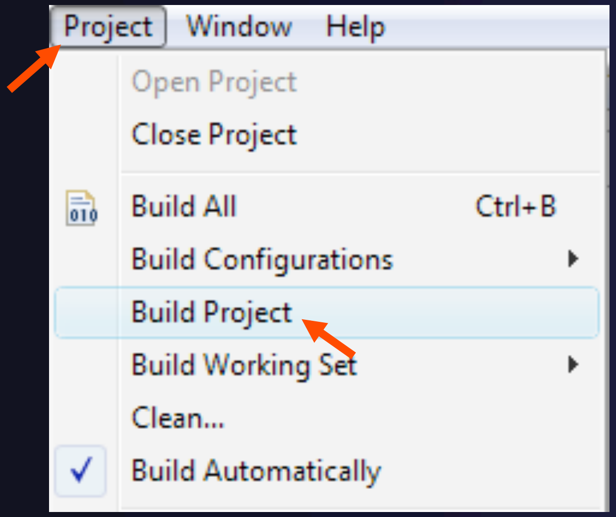
```
----+----1----+----2----+----3----+----4----+----5----+---

use mpi
implicit none

integer, parameter :: LEN = 100                  ! message

integer            :: ierror                     ! error co
integer            :: my_rank                    ! rank of
integer            :: p                          ! number o
integer            :: source                     ! rank of
integer            :: dest                       ! rank of
integer            :: tag                        ! tag for
character(len=LEN) :: message                    ! storage
integer            :: status(MPI_STATUS_SIZE)    ! return s
```

0 items

Description	Resource	Path	Location

Et Cetera

- Building (compiling) is identical



Project Window Help

Open Project

Close Project

Build All Ctrl+B

Build Configurations

Build Project

Build Working Set

Clean...

✓ Build Automatically

Tip: *Are compile errors not shown in the Problems view?*

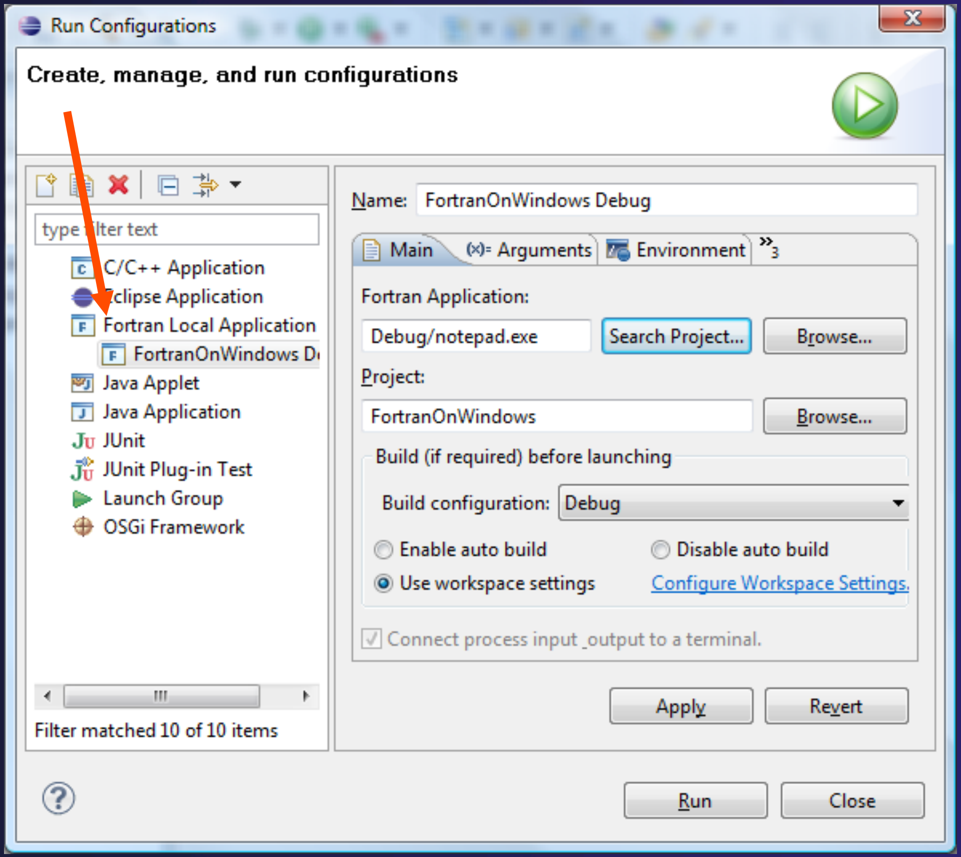
- Right-click on the project in the Fortran Projects view, and choose **Properties**
- Expand **Fortran Build ▸ Settings**
- Switch to the **Error Parsers** tab
- Are Photran's error parsers checked? If not, click **Check all**
- Click **OK** and re-build

Et Cetera

- Creating a launch configuration is identical
  (Suggestion: Uncheck **Stop on startup at main** in the Debugger tab)

Tip: *Is your binary not listed when you create a launch configuration?*

- Right-click on the project in the Fortran Projects view, and choose **Properties**
- Expand **Fortran Build▶Settings**
- Switch to the **Binary Parsers** tab
- Make sure the parser for your platform is checked
  - PE = Windows
  - Elf = Linux
  - Mach-O = Mac OS X
- Click **OK**


Run Configurations

Create, manage, and run configurations

type filter text

Filter matched 10 of 10 items

Name: FortranOnWindows Debug

Fortran Application: Debug/notepad.exe

Project: FortranOnWindows

Build (if required) before launching

Build configuration: Debug

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

☑ Connect process input _output to a terminal.

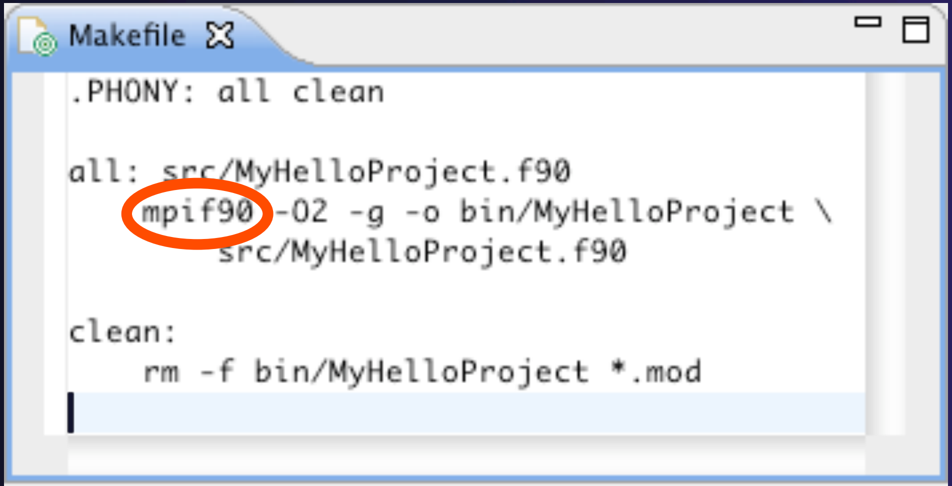
Et Cetera

- Debugging is identical
- Launching a parallel application is identical
- Debugging a parallel debugging is identical

Differences (1): MPI Project Wizard

- In the MPI Hello World C Project, the MPI compiler is set in the project settings… (See "Changing the C/C++ Build Settings Manually" in Module 3)

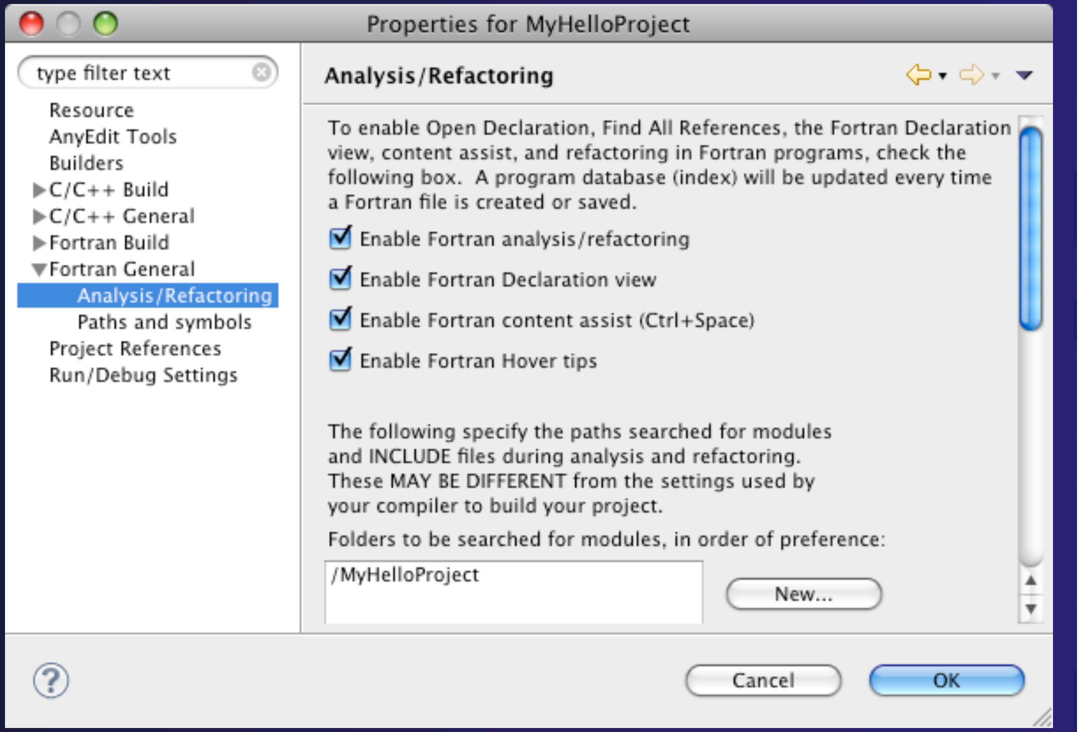
- …but in the MPI Hello World Fortran Project, the MPI compiler is set in a Makefile.



```
.PHONY: all clean

all: src/MyHelloProject.f90
	mpif90 -O2 -g -o bin/MyHelloProject \
		src/MyHelloProject.f90

clean:
	rm -f bin/MyHelloProject *.mod
```

Differences (2): Content Assist

- Content assist is *disabled* by default.
  (So are Declaration View, Hover Tips, Fortran Search, and refactorings.)
  You must specifically enable it for your project.
  - Right-click on the project in the Fortran Projects view, and choose **Properties**
  - Expand **Fortran ▶ Analysis/Refactoring**
  - Check **Enable Fortran analysis/refactoring**
  - Click **OK**
  - Close and re-open any Fortran editors



Analysis/Refactoring

To enable Open Declaration, Find All References, the Fortran Declaration view, content assist, and refactoring in Fortran programs, check the following box. A program database (index) will be updated every time 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

The following specify the paths searched for modules and INCLUDE files during analysis and refactoring. These MAY BE DIFFERENT from the settings used by your compiler to build your project.

Folders to be searched for modules, in order of preference:

/MyHelloProject

Differences (3): Source Form

- Fortran files are either *free form* or *fixed form*
  - Determined by filename extension
  - Extensions are set in the workspace preferences
  - Defaults:

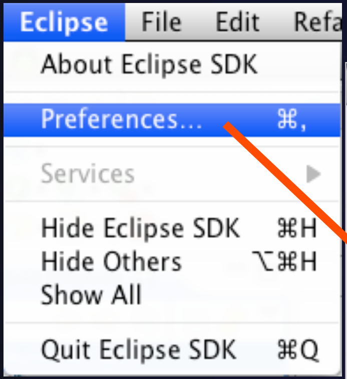
Fixed form:	.f	.fix	.for	.fpp	.ftn
Free form:	.f03	.f95	.f90	.f77	

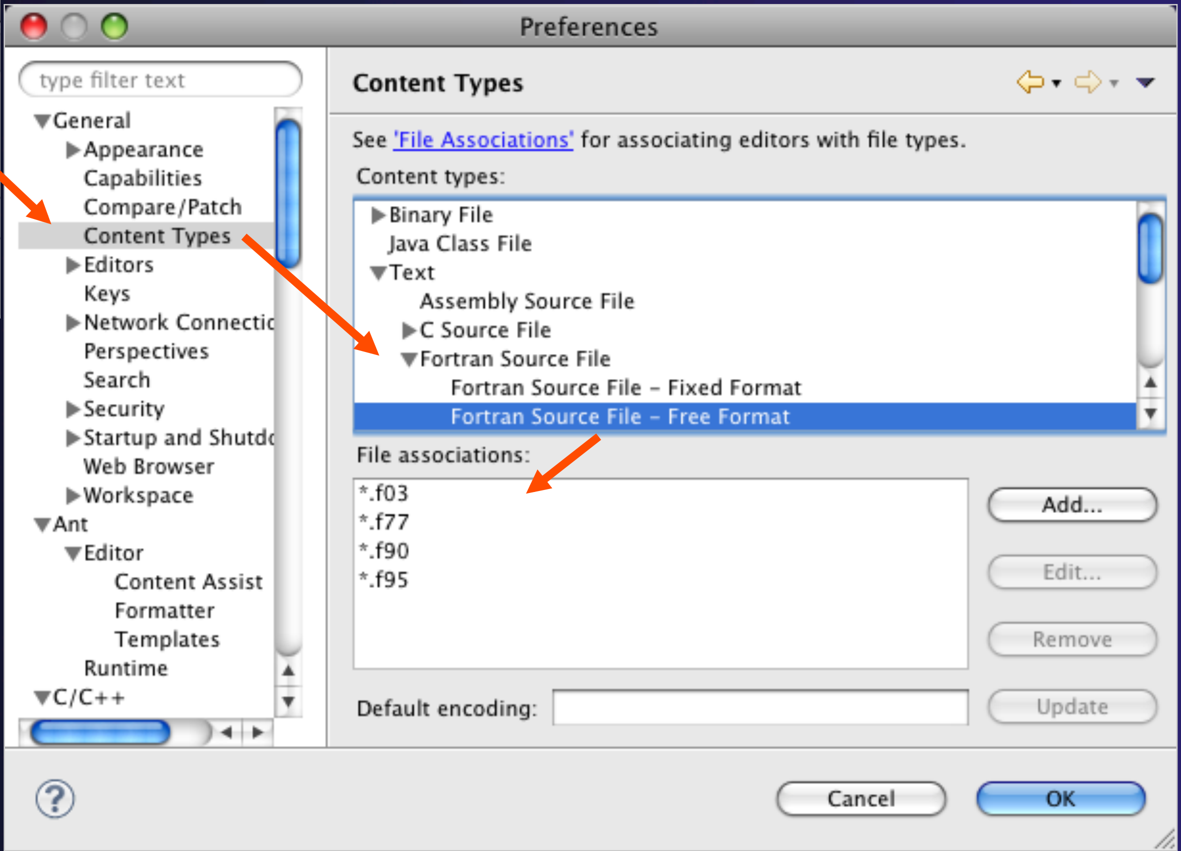
- Many features *will not work* if filename extensions are associated incorrectly

  (Outline view, content assist, Fortran Search, refactorings, Open Declaration, …)

Differences (3): Source Form

Set fixed/free form filename extensions in the preferences





See 'File Associations' for associating editors with file types.

Content types:

- Binary File
- Java Class File
- Text
  - Assembly Source File
  - C Source File
  - Fortran Source File
    - Fortran Source File – Fixed Format
    - Fortran Source File – Free Format

File associations:

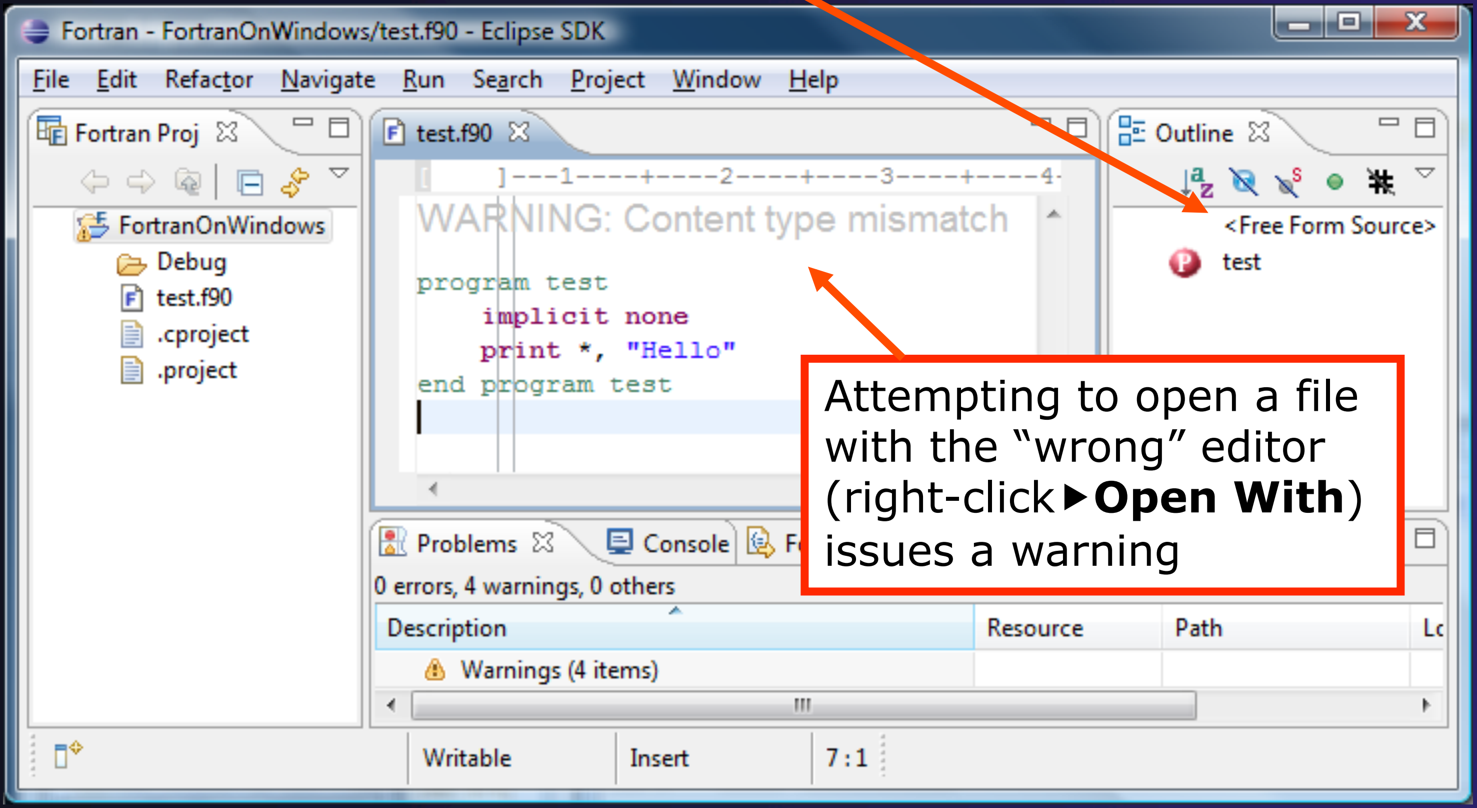
*.f03
*.f77
*.f90
*.f95

Default encoding:

- Navigate the tree to **General ▸ Content Types**
- Expand **Text ▸ Fortran Source File**
- Select **Fixed** or **Free Format** and set associations

Differences (3): Source Form

Outline view displays expected source form of file in editor
(according to the workspace preferences)



```
]---1----+----2----+----3----+----4-
WARNING: Content type mismatch

program test
    implicit none
    print *, "Hello"
end program test
```

0 errors, 4 warnings, 0 others

Warnings (4 items)

Attempting to open a file with the "wrong" editor (right-click▶**Open With**) issues a warning

For More Information

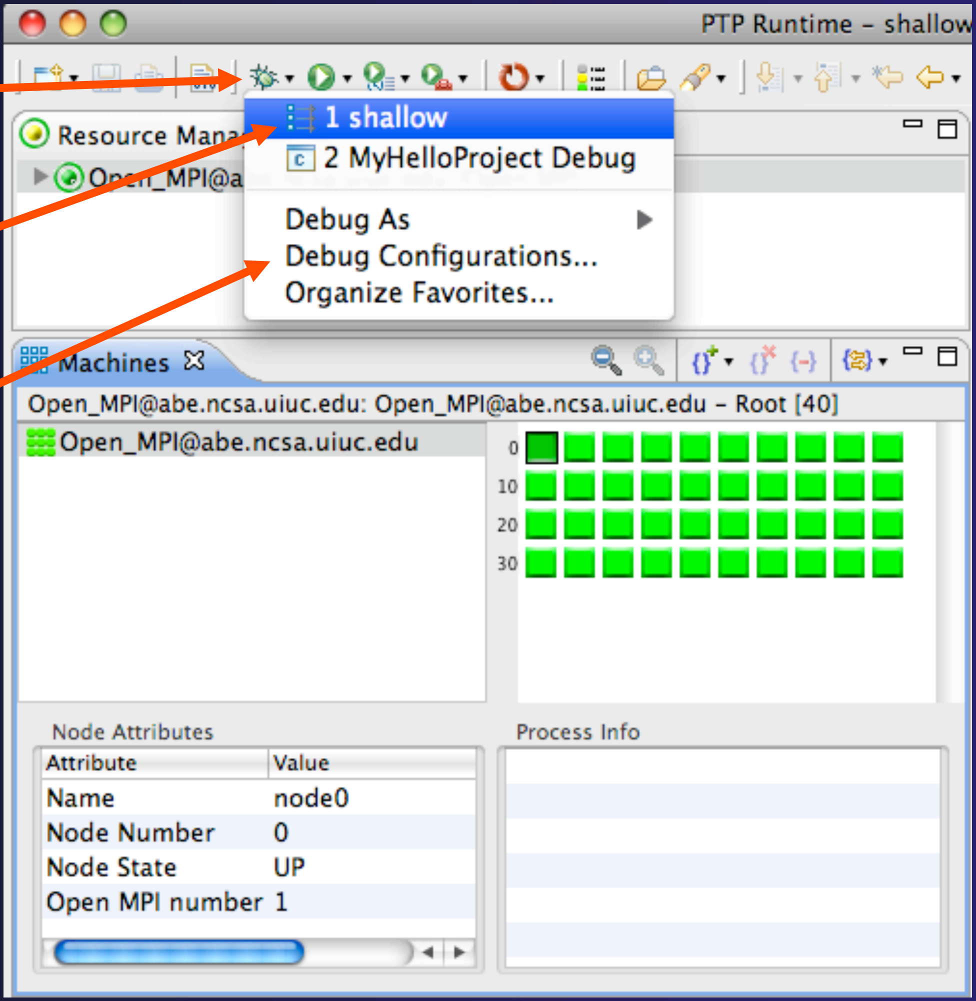
- **Module 7:** Fortran Search, Refactoring
- **Photran online documentation**
  linked from http://www.eclipse.org/photran
  - **User's Guide**
    General introduction, basic features
  - **Advanced Features Guide**
    Features requiring analysis/refactoring to be enabled
- **Online tutorial:** Compiling and running the Parallel Ocean Program using Photran and PTP
  linked from http://wiki.eclipse.org/PTP/photran/tutorials

Module 6: Parallel Debugging

- Objective
  - Learn the basics of debugging parallel programs with PTP
- Contents
  - Launching a parallel debug session
  - The PTP Debug Perspective
  - Controlling sets of processes
  - Controlling individual processes
  - Parallel Breakpoints
  - Terminating processes

Launching A Debug Session

- Use the drop-down next to the debug button (bug icon) instead of run button
- Select the project to launch
- The debug launch will use the same number of processes that the normal launch used (edit the **Debug Launch Configuration** to change)

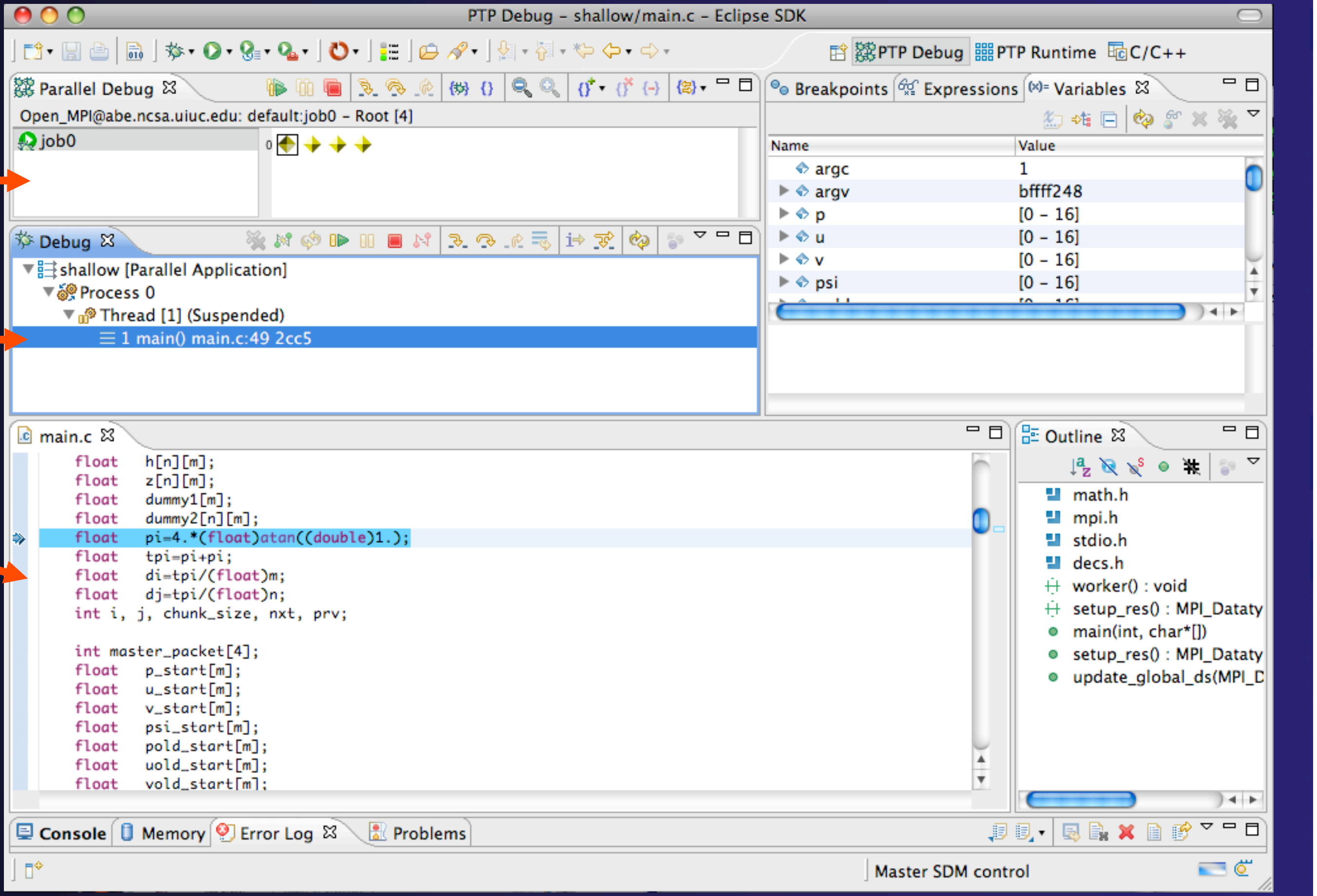


PTP Runtime – shallow

Attribute	Value
Name	node0
Node Number	0
Node State	UP
Open MPI number	1

The PTP Debug Perspective (1)

- **Parallel Debug view** shows job and processes being debugged
- **Debug** view shows threads and call stack for individual processes
- **Source** view shows a **current line marker** for all processes



Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

job0 0

- shallow [Parallel Application]
  - Process 0
    - Thread [1] (Suspended)
      - 1 main() main.c:49 2cc5

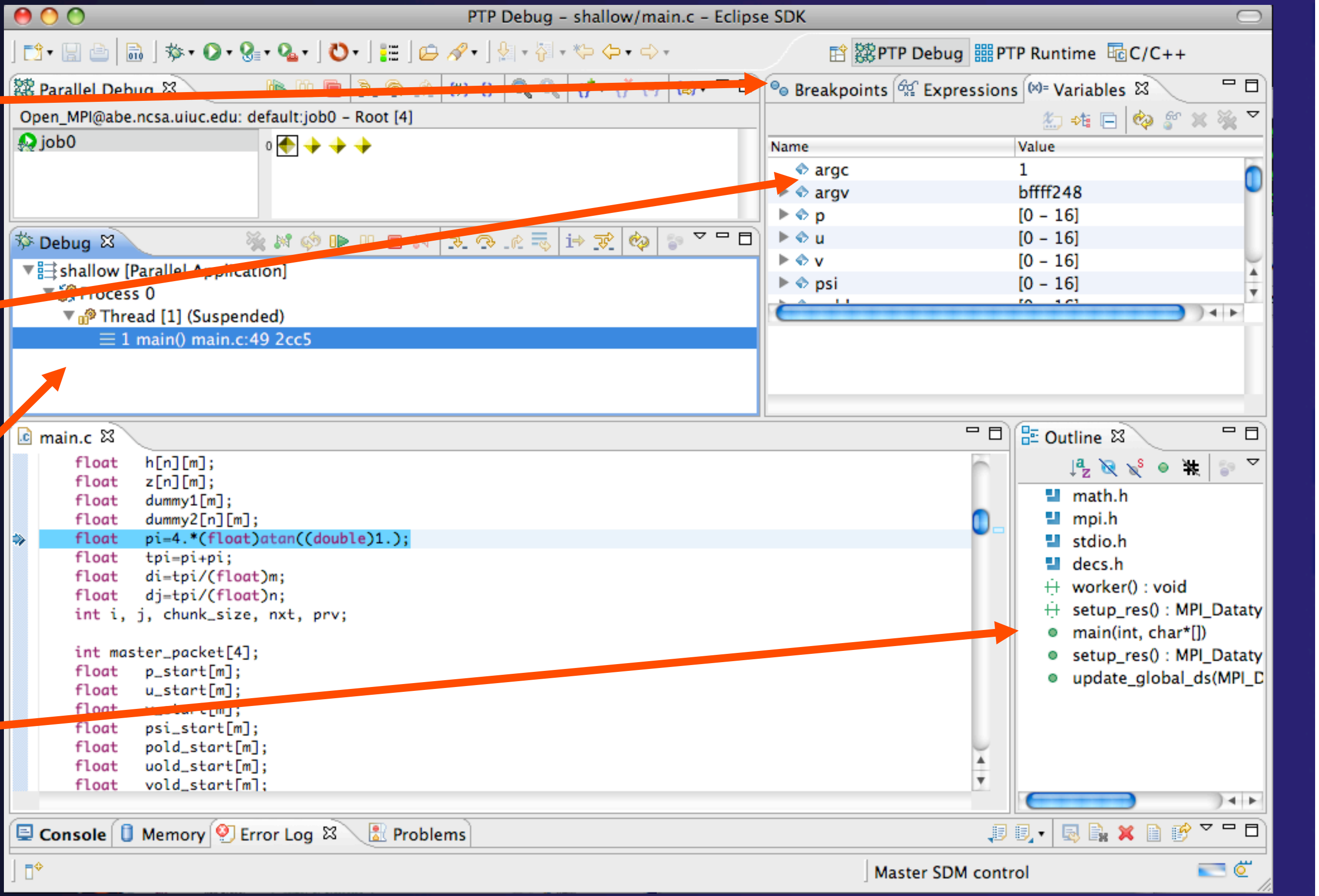
Name	Value
argc	1
argv	bffff248
p	[0 – 16]
u	[0 – 16]
v	[0 – 16]
psi	[0 – 16]

```
float    h[n][m];
float    z[n][m];
float    dummy1[m];
float    dummy2[n][m];
float    pi=4.*(float)atan((double)1.);
float    tpi=pi+pi;
float    di=tpi/(float)m;
float    dj=tpi/(float)n;
int i, j, chunk_size, nxt, prv;

int master_packet[4];
float    p_start[m];
float    u_start[m];
float    v_start[m];
float    psi_start[m];
float    pold_start[m];
float    uold_start[m];
float    vold_start[m];
```

- math.h
- mpi.h
- stdio.h
- decs.h
- worker() : void
- setup_res() : MPI_Dataty
- main(int, char*[])
- setup_res() : MPI_Dataty
- update_global_ds(MPI_D

The PTP Debug Perspective (2)

- **Breakpoints** view shows breakpoints that have been set (more on this later)
- **Variables** view shows the current values of variables for the currently selected process in the **Debug** view
- **Outline** view (from CDT) of source code

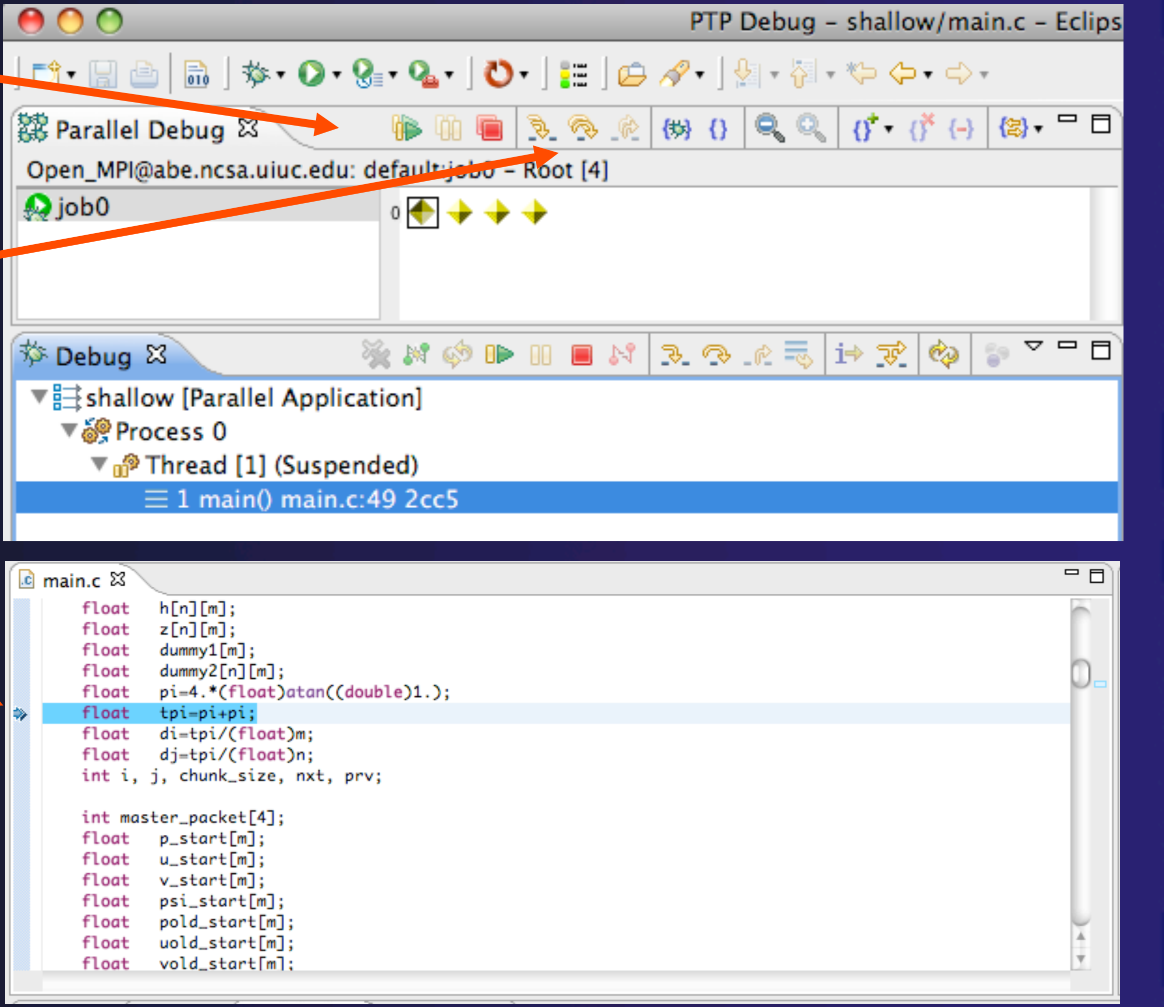


Name	Value
argc	1
argv	bffff248
p	[0 – 16]
u	[0 – 16]
v	[0 – 16]
psi	[0 – 16]

```
    float    h[n][m];
    float    z[n][m];
    float    dummy1[m];
    float    dummy2[n][m];
    float    pi=4.*(float)atan((double)1.);
    float    tpi=pi+pi;
    float    di=tpi/(float)m;
    float    dj=tpi/(float)n;
    int i, j, chunk_size, nxt, prv;

    int master_packet[4];
    float    p_start[m];
    float    u_start[m];
    float    v_start[m];
    float    psi_start[m];
    float    pold_start[m];
    float    uold_start[m];
    float    vold_start[m];
```

Stepping All Processes

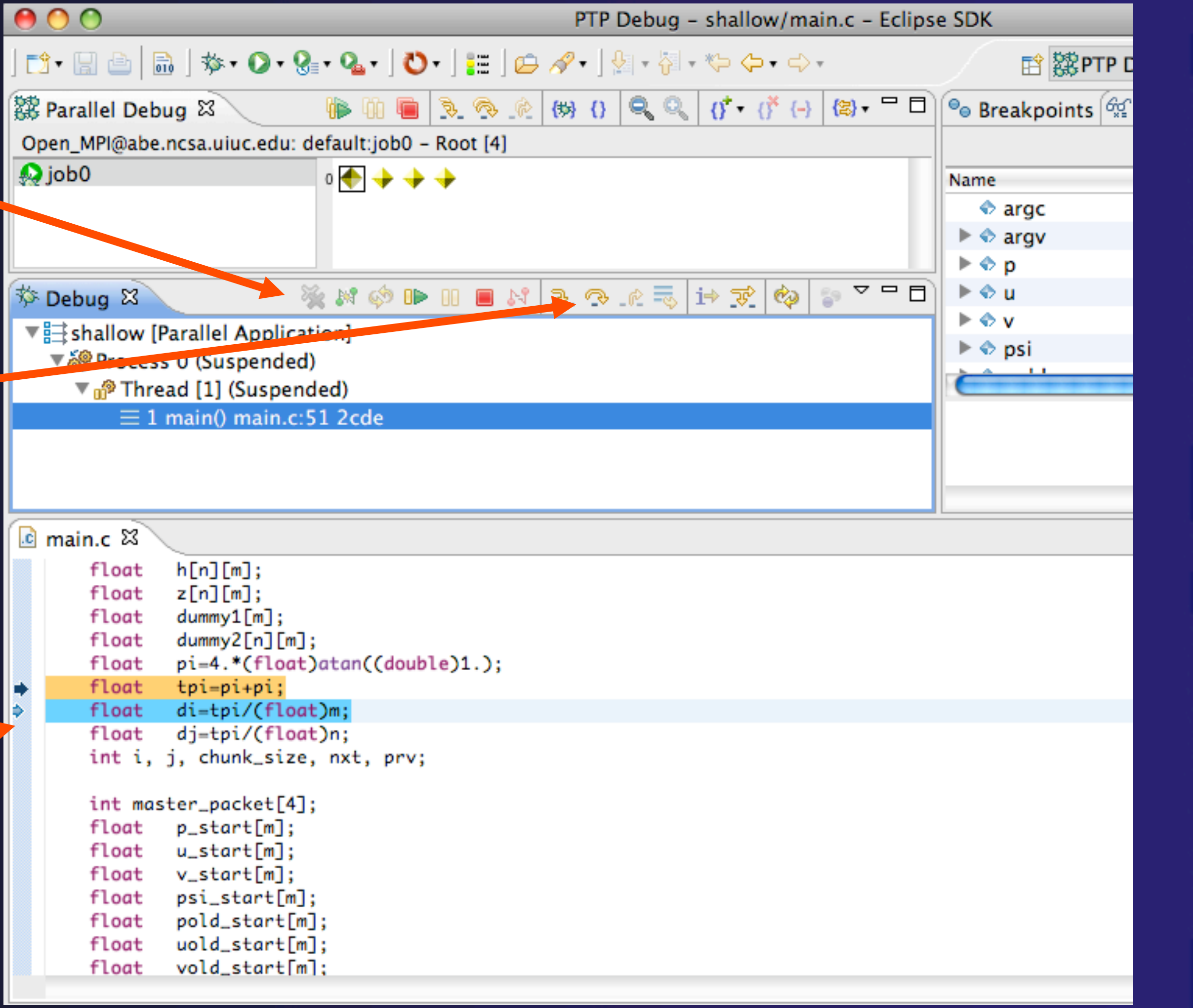
- The buttons in the **Parallel Debug View** control groups of processes
- Click on the **Step Over** button
- Observe that all process icons change to green, then back to yellow
- Notice that the current line marker has moved to the next source line


PTP Debug – shallow/main.c – Eclips
Parallel Debug
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]
job0
0
Debug
shallow [Parallel Application]
Process 0
Thread [1] (Suspended)
1 main() main.c:49 2cc5
main.c

```
    float   h[n][m];
    float   z[n][m];
    float   dummy1[m];
    float   dummy2[n][m];
    float   pi=4.*(float)atan((double)1.);
    float   tpi=pi+pi;
    float   di=tpi/(float)m;
    float   dj=tpi/(float)n;
    int i, j, chunk_size, nxt, prv;

    int master_packet[4];
    float   p_start[m];
    float   u_start[m];
    float   v_start[m];
    float   psi_start[m];
    float   pold_start[m];
    float   uold_start[m];
    float   vold_start[m];
```

Stepping An Individual Process

- The buttons in the **Debug view** are used to control an individual process, in this case process 0
- Click the **Step Over** button
- You will now see two current line markers, the first shows the position of process 0, the second shows the positions of processes 1-3



Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

job0

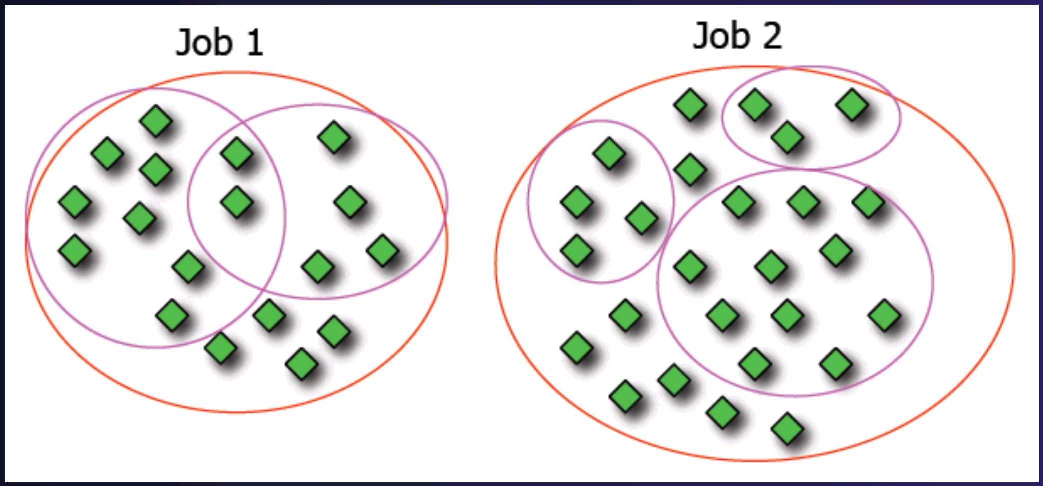
0

shallow [Parallel Application]
Process 0 (Suspended)
Thread [1] (Suspended)
1 main() main.c:51 2cde

```
    float   h[n][m];
    float   z[n][m];
    float   dummy1[m];
    float   dummy2[n][m];
    float   pi=4.*(float)atan((double)1.);
    float   tpi=pi+pi;
    float   di=tpi/(float)m;
    float   dj=tpi/(float)n;
    int i, j, chunk_size, nxt, prv;

    int master_packet[4];
    float   p_start[m];
    float   u_start[m];
    float   v_start[m];
    float   psi_start[m];
    float   pold_start[m];
    float   uold_start[m];
    float   vold_start[m];
```

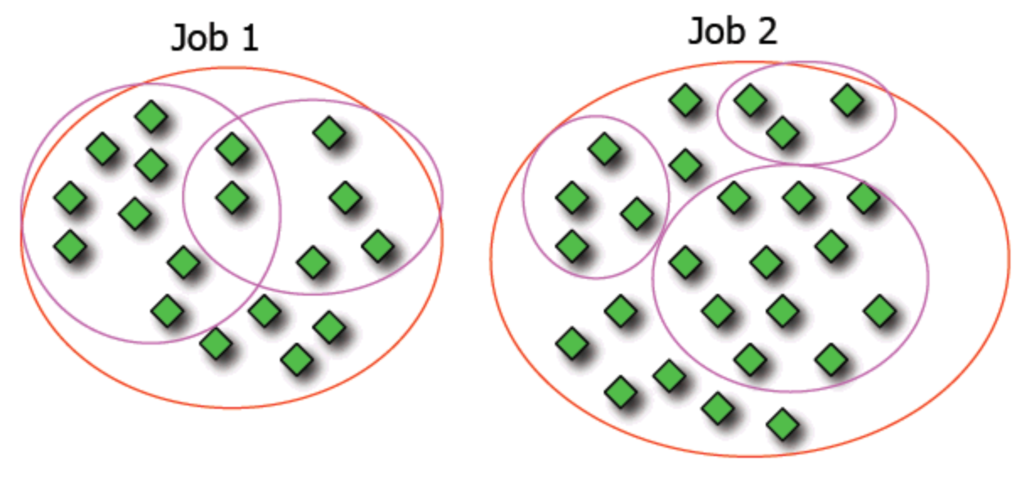
Process Sets (1)

- Traditional debuggers apply operations to a single process
- Parallel debugging operations apply to a single process or to arbitrary collections of processes
- A process set is a means of simultaneously referring to one or more processes



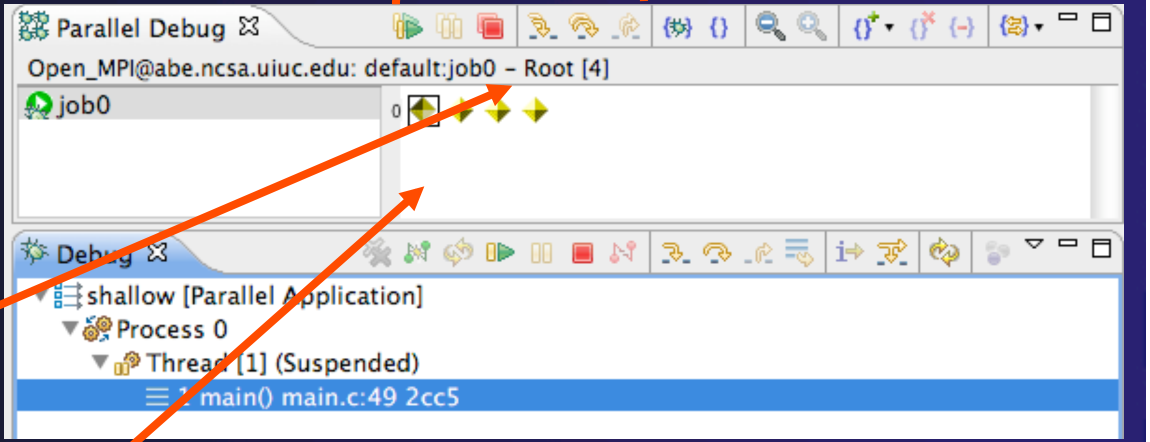
Process Sets (2)

- When a parallel debug session is first started, all processes are placed in a set, called the **Root** set
- Sets are always associated with a single job
- A job can have any number of process sets
- A set can contain from 1 to the number of processes in a job



Operations On Process Sets

- Debug operations on the **Parallel Debug view** toolbar always apply to the current set:
    - Resume, suspend, stop, step into, step over, step return
- The current process set is listed next to job name along with number of processes in the set
- The processes in process set are visible in right hand part of the view



Parallel Debug

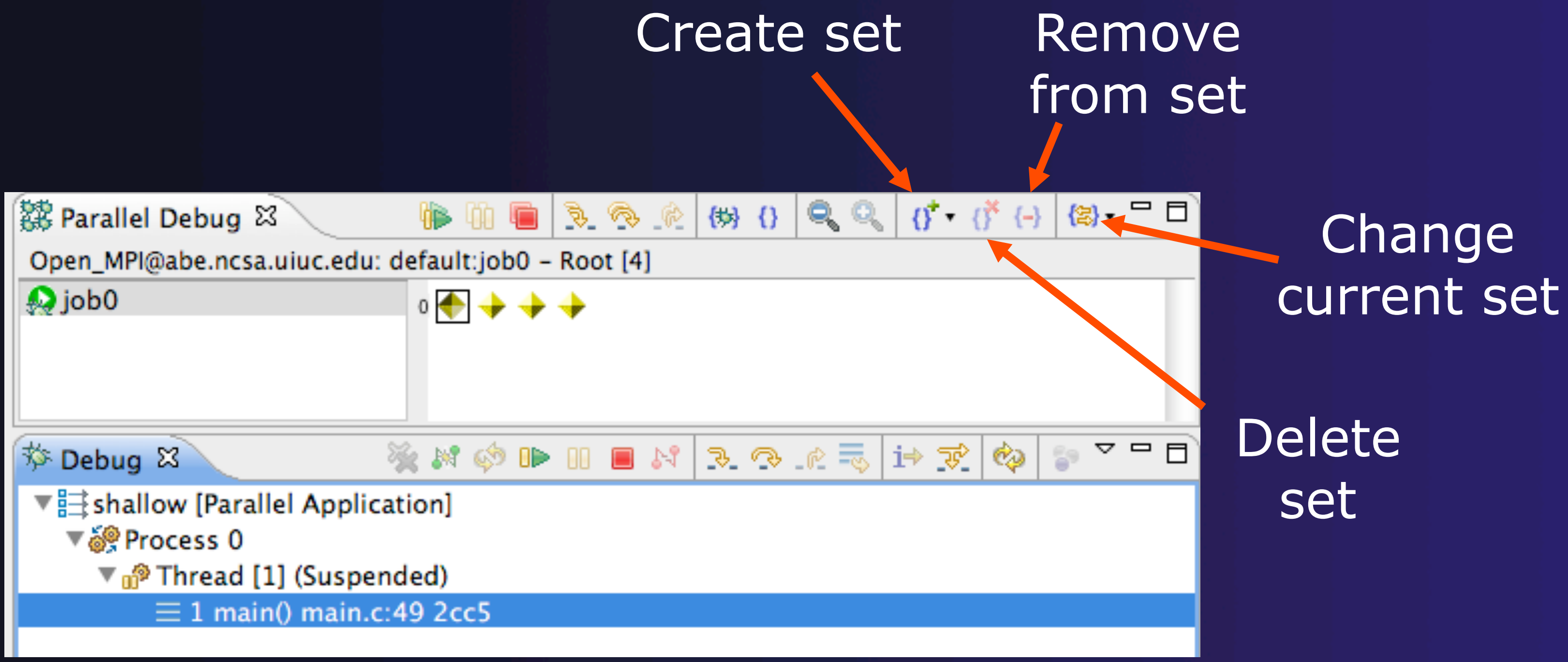
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

shallow [Parallel Application]
Process 0
Thread [1] (Suspended)
1 main() main.c:49 2cc5

Root set = all processes

Managing Process Sets

- The remaining icons in the toolbar of the **Parallel Debug view** allow you to create, modify, and delete process sets, and to change the current process set



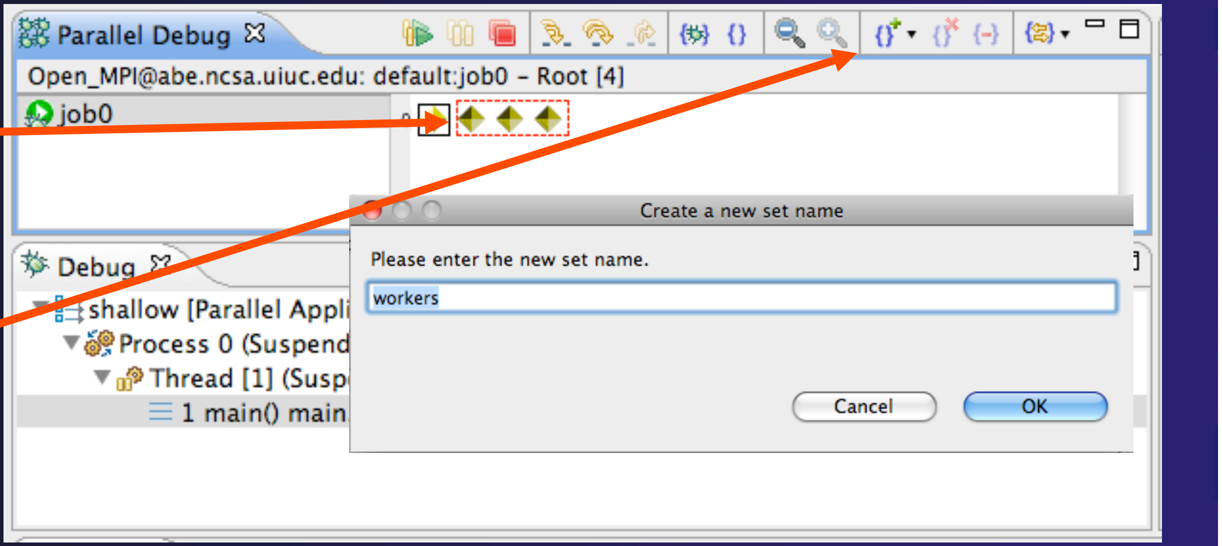
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

job0

shallow [Parallel Application]
Process 0
Thread [1] (Suspended)
1 main() main.c:49 2cc5

Creating A New Process Set

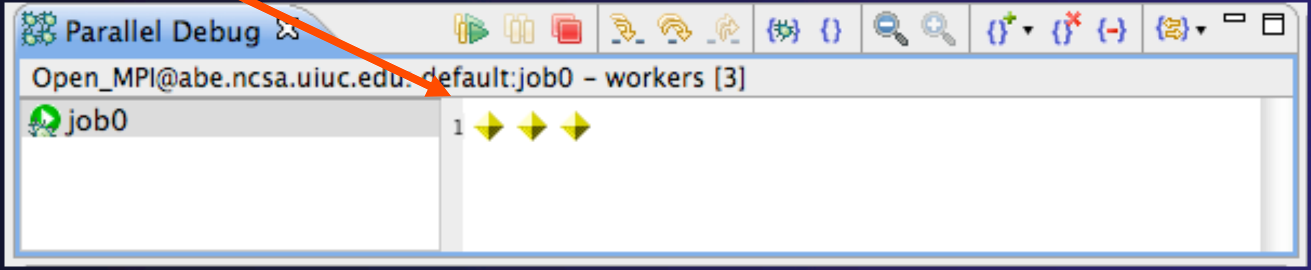
- Select the processes you want in the set by clicking and dragging, in this case, the last three
- Click on the **Create Set** button
- Enter a name for the set, in this case **workers**, and click **OK**
- You will see the view change to display only the selected processes



Open_MPI@abe.ncsa.uiuc.edu: default:job0 - Root [4]

Please enter the new set name.

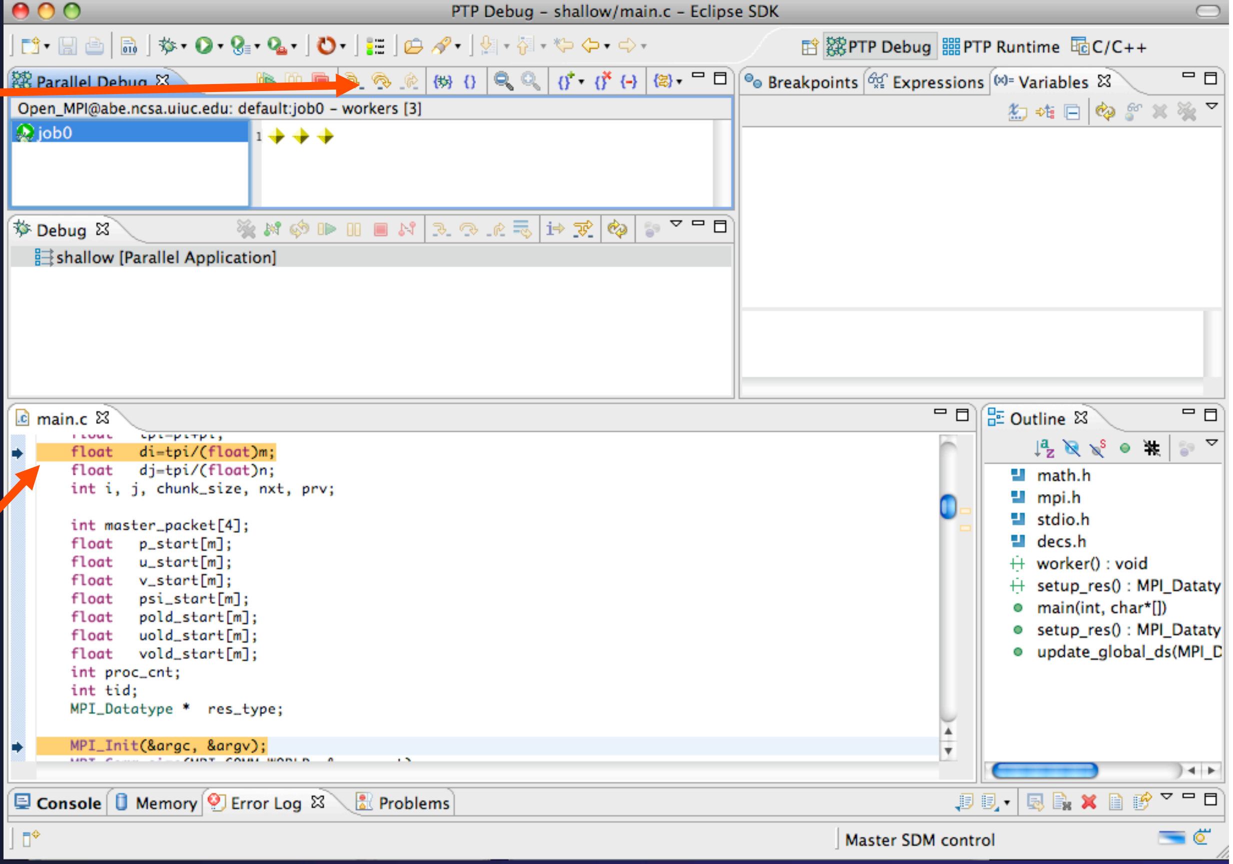
workers



Open_MPI@abe.ncsa.uiuc.edu: default:job0 - workers [3]

Stepping Using New Process Set

- With the **workers** set active, click the **Step Over** button
- You will see only the first current line marker move
- Step a couple more times
- You should see two line markers, one for the single master process, and one for the 3 worker processes



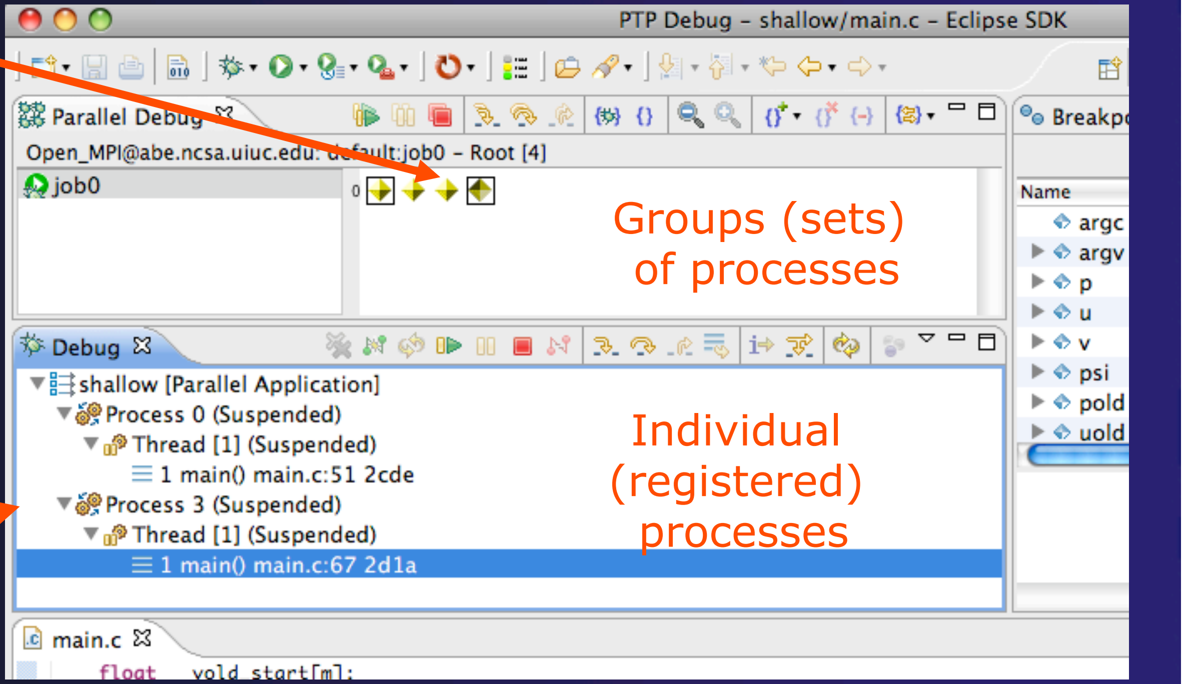
```
    float   di=tpi/(float)m;
    float   dj=tpi/(float)n;
    int i, j, chunk_size, nxt, prv;

    int master_packet[4];
    float   p_start[m];
    float   u_start[m];
    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);
```

Process Registration

- Process set commands apply to groups of processes
- For finer control and more detailed information, a process can be registered and isolated in the **Debug view**
- Registered processes, including their stack traces and threads, appear in the **Debug view**
- Any number of processes can be registered, and processes can be registered or un-registered at any time

Registering A Process

- To register a process, double-click its process icon in the **Parallel Debug view** or select a number of processes and click on the **register** button
- The process icon will be surrounded by a box and the process appears in the **Debug view**
- To un-register a process, double-click on the process or select a number of processes and click on the **unregister** button



Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

job0

Groups (sets) of processes

```
▼ shallow [Parallel Application]
  ▼ Process 0 (Suspended)
    ▼ Thread [1] (Suspended)
        1 main() main.c:51 2cde
  ▼ Process 3 (Suspended)
    ▼ Thread [1] (Suspended)
        1 main() main.c:67 2d1a
```

Individual (registered) processes

argc
argv
p
u
v
psi
pold
uold

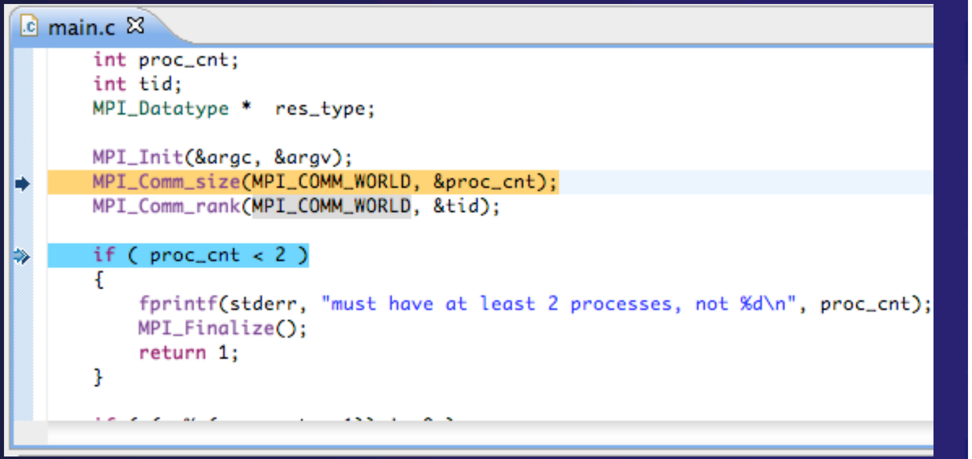
```
float    void start[m];
```

Current Line Marker

- The current line marker is used to show the current location of suspended processes
- In traditional programs, there is a single current line marker (the exception to this is multi-threaded programs)
- In parallel programs, there is a current line marker for every process
- The PTP debugger shows one current line marker for every group of processes at the same location

Colors And Markers

- The highlight color depends on the processes suspended at that line:
  - **Blue:** All registered process(es)
  - **Orange:** All unregistered process(es)
  - **Green:** Registered or unregistered process with no source line (e.g. suspended in a library routine)
- The marker depends on the type of process stopped at that location
- Hover over marker for more details about the processes suspend at that location

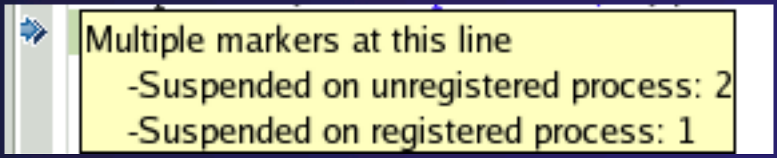


```
    int proc_cnt;
    int tid;
    MPI_Datatype *  res_type;

    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, not %d\n", proc_cnt);
        MPI_Finalize();
        return 1;
    }
```

➡ Multiple processes marker

➡ Registered process marker

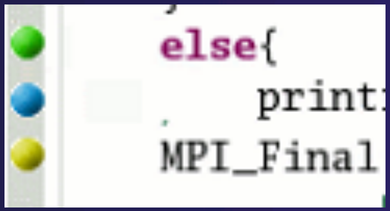
➡ Un-registered process marker



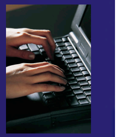
Multiple markers at this line
-Suspended on unregistered process: 2
-Suspended on registered process: 1

Breakpoints

- Apply only to processes in the particular set that is active in the **Parallel Debug view** when the breakpoint is created
- Breakpoints are colored depending on the active process set and the set the breakpoint applies to:
  - Green indicates the breakpoint set is the same as the active set.
  - Blue indicates some processes in the breakpoint set are also in the active set (i.e. the process sets overlap)
  - Yellow indicates the breakpoint set is different from the active set (i.e. the process sets are disjoint)
- When the job completes, the breakpoints are automatically removed

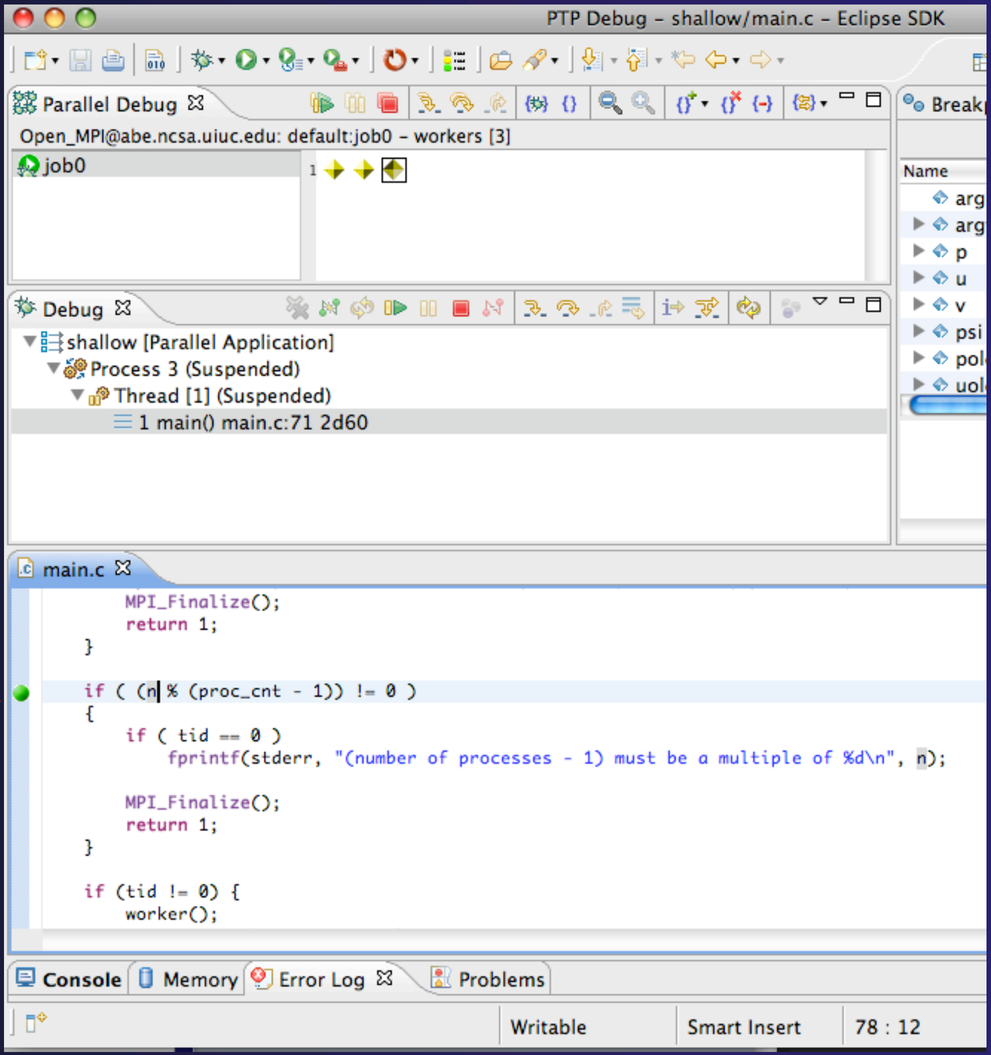


```
● else{
●     print
● MPI_Final
```



Creating A Breakpoint

- Select the process set that the breakpoint should apply to, in this case, the **workers** set
- Double-click on the left edge of an editor window, at the line on which you want to set the breakpoint, or right click and use the **Parallel Breakpoint▶Toggle Breakpoint** context menu
- The breakpoint is displayed on the marker bar



PTP Debug – shallow/main.c – Eclipse SDK

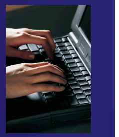
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – workers [3]

job0

shallow [Parallel Application]
Process 3 (Suspended)
Thread [1] (Suspended)
1 main() main.c:71 2d60

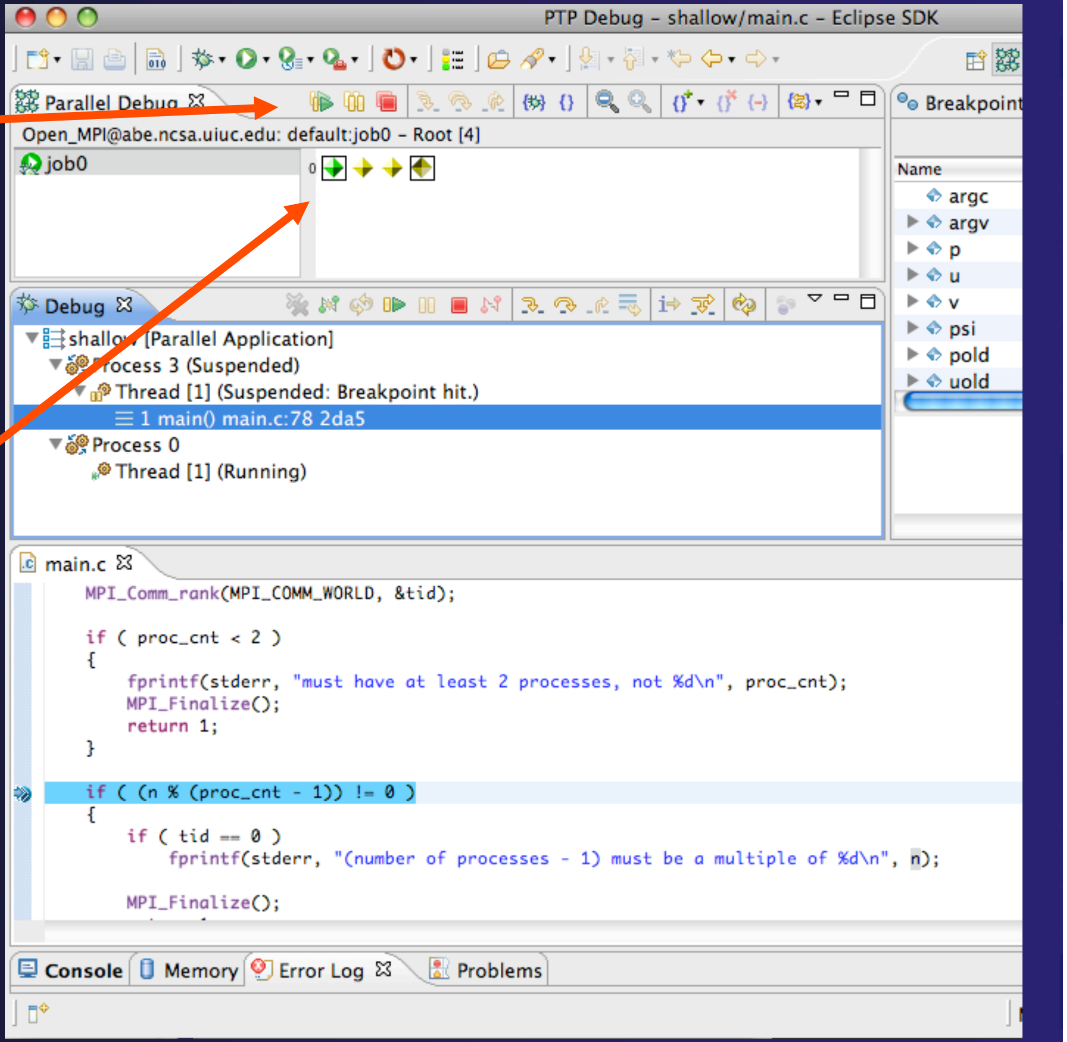
```
        MPI_Finalize();
        return 1;
    }

    if ( (n % (proc_cnt - 1)) != 0 )
    {
        if ( tid == 0 )
            fprintf(stderr, "(number of processes - 1) must be a multiple of %d\n", n);

        MPI_Finalize();
        return 1;
    }

    if (tid != 0) {
        worker();
```

Writable | Smart Insert | 78 : 12



Hitting the Breakpoint

- Click on the **Resume** button in the **Parallel Debug view**
- In this example, the three worker processes have hit the breakpoint, as indicated by the yellow process icons and the current line marker
- Process 0 is still running as its icon is green
- Processes 1-3 are suspended on the breakpoint



```
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]
job0
0
```

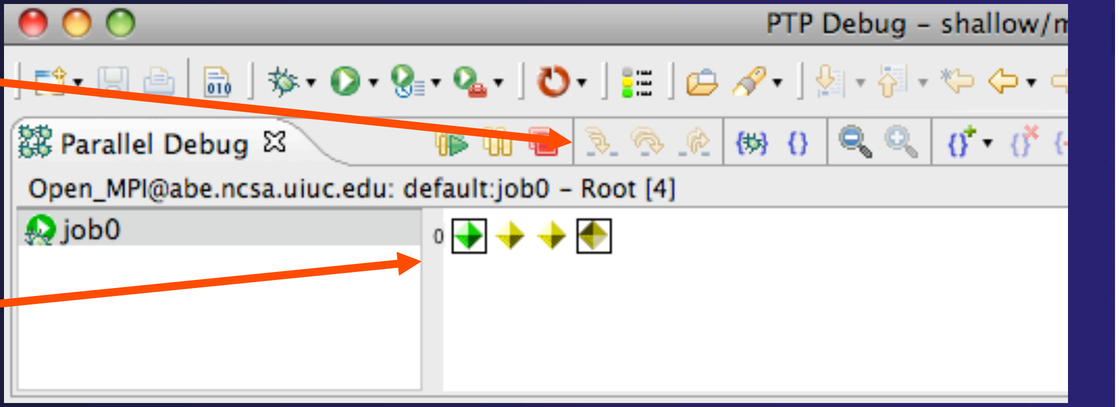
```
Name
argc
argv
p
u
v
psi
pold
uold
```

```
shallow [Parallel Application]
  Process 3 (Suspended)
    Thread [1] (Suspended: Breakpoint hit.)
      1 main() main.c:78 2da5
  Process 0
    Thread [1] (Running)
```

```
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();
    return 1;
}

if ( (n % (proc_cnt - 1)) != 0 )
{
    if ( tid == 0 )
        fprintf(stderr, "(number of processes - 1) must be a multiple of %d\n", n);

    MPI_Finalize();
```

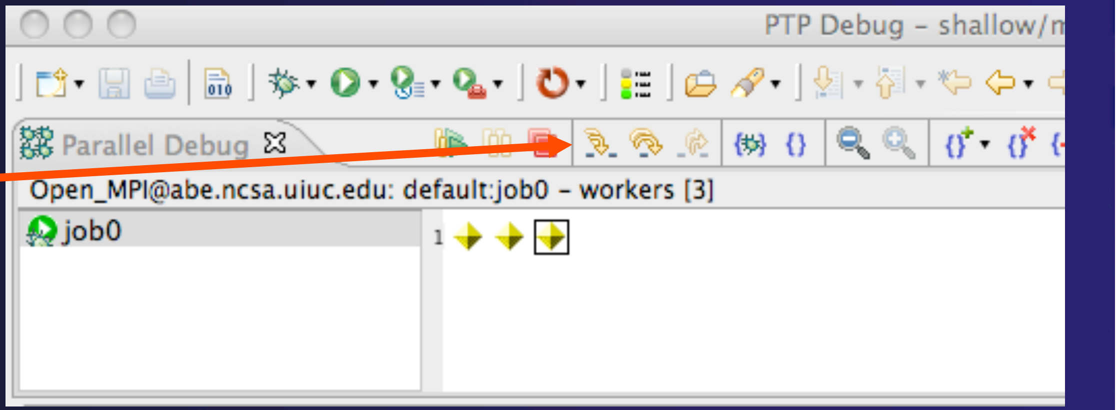
More On Stepping

- The **Step** buttons are only enabled when all processes in the active set are **suspended** (yellow icon)
- In this case, process 0 is still running



PTP Debug – shallow/m

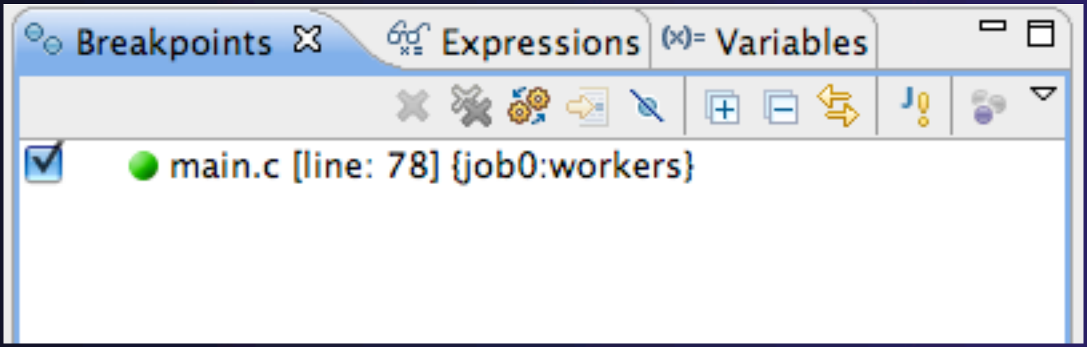
- Switch to the set of suspended processes (the **workers** set)
- You will now see the **Step** buttons become enabled



PTP Debug – shallow/m

Breakpoint Information

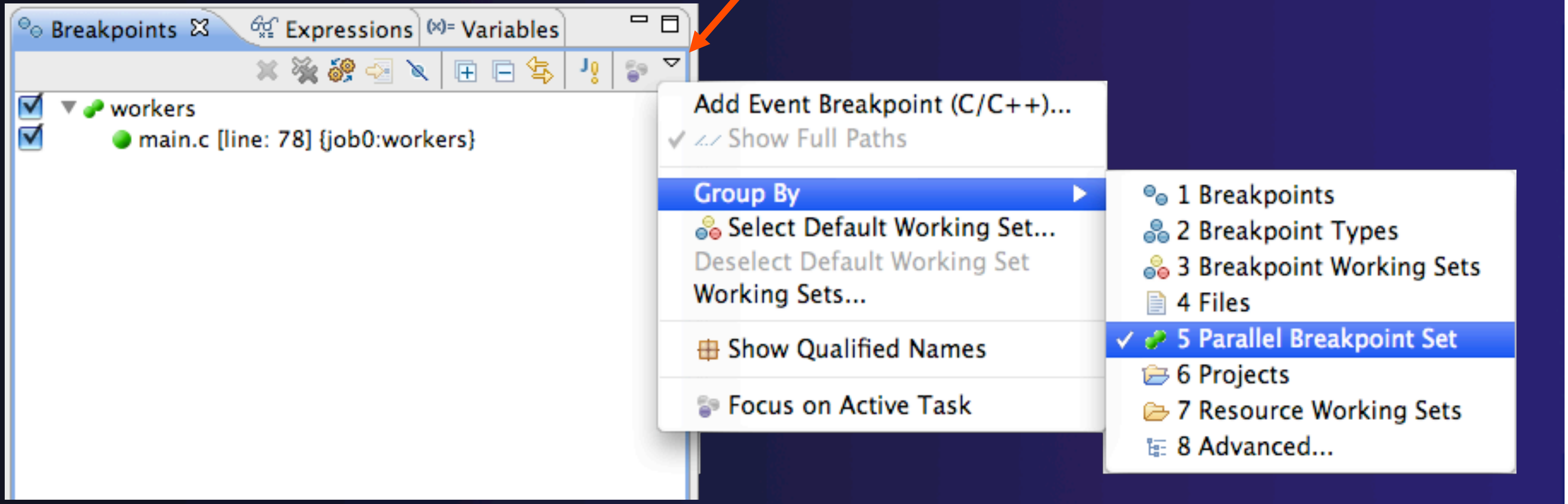
- Hover over breakpoint icon
  - Will show the sets this breakpoint applies to
- Select **Breakpoints** view
  - Will show all breakpoints in all projects



☑ ● main.c [line: 78] {job0:workers}

Breakpoints View

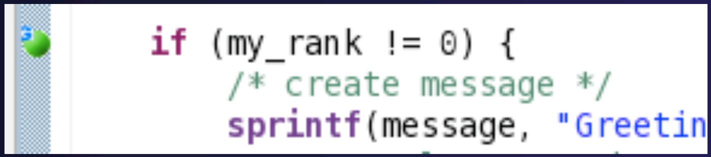
- Use the menu in the breakpoints view to group breakpoints by type
- Breakpoints sorted by breakpoint set (process set)



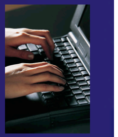
☑ ▼ workers
☑ ● main.c [line: 78] {job0:workers}

Global Breakpoints

- Apply to all processes and all jobs
- Used for gaining control at debugger startup
- To create a global breakpoint
  - First make sure that no jobs are selected (click in white part of jobs view if necessary)
  - Double-click on the left edge of an editor window
  - Note that if a job is selected, the breakpoint will apply to the current set

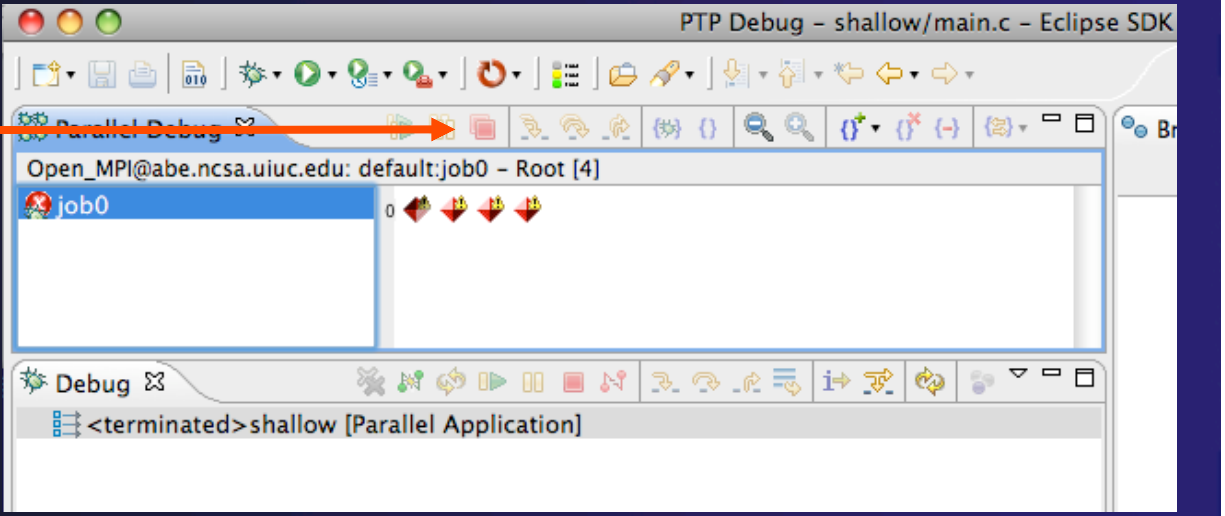


```
    if (my_rank != 0) {
        /* create message */
        sprintf(message, "Greetin
```



Terminating A Debug Session

- Click on the **Terminate** icon in the **Parallel Debug view** to terminate all processes in the active set
- Make sure the **Root** set is active if you want to terminate all processes



PTP Debug – shallow/main.c – Eclipse SDK

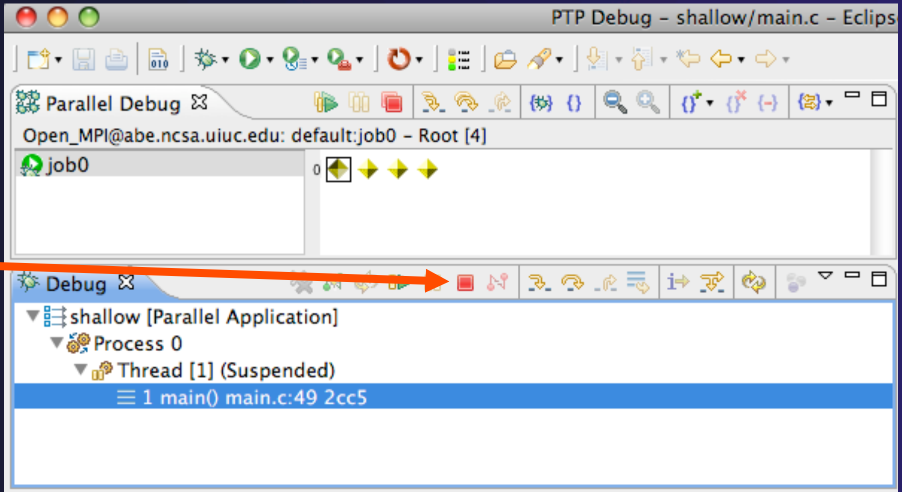
Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

job0

0

<terminated>shallow [Parallel Application]

- You can also use the terminate icon in the **Debug** view to terminate the currently selected process



PTP Debug – shallow/main.c – Eclips

Open_MPI@abe.ncsa.uiuc.edu: default:job0 – Root [4]

job0

0

```
▼ shallow [Parallel Application]
  ▼ Process 0
    ▼ Thread [1] (Suspended)
        1 main() main.c:49 2cc5
```

Module 7: Advanced Development

- Objective
  - Explore some of the advanced features of Eclipse and PTP
- Contents
  - Advanced Eclipse Concepts (generic, not CDT/PTP)
  - Refactoring and Search in Fortran and C/C++
  - Parallel Language Development Tools: MPI, OpenMP, UPC
    - Special Tools for parallel development
  - Remote Development

Advanced Eclipse Concepts

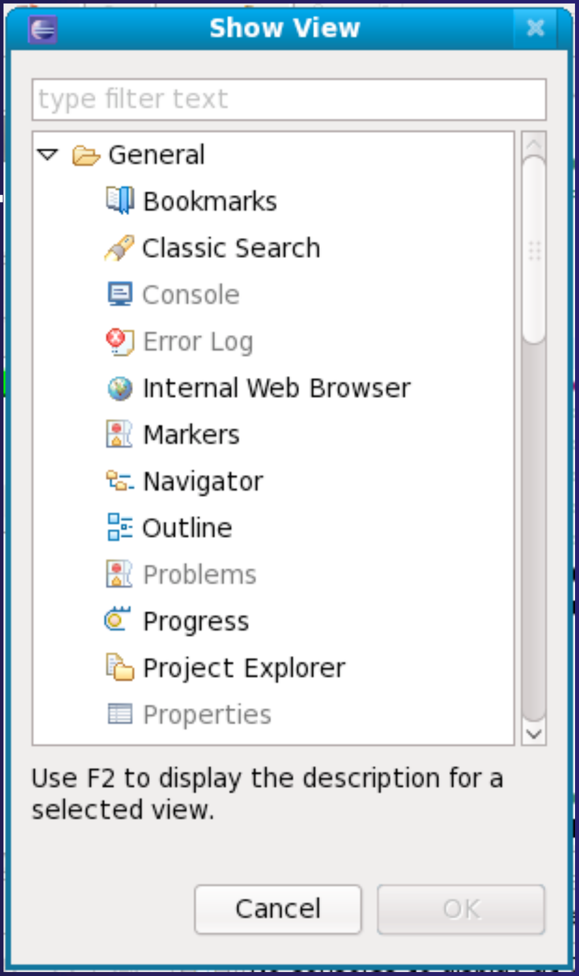
- Perspectives, views and customizing
- Workbench Preferences
- Version Control
- Task Tags

Customizing Perspectives

- Items such as shortcuts, menu items and views may be customized
  - **Window▸Customize Perspective…**
- Save changes
  - **Window▸Save Perspective As…**
- Close Perspective
  - Right-click on perspective title and select **Close**
- Reset Perspective
  - **Window▸Reset Perspective** resets the current perspective to its default layout

Opening New Views

- To open a view:
  - Choose **Window▶Show View▶Other…**
  - The **Show View** dialog comes up
  - Select the view to be shown
  - Select **OK**



Use F2 to display the description for a selected view.

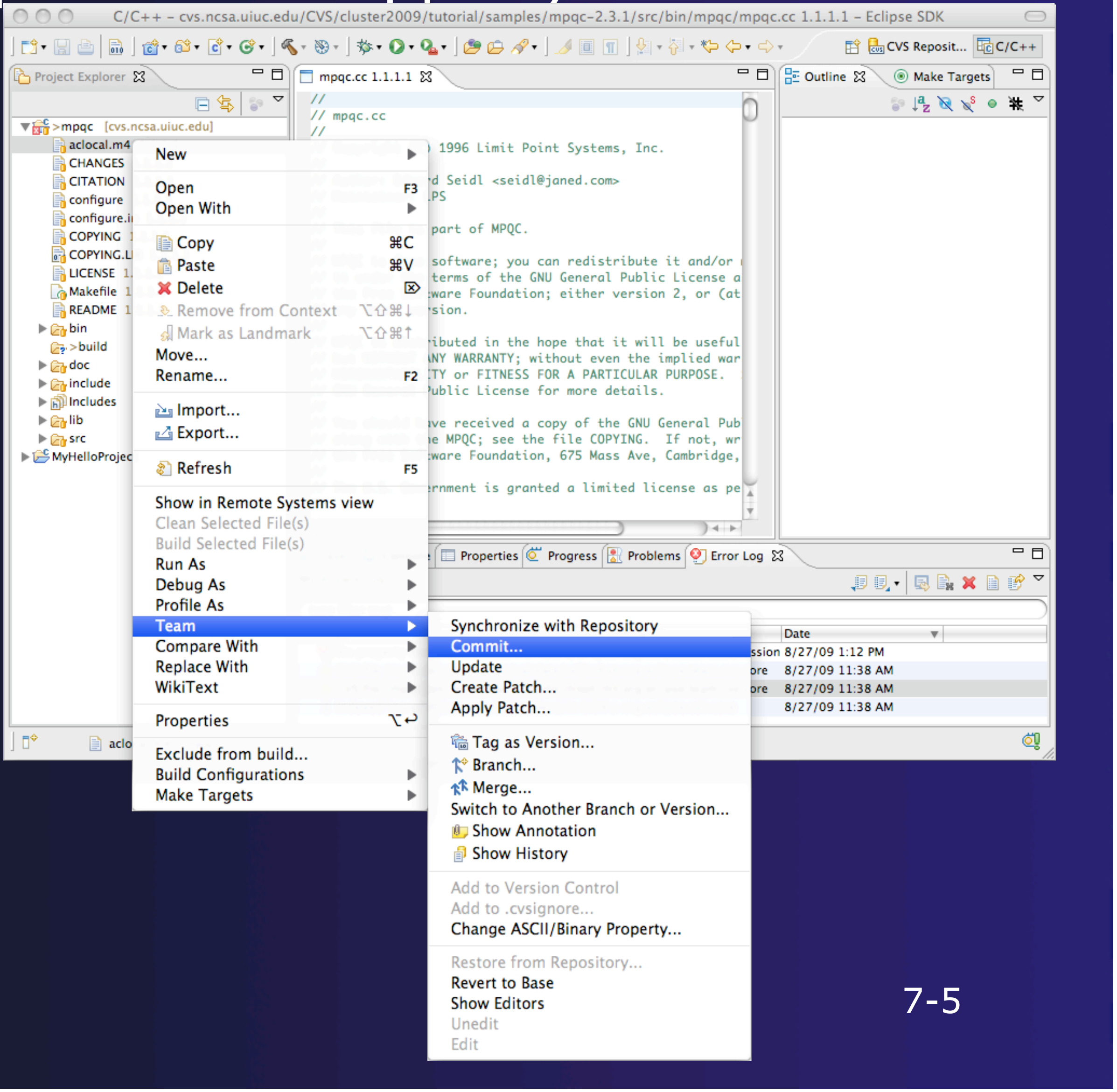
Workbench Preferences

- Preferences provide a way for you to customize your Workbench
  - By selecting **Window▶Preferences…** or **Eclipse▶Preferences…** (Mac)
- Examples of preference settings
  - Use Emacs bindings for editor **keys**
  - Modify editor folding defaults
    - E.g., fold all macro definitions
  - Associate file types with file extensions
    - E.g., *.f03 with the Fortran editor
  - Toggle automatic builds
  - Change key sequence shortcuts
    - E.g., Ctrl+/ for Comment

Version Control

(Eclipse Team Support)

- Version control provided through the **Project Explorer** view, in the **Team** context menu
- Provides familiar actions:
  - **Commit…**
  - **Update…**
- Also less used tasks:
  - **Create/Apply Patch…**
  - **Tag as Version…**
  - **Branch…**
  - **Merge…**
  - **Add to .cvsignore…**

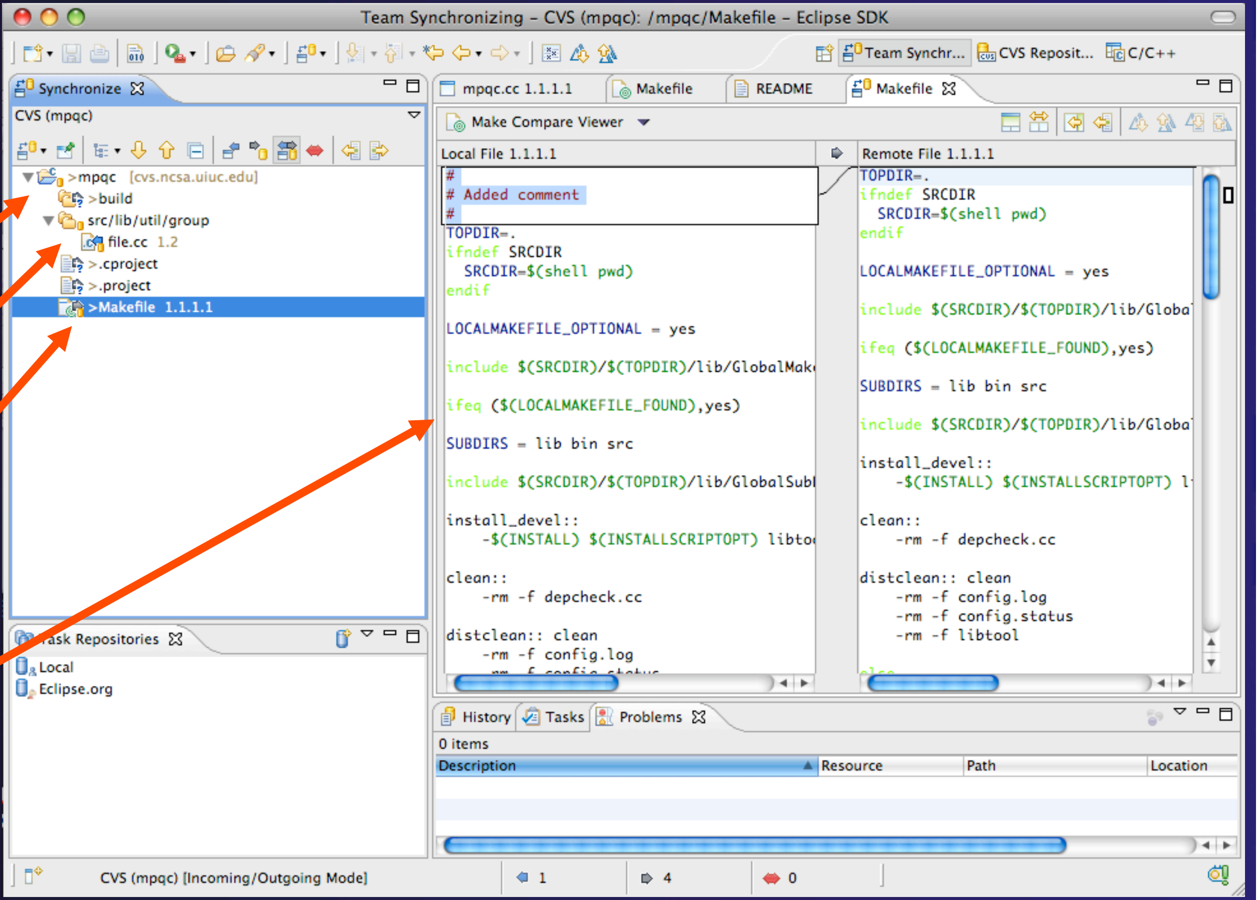


```
//
// mpqc.cc
//
) 1996 Limit Point Systems, Inc.

d Seidl <seidl@janed.com>
PS

part of MPQC.

software; you can redistribute it and/or
terms of the GNU General Public License a
ware Foundation; either version 2, or (at
sion.

ibuted in the hope that it will be useful
NY WARRANTY; without even the implied war
TY or FITNESS FOR A PARTICULAR PURPOSE.
ublic License for more details.

ve received a copy of the GNU General Pub
e MPQC; see the file COPYING.  If not, wr
ware Foundation, 675 Mass Ave, Cambridge,

rnment is granted a limited license as pe
```

	Date
ssion	8/27/09 1:12 PM
ore	8/27/09 11:38 AM
ore	8/27/09 11:38 AM
	8/27/09 11:38 AM

Team Synchronizing

- Accessed from the **Team ▶ Synchronize with Repository** context menu
- Shows:
  - Files to be added
  - Files to be updated
  - Files to be committed
  - Files to be deleted
  - Merge conflicts
- Double-click on file to show compare viewer
- Operations can be performed on individual files or all at once



Local File 1.1.1.1

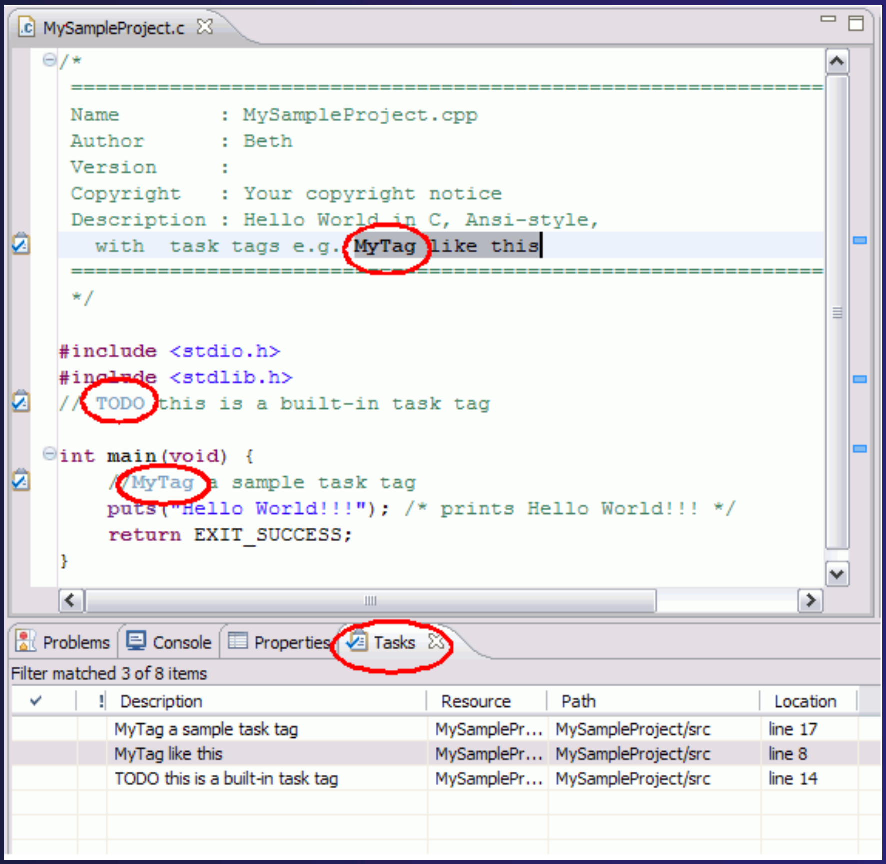
```
#
# Added comment
#
TOPDIR=.
ifndef SRCDIR
  SRCDIR=$(shell pwd)
endif

LOCALMAKEFILE_OPTIONAL = yes

include $(SRCDIR)/$(TOPDIR)/lib/GlobalMak

ifeq ($(LOCALMAKEFILE_FOUND),yes)

SUBDIRS = lib bin src

include $(SRCDIR)/$(TOPDIR)/lib/GlobalSubl

install_devel::
	-$(INSTALL) $(INSTALLSCRIPTOPT) libto

clean::
	-rm -f depcheck.cc

distclean:: clean
	-rm -f config.log
```

Remote File 1.1.1.1

```
TOPDIR=.
ifndef SRCDIR
  SRCDIR=$(shell pwd)
endif

LOCALMAKEFILE_OPTIONAL = yes

include $(SRCDIR)/$(TOPDIR)/lib/Globa

ifeq ($(LOCALMAKEFILE_FOUND),yes)

SUBDIRS = lib bin src

include $(SRCDIR)/$(TOPDIR)/lib/Globa

install_devel::
	-$(INSTALL) $(INSTALLSCRIPTOPT) l

clean::
	-rm -f depcheck.cc

distclean:: clean
	-rm -f config.log
	-rm -f config.status
	-rm -f libtool
```

Task Tags

- ✦ Task tags are identifiers in C/C++ comments
- ✦ TODO is a built-in task tag
- ✦ The build locates task tags during compilation
- ✦ View task tags in Tasks View
  - ✦ If it's not shown, **Window ▶ Show View ▶ Other…** Open **General** and select **Tasks**
- ✦ Configure your own task tag in **Window ▶ Preferences**
  - ✦ Under C/C++, select Task Tags



```
/*
 ============================================================
 Name        : MySampleProject.cpp
 Author      : Beth
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style,
   with  task tags e.g. MyTag like this
 ============================================================
 */

#include <stdio.h>
#include <stdlib.h>
// TODO this is a built-in task tag

int main(void) {
    //MyTag a sample task tag
    puts("Hello World!!!"); /* prints Hello World!!! */
    return EXIT_SUCCESS;
}
```

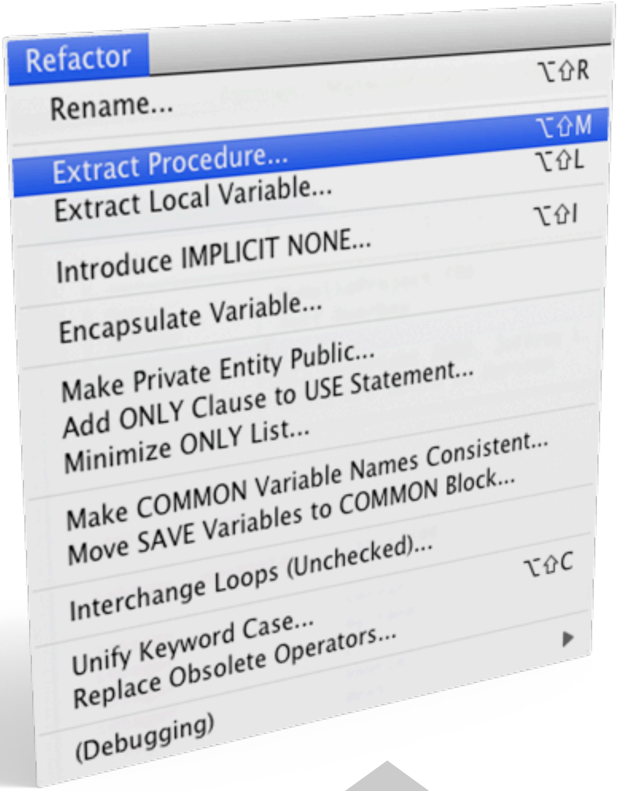
Filter matched 3 of 8 items

✓	!	Description	Resource	Path	Location
		MyTag a sample task tag	MySamplePr…	MySampleProject/src	line 17
		MyTag like this	MySamplePr…	MySampleProject/src	line 8
		TODO this is a built-in task tag	MySamplePr…	MySampleProject/src	line 14

Refactoring
and
Search
in Fortran and C/C++

Refactoring

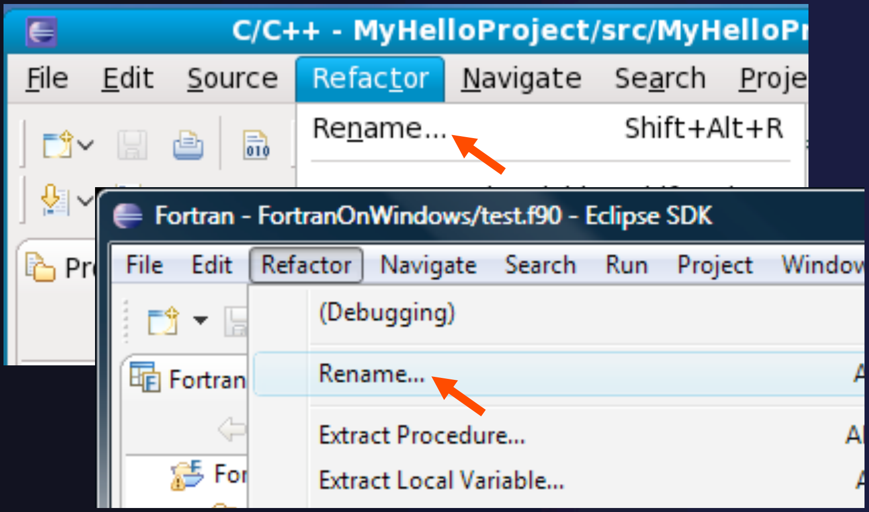
(making changes to source code that don't affect the behavior of the program)


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)...
Unify Keyword Case... ⌥⇧C
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 5.0: 10-15 refactorings

Rename Refactoring

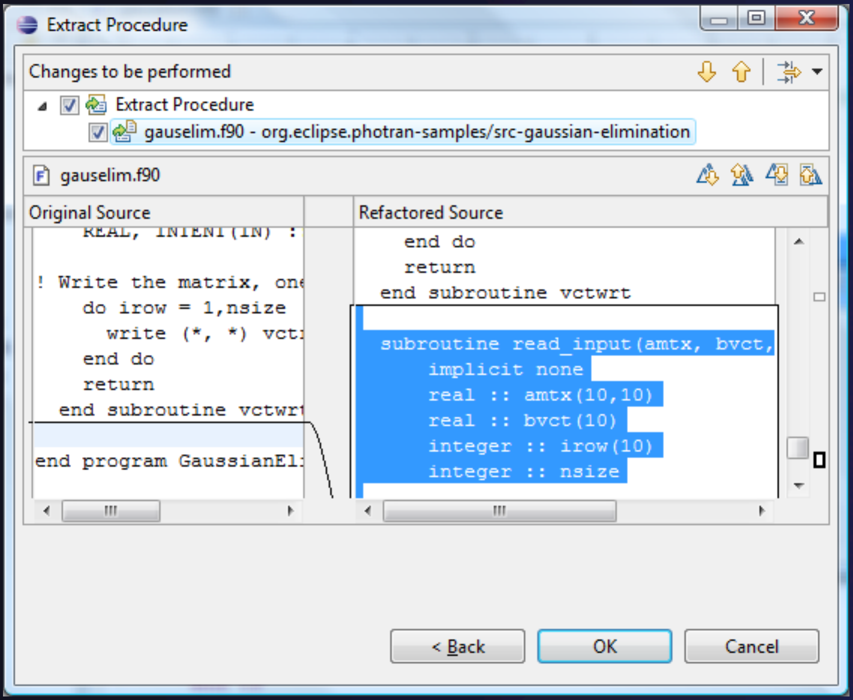
- Changes the name of a variable, function, etc., *including every use*
  (change is semantic, not textual, and can be workspace-wide)
- Only proceeds if the new name will be legal
  (aware of scoping rules, namespaces, etc.)


C/C++ - MyHelloProject/src/MyHelloPr
File Edit Source Refactor Navigate Search Proje
Rename... Shift+Alt+R
Pr
Fortran - FortranOnWindows/test.f90 - Eclipse SDK
File Edit Refactor Navigate Search Run Project Window
(Debugging)
Rename...
Extract Procedure...
Extract Local Variable...
Fortran
For

- Select **C/C++ Perspective** or **Fortran Perspective**
- Open a source file
- Click in editor view on declaration of a variable
- Select menu item **Refactor▶Rename**
  - Or use context menu
- Enter new name

Extract Procedure Refactoring

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



Original Source:

```
    REAL, INTENT(IN) :

! Write the matrix, one
    do irow = 1,nsize
      write (*, *) vct
    end do
    return
  end subroutine vctwrt

end program GaussianEl
```

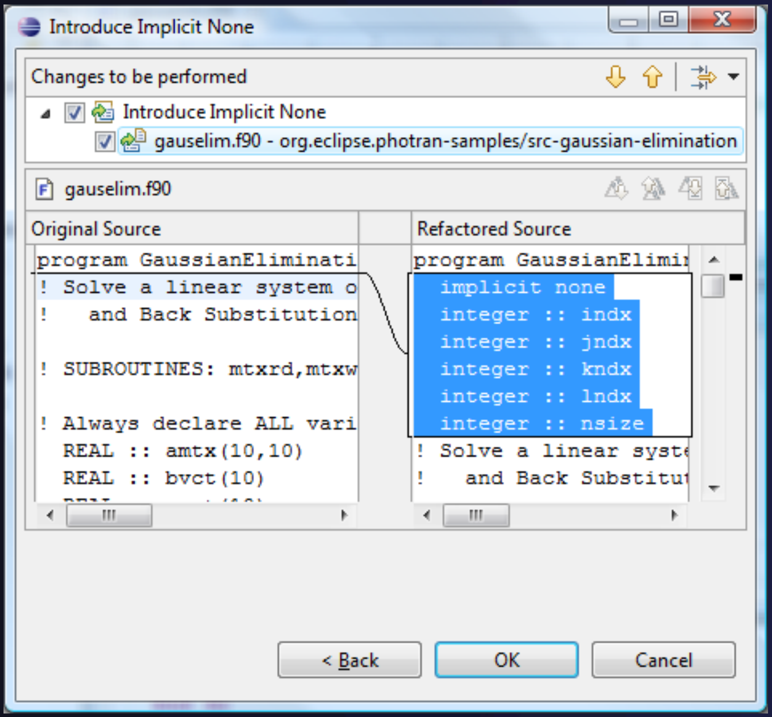
Refactored Source:

```
    end do
    return
  end subroutine vctwrt

  subroutine read_input(amtx, bvct,
      implicit none
      real :: amtx(10,10)
      real :: bvct(10)
      integer :: irow(10)
      integer :: nsize
```

- ✦ Select a sequence of statements
- ✦ Select menu item **Refactor▶Extract Procedure…** in Fortran, or, in C/C++, **Refactor▶Extract Function…**
  - ✦ Or use context menu
- ✦ Enter new name

Photran Implicit Refactoring

- Introduce Implicit None adds an IMPLICIT NONE statement and adds explicit variable declarations for all implicitly declared variables



Original Source:

```
program GaussianEliminati
! Solve a linear system o
!   and Back Substitution

! SUBROUTINES: mtxrd,mtxw

! Always declare ALL vari
  REAL :: amtx(10,10)
  REAL :: bvct(10)
```

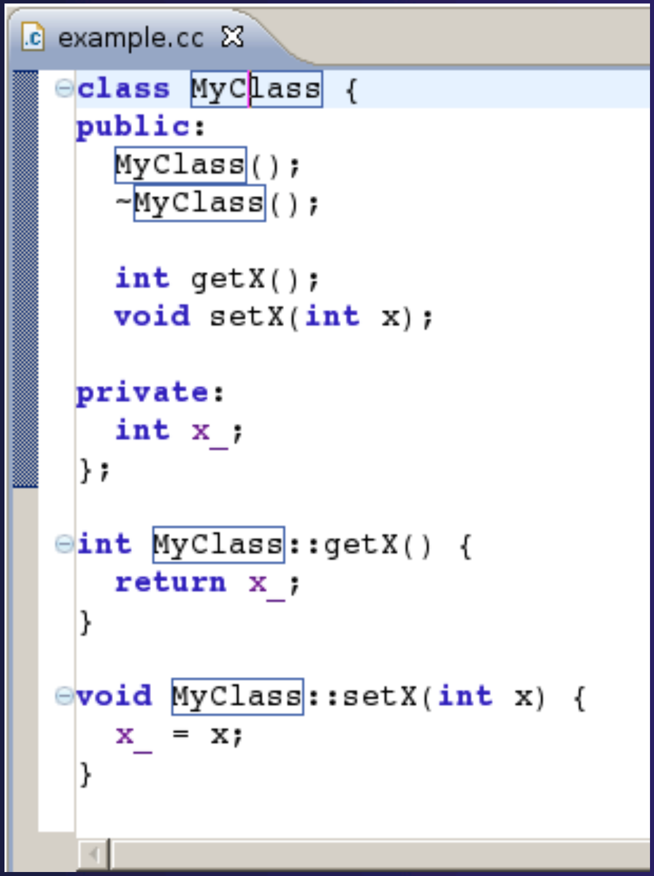
Refactored Source:

```
program GaussianElimi
  implicit none
  integer :: indx
  integer :: jndx
  integer :: kndx
  integer :: lndx
  integer :: nsize
! Solve a linear syste
!   and Back Substitut
```

- Introduce in a single file by opening the file and selecting **Refactor ▸ Introduce IMPLICIT NONE...**
- Introduce in multiple files by selecting them in the Fortran Projects view, right-clicking on the selection, and choosing **Refactor ▸ Introduce IMPLICIT NONE...**

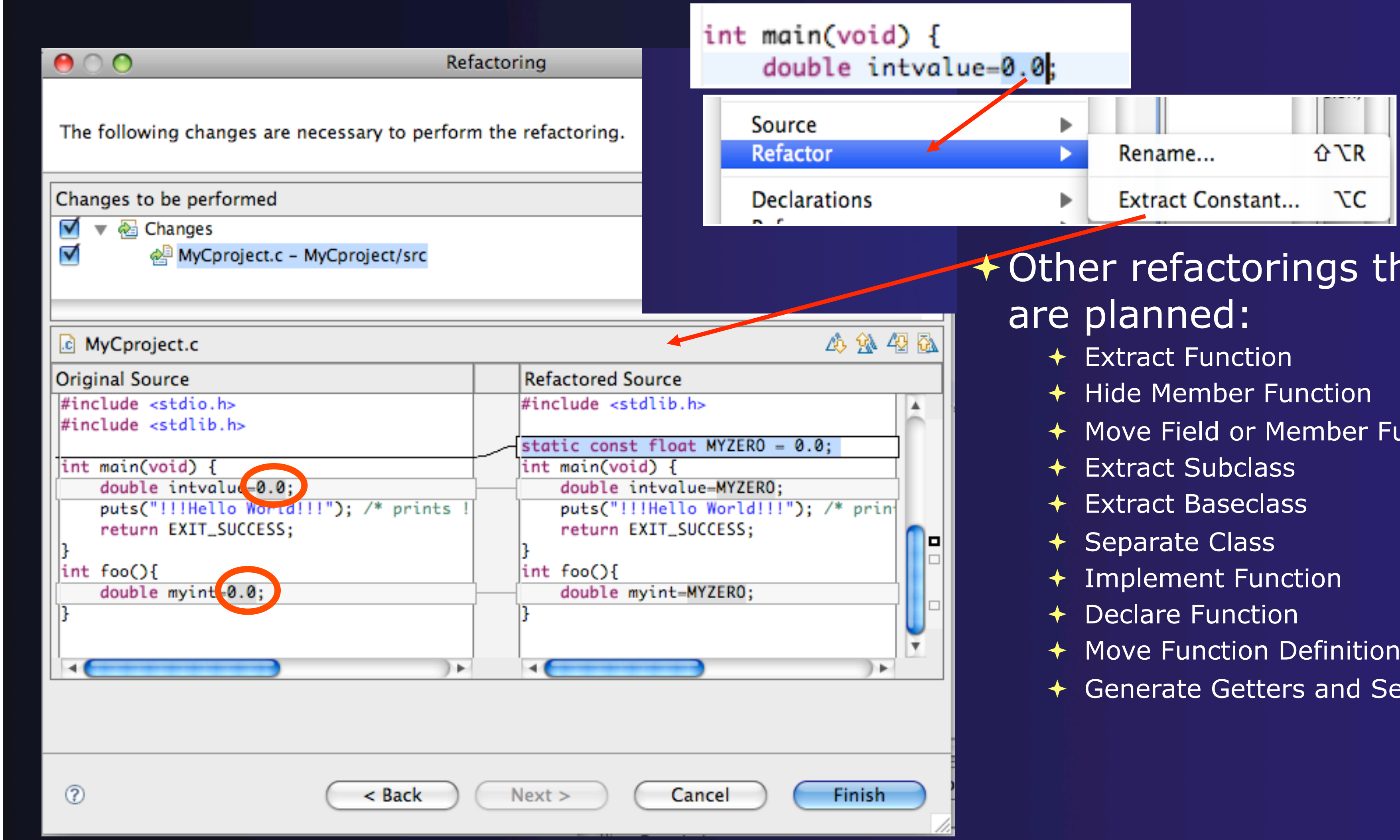
CDT Rename in File

- 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.



```
class MyClass {
public:
  MyClass();
  ~MyClass();

  int getX();
  void setX(int x);

private:
  int x_;
};

int MyClass::getX() {
  return x_;
}

void MyClass::setX(int x) {
  x_ = x;
}
```

CDT Extract Constant Refactoring



```
int main(void) {
    double intvalue=0.0;
```

The following changes are necessary to perform the refactoring.

Changes to be performed

- ☑ Changes
  - ☑ MyCproject.c – MyCproject/src

MyCproject.c

Original Source

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

int main(void) {
    double intvalue=0.0;
    puts("!!!Hello World!!!"); /* prints !
    return EXIT_SUCCESS;
}
int foo(){
    double myint=0.0;
}
```

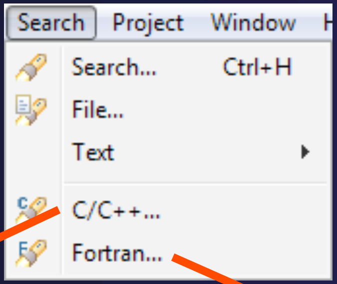
Refactored Source

```
#include <stdlib.h>

static const float MYZERO = 0.0;
int main(void) {
    double intvalue=MYZERO;
    puts("!!!Hello World!!!"); /* prin
    return EXIT_SUCCESS;
}
int foo(){
    double myint=MYZERO;
}
```

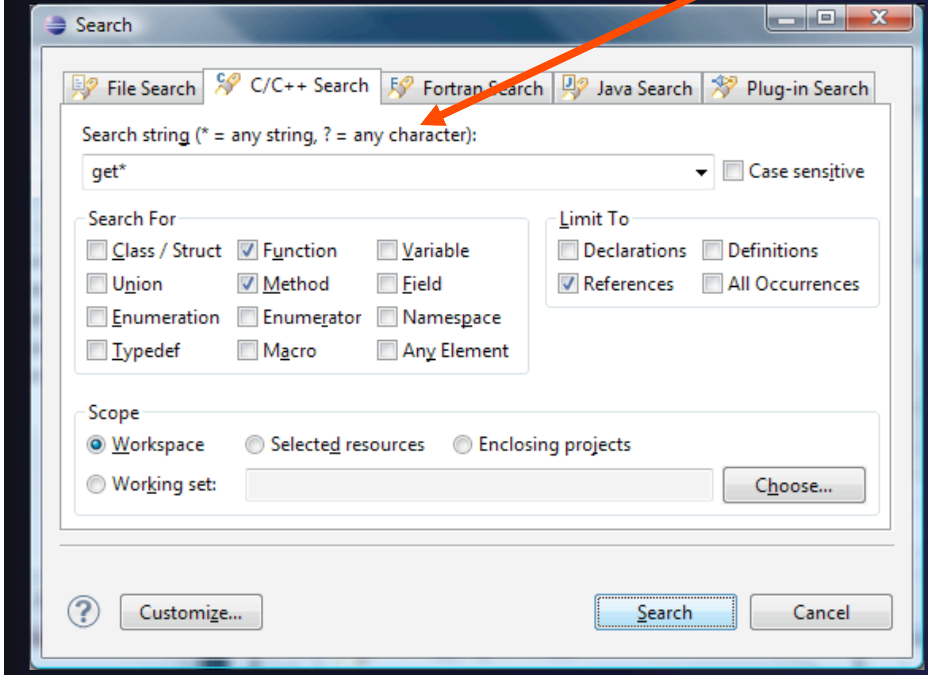
- Other refactorings that are planned:
  - Extract Function
  - Hide Member Function
  - Move Field or Member Function
  - Extract Subclass
  - Extract Baseclass
  - Separate Class
  - Implement Function
  - Declare Function
  - Move Function Definition
  - Generate Getters and Setters

Language-Based Searching

- "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



Search string (* = any string, ? = any character): get*

□ Case sensitive

Search For

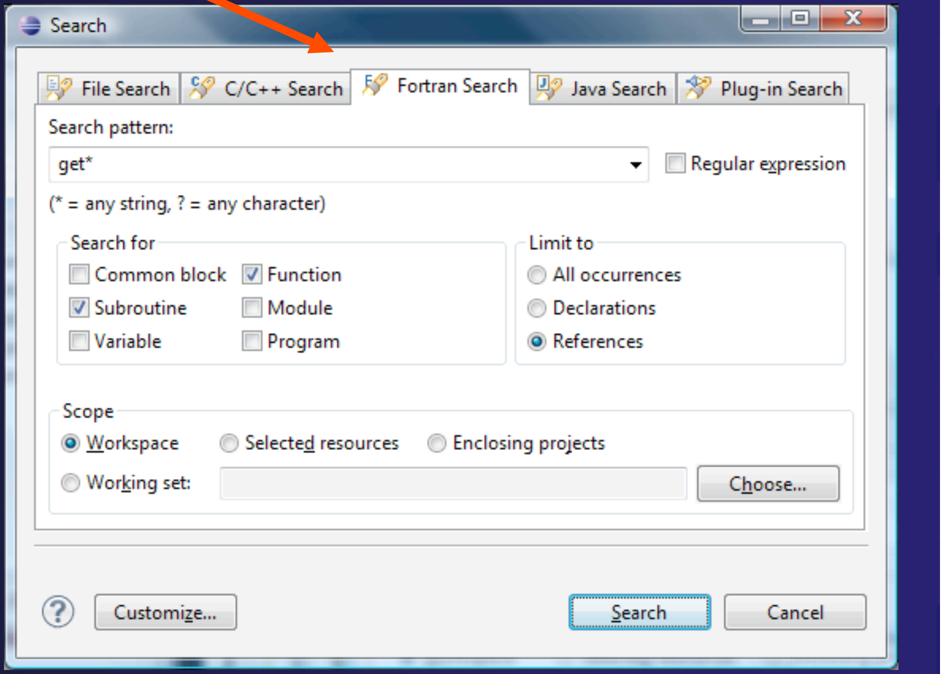
☐ Class / Struct ☑ Function ☐ Variable
☐ Union ☑ Method ☐ Field
☐ Enumeration ☐ Enumerator ☐ Namespace
☐ Typedef ☐ Macro ☐ Any Element

Limit To

☐ Declarations ☐ Definitions
☑ References ☐ All Occurrences

Scope

● Workspace ○ Selected resources ○ Enclosing projects
○ Working set:



Search pattern: get*

☐ Regular expression

(* = any string, ? = any character)

Search for

☐ Common block ☑ Function
☑ Subroutine ☐ Module
☐ Variable ☐ Program

Limit to

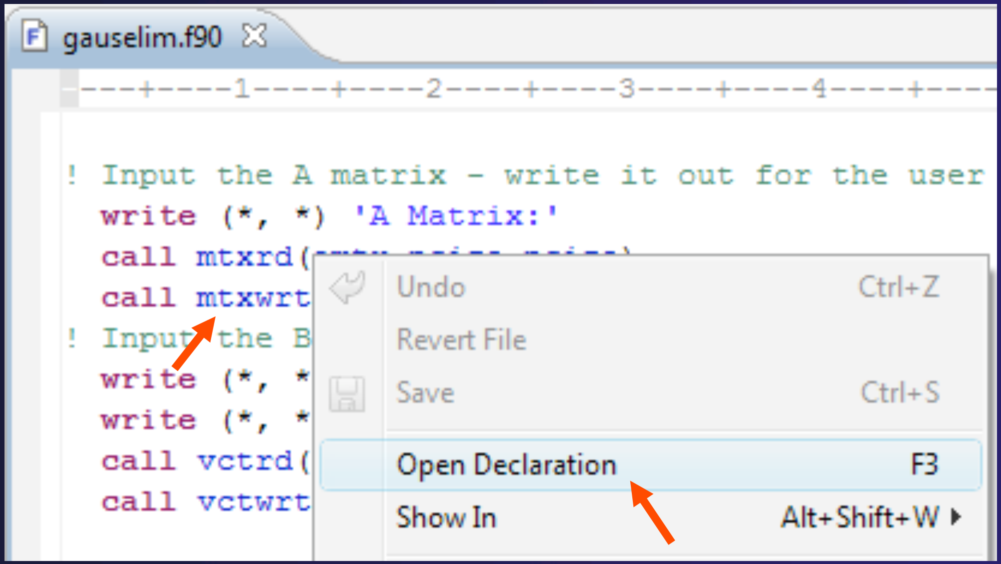
○ All occurrences
○ Declarations
● References

Scope

● Workspace ○ Selected resources ○ Enclosing projects
○ Working set:

Open Declaration

- Jumps to the declaration of a variable, function, etc., even if it's in a different file
- Right-click on an identifier
- Click **Open Declaration**



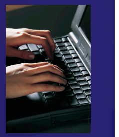
```
! Input the A matrix - write it out for the user
  write (*, *) 'A Matrix:'
  call mtxrd(
  call mtxwrt
! Input the B
  write (*, *
  write (*, *
  call vctrd(
  call vctwrt
```

***Tip:** Open Declaration works in C/C++, and it works in Fortran, but it cannot jump "across languages"*

Parallel Lang. Dev. Tools

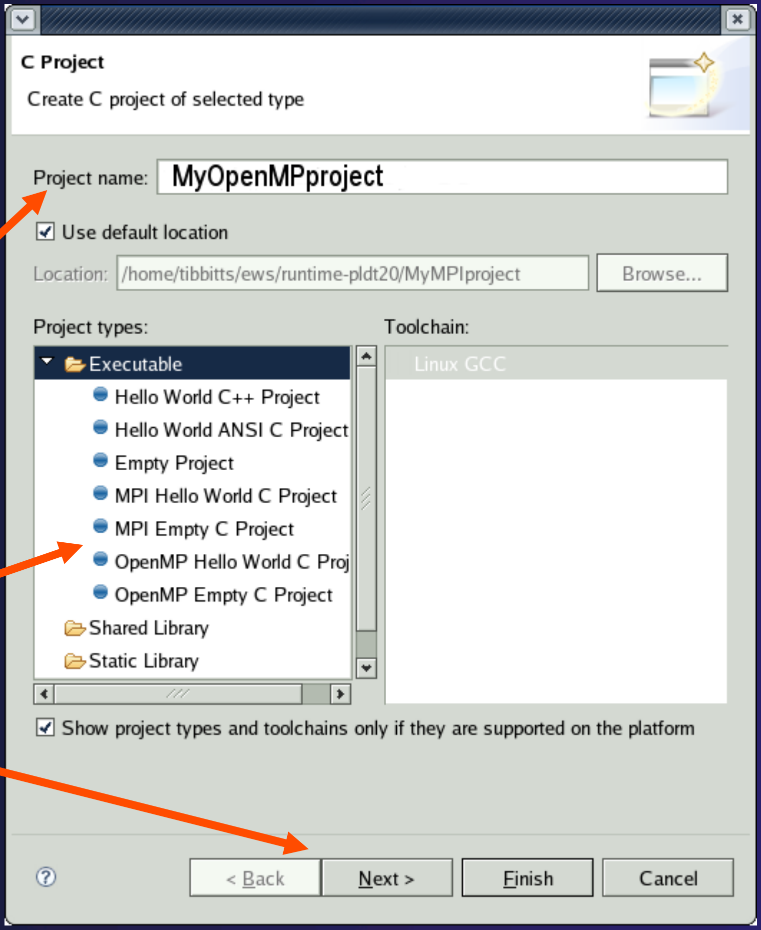
- PLDT Features
  - Analysis of C and C++ code to determine the location of MPI, OpenMP, and UPC Artifacts
  - Content assist via **ctrl+space** ("completion")
  - Hover help
  - Reference information about the API calls via Dynamic Help
  - New project wizard automatically configures managed build projects for MPI & OpenMP
  - OpenMP problems view of common errors
  - OpenMP "show #pragma region" , "show concurrency"
  - MPI Barrier analysis - detects potential deadlocks

Some MPI features were covered in Module 4



OpenMP Managed Build Project

- If you haven’t set up OpenMP preferences e.g. include file location, do it now
- Create a new OpenMP project
  - **File▶New▶C Project**
  - Name the project e.g. ‘MyOpenMPproject’
  - Select **OpenMP Hello World C Project**
  - Select **Next,** then fill in other info like MPI project



C Project

Create C project of selected type

Project name: MyOpenMPproject

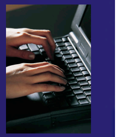
☑ Use default location

Location: /home/tibbitts/ews/runtime-pldt20/MyMPIproject

Project types:
- Executable
  - Hello World C++ Project
  - Hello World ANSI C Project
  - Empty Project
  - MPI Hello World C Project
  - MPI Empty C Project
  - OpenMP Hello World C Proj
  - OpenMP Empty C Project
- Shared Library
- Static Library

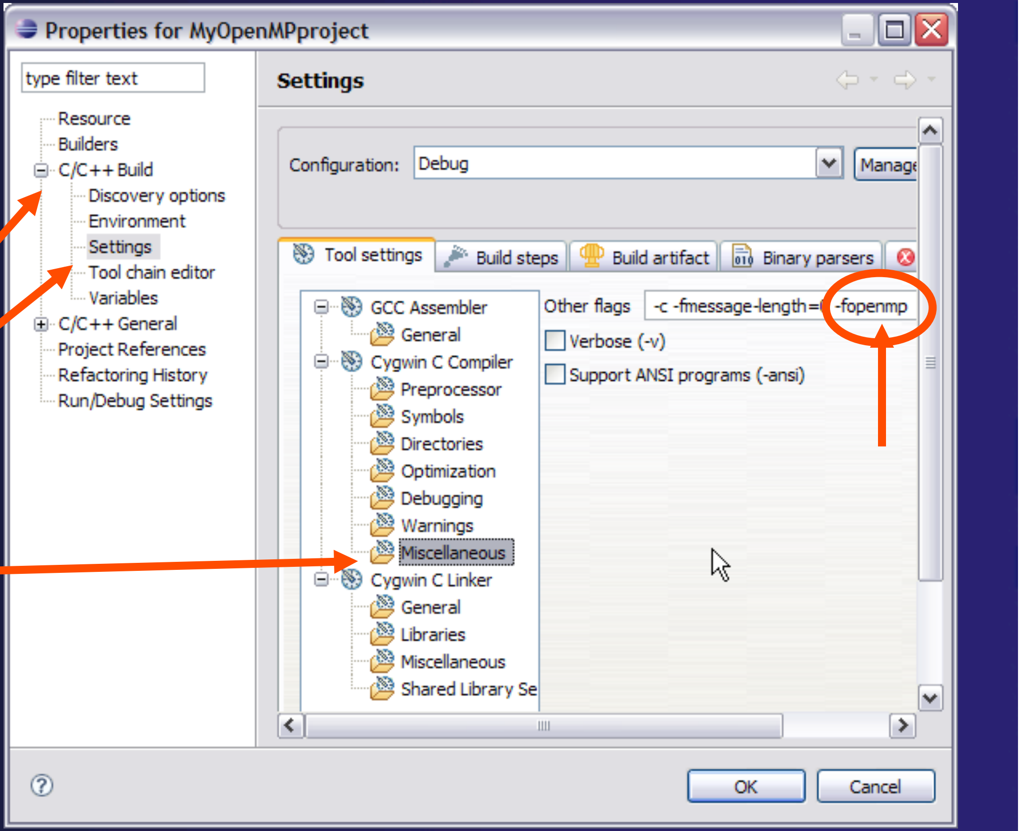
Toolchain:
- Linux GCC

☑ Show project types and toolchains only if they are supported on the platform



Setting OpenMP Special Build Options

- OpenMP typically requires special compiler options.
  - Open the project properties
  - Select **C/C++ Build**
  - Select **Settings**
  - Select **C Compiler**
    - In Miscellaneous, add option(s).

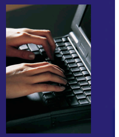


Configuration: Debug

Other flags: -c -fmessage-length=0 -fopenmp

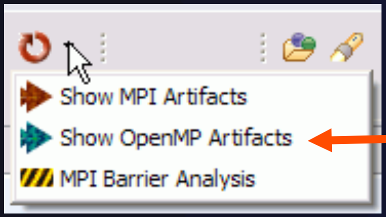
□ Verbose (-v)

□ Support ANSI programs (-ansi)

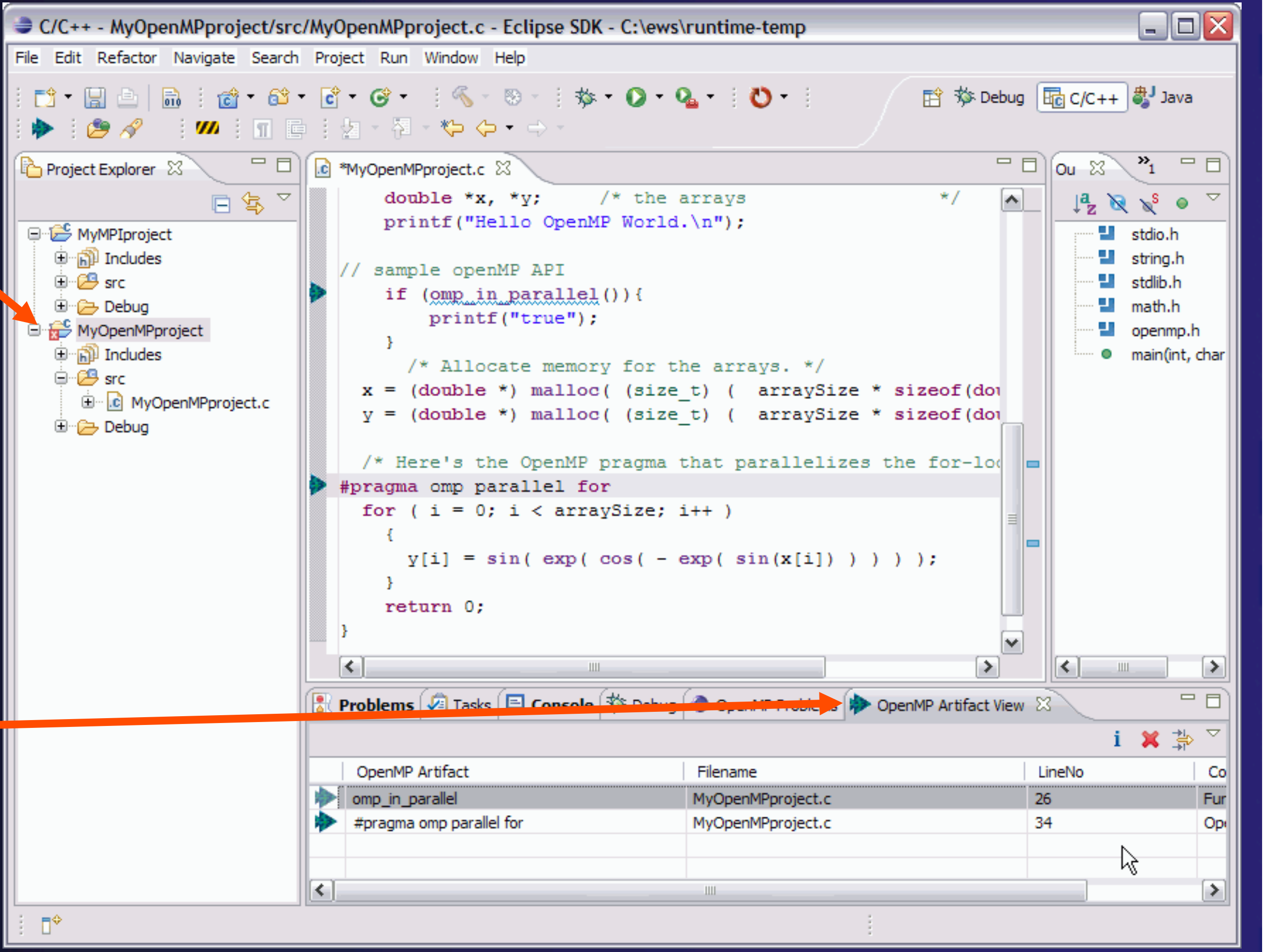


Show OpenMP Artifacts

- Select source file, folder, or project
- Run analysis



- See artifacts in **OpenMP Artifact view**

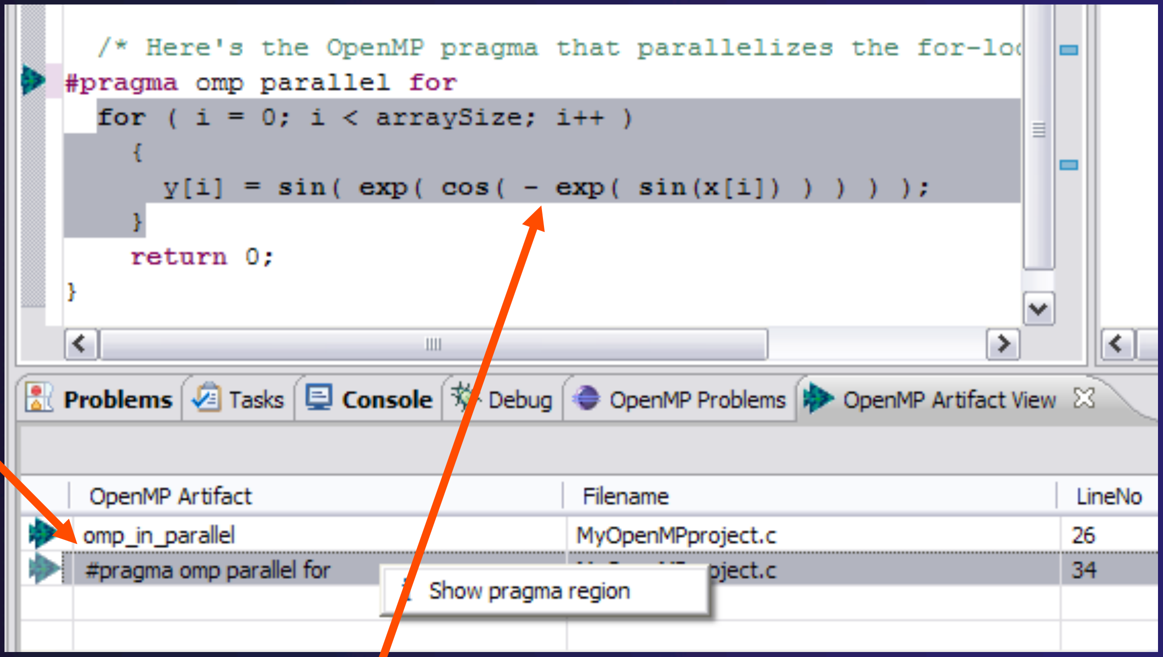


```
    double *x, *y;      /* the arrays                          */
    printf("Hello OpenMP World.\n");

// sample openMP API
    if (omp_in_parallel()){
        printf("true");
    }
      /* Allocate memory for the arrays. */
  x = (double *) malloc( (size_t) (  arraySize * sizeof(dou
  y = (double *) malloc( (size_t) (  arraySize * sizeof(dou

  /* Here's the OpenMP pragma that parallelizes the for-loo
#pragma omp parallel for
  for ( i = 0; i < arraySize; i++ )
    {
      y[i] = sin( exp( cos( - exp( sin(x[i]) ) ) ) );
    }
    return 0;
}
```

OpenMP Artifact	Filename	LineNo	Co
omp_in_parallel	MyOpenMPproject.c	26	Fur
#pragma omp parallel for	MyOpenMPproject.c	34	Op

Show Pragma Region

- Run OpenMP analysis
- Right click on pragma in artifact view
- Select **Show pragma region**
- See highlighted region in C editor

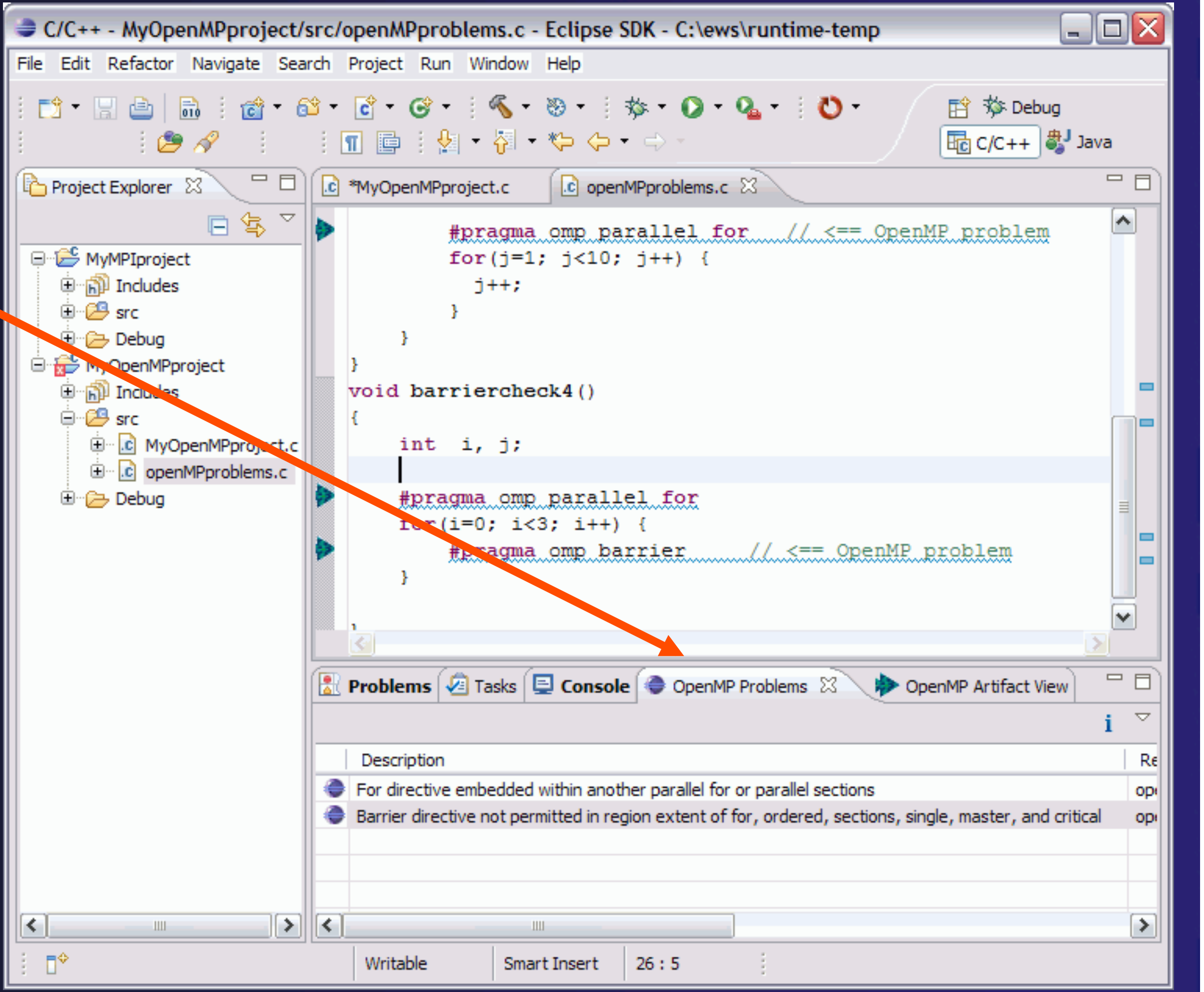


```
  /* Here's the OpenMP pragma that parallelizes the for-loo
#pragma omp parallel for
  for ( i = 0; i < arraySize; i++ )
    {
      y[i] = sin( exp( cos( - exp( sin(x[i]) ) ) ) );
    }
    return 0;
}
```

OpenMP Artifact	Filename	LineNo
omp_in_parallel	MyOpenMPproject.c	26
#pragma omp parallel for	oject.c	34

Show OpenMP Problems

- Select **OpenMP problems view**
- Will identify standard OpenMP restrictions

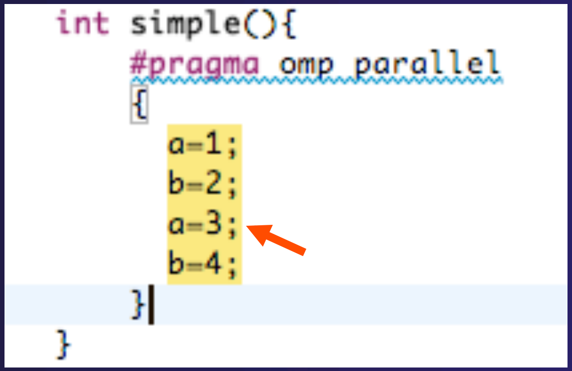


```
        #pragma omp parallel for    // <== OpenMP problem
        for(j=1; j<10; j++) {
          j++;
        }
    }
}
void barriercheck4()
{
    int  i, j;

    #pragma omp parallel for
    for(i=0; i<3; i++) {
        #pragma omp barrier      // <== OpenMP problem
    }
```

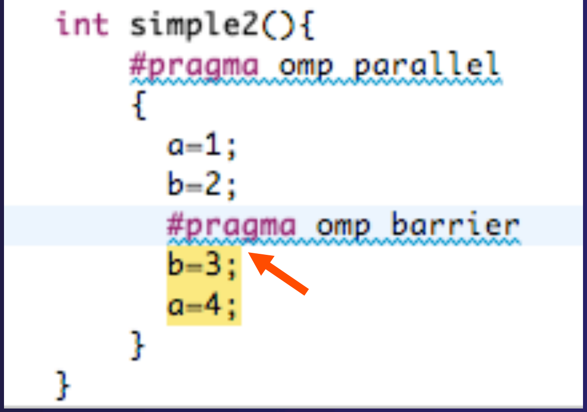
Description	Re
For directive embedded within another parallel for or parallel sections	op
Barrier directive not permitted in region extent of for, ordered, sections, single, master, and critical	op

Show Concurrency

- Highlight a statement
- Select the context menu on the highlighted statement, and click **Show concurrency**
- Other statements will be highlighted in yellow
- The yellow highlighted statements *might* execute concurrently to the selected statement



```
int simple(){
    #pragma omp parallel
    {
        a=1;
        b=2;
        a=3;
        b=4;
    }
}
```

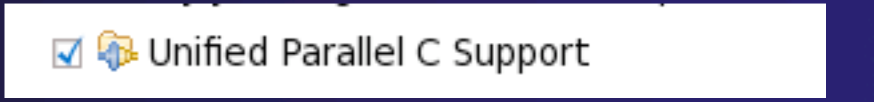


```
int simple2(){
    #pragma omp parallel
    {
        a=1;
        b=2;
        #pragma omp barrier
        b=3;
        a=4;
    }
}
```

UPC Support

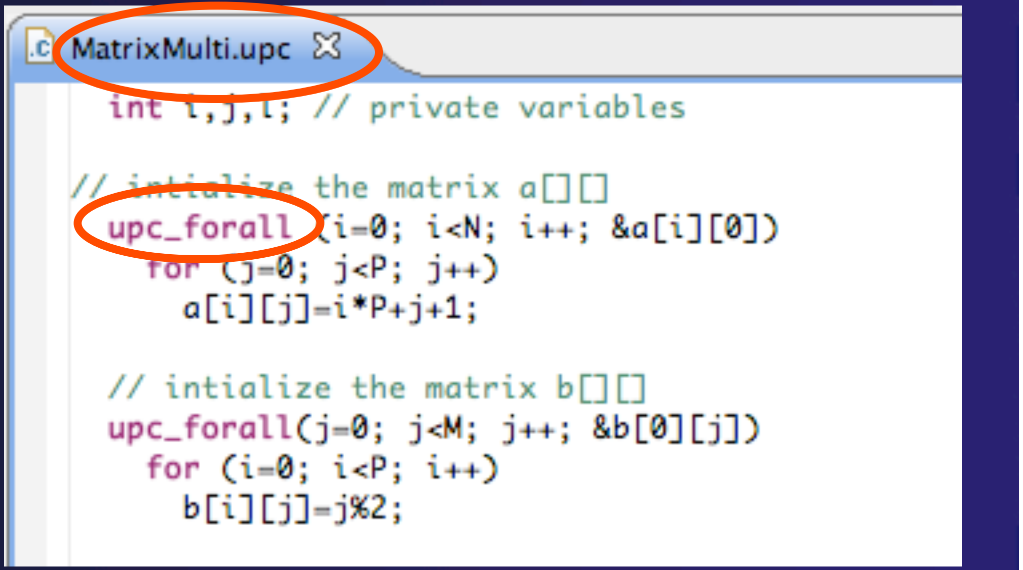
- To see UPC support in C editor, install the optional feature from CDT
- See Also: http://wiki.eclipse.org/PTP/other_tools_setup#Using_UPC_features

Under Optional Features



☑ Unified Parallel C Support

- Filetypes of "upc" will get UPC syntax highlighting, content assist, etc.
- Use preferences to change default for *.c if you like



```
int i,j,l; // private variables

// intialize the matrix a[][]
upc_forall(i=0; i<N; i++; &a[i][0])
  for (j=0; j<P; j++)
    a[i][j]=i*P+j+1;

// intialize the matrix b[][]
upc_forall(j=0; j<M; j++; &b[0][j])
  for (i=0; i<P; i++)
    b[i][j]=j%2;
```

Remote Development

- PTP already provides the ability to launch/debug remotely
  - However it is often desirable to be able to edit and build remotely
  - If projects are very large, build times may be considerable
- The PTP Remote Development Tools (RDT) will provide a complete remote development environment
  - C/C++ (and Fortran) projects can be hosted on a remote machine
  - Eclipse runs on your local workstation or laptop
  - Files are pulled to local machine only for editing
  - Remote indexing and other services are performed remotely
  - Both managed and Makefile projects are built remotely
  - Uses either Remote System Explorer (RSE) or PTP’s Remote Tools
  - Will have the ability to tunnel over ssh connections

Remote Development (2)

- RDT was introduced with PTP 2.1
  - Configuration is separate from PTP configuration
  - Restricted to RSE connections only (no tunneling)
  - Manual server launch
- PTP 3.0 will seamlessly integrate RDT configuration and simplify setup and use
  - New service model combines PTP and RDT configuration
  - New project wizard has been enhanced and simplified
  - Automatically launch remote server process
  - Still under active development

…. So we won’t cover it today

Module 8: Other Tools and Wrap-up

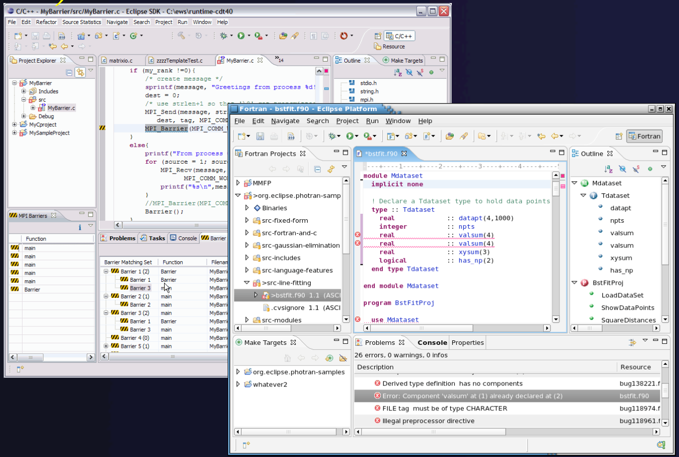
- Objective
  - How to find more information on PTP
  - Learn about other tools related to PTP
  - See PTP upcoming features
- Contents
  - Links to other tools, including performance tools
  - Planned features for new versions of PTP
  - Additional documentation
  - How to get involved

NCSA HPC Workbench

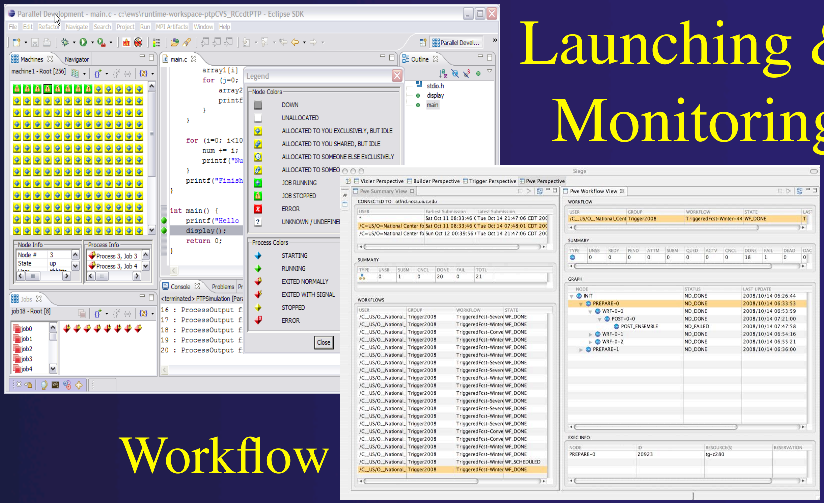
- Tools for NCSA Blue Waters
  - http://www.ncsa.illinois.edu/BlueWaters/
  - Sustained Petaflop system
- Based on Eclipse and PTP
- Includes some related tools
  - Performance tools
  - Scalable debugger
  - Workflow tools (https://wiki.ncsa.uiuc.edu/display/MRDPUB/MRD+Public+Space+Home+Page)
- Part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

NCSA HPC Workbench

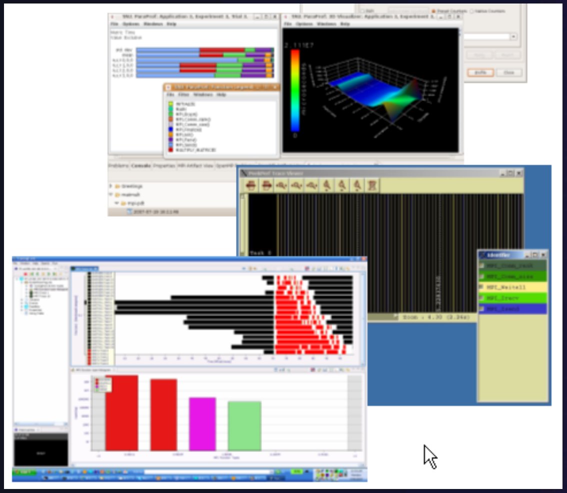
Coding & Analysis (CDT, PLDT, Photran)



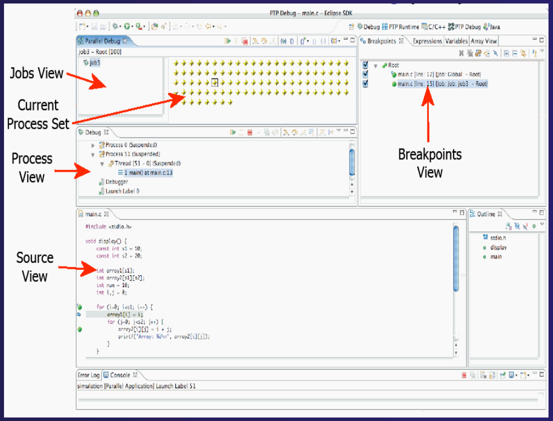
PTP Launching & Monitoring



Workflow



Performance Tuning (HPC toolkit, HPCS toolkit, RENCI, …)



PTP Debugging

PTP-Related Tools

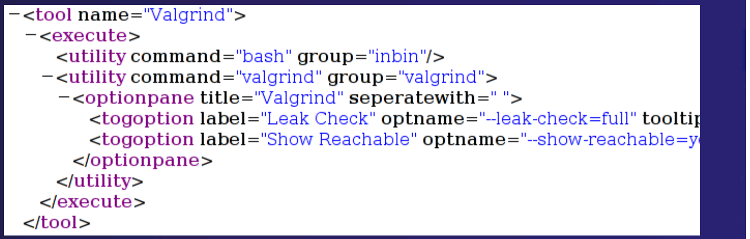
- External Tools Framework
  - Formerly Performance Tools Framework
- Tuning and Analysis Utilities (TAU)
- ISP – In-situ Partial Ordering
  - MPI analysis tools from U.Utah

PTP/External Tools Framework

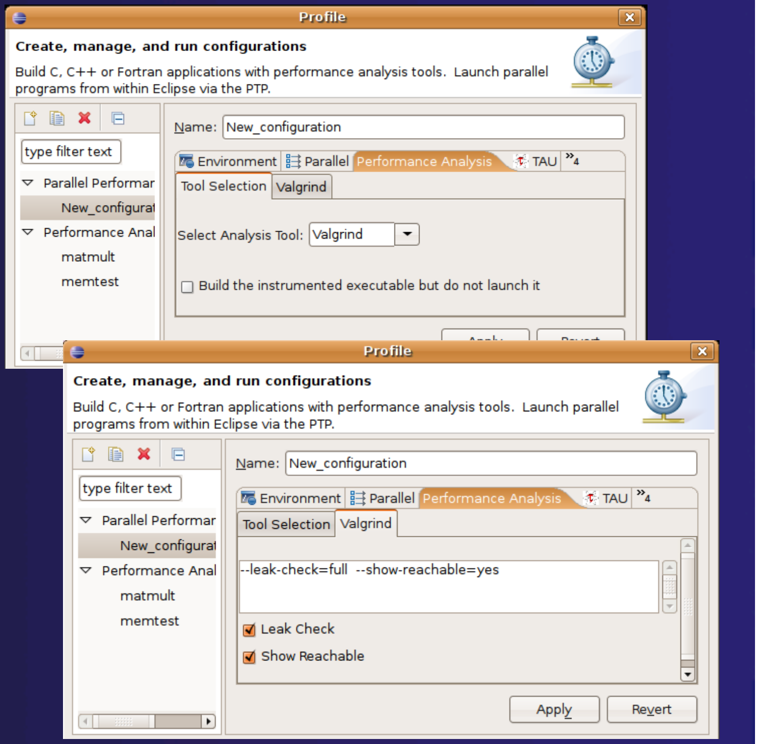
formerly “Performance Tools Framework”

Goal:

- **Reduce the “eclipse plumbing” necessary to integrate tools**
- Provide integration for instrumentation, measurement, and analysis for a variety of performance tools
  - Dynamic Tool Definitions: Workflows & UI
  - Tools and tool workflows are specified in an XML file
  - Tools are selected and configured in the launch configuration window
  - Output is generated, managed and analyzed as specified in the workflow



```
−<tool name="Valgrind">
  −<execute>
     <utility command="bash" group="inbin"/>
    −<utility command="valgrind" group="valgrind">
      −<optionpane title="Valgrind" seperatewith=" ">
          <togoption label="Leak Check" optname="--leak-check=full" tooltip
          <togoption label="Show Reachable" optname="--show-reachable=y
        </optionpane>
     </utility>
   </execute>
 </tool>
```



Profile

Create, manage, and run configurations

Build C, C++ or Fortran applications with performance analysis tools. Launch parallel programs from within Eclipse via the PTP.

type filter text

- Parallel Performar
  - New_configurat
- Performance Anal
  - matmult
  - memtest

Name: New_configuration

Environment | Parallel | Performance Analysis | TAU | »4

Tool Selection | Valgrind

Select Analysis Tool: Valgrind

□ Build the instrumented executable but do not launch it

Profile

Create, manage, and run configurations

Build C, C++ or Fortran applications with performance analysis tools. Launch parallel programs from within Eclipse via the PTP.

type filter text

- Parallel Performar
  - New_configurat
- Performance Anal
  - matmult
  - memtest

Name: New_configuration

Environment | Parallel | Performance Analysis | TAU | »4

Tool Selection | Valgrind

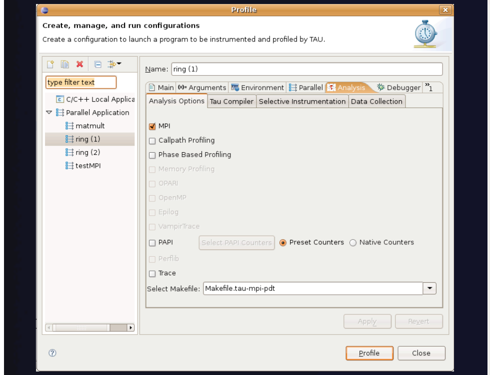
--leak-check=full --show-reachable=yes

☑ Leak Check

☑ Show Reachable

PTP TAU plug-ins http://www.cs.uoregon.edu/research/tau/home.php

- TAU (Tuning and Analysis Utilities)
- First implementation of Performance Tools Framework
- Eclipse plug-ins wrap TAU functions, make them available from Eclipse
- Compatible with Photran and CDT projects and with PTP parallel application launching
- Other plug-ins launch Paraprof from Eclipse too



Profile

Create, manage, and run configurations

Create a configuration to launch a program to be instrumented and profiled by TAU.

type filter text

- C/C++ Local Applica
- Parallel Application
  - matmult
  - ring (1)
  - ring (2)
  - testMPI

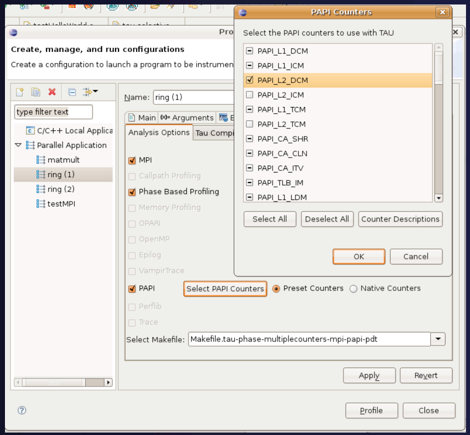
Name: ring (1)

Main | Arguments | Environment | Parallel | Analysis | Debugger

Analysis Options | Tau Compiler | Selective Instrumentation | Data Collection

☑ MPI
□ Callpath Profiling
□ Phase Based Profiling
□ Memory Profiling
□ OPARI
□ OpenMP
□ Epilog
□ VampirTrace
□ PAPI Select PAPI Counters ● Preset Counters ○ Native Counters
□ Perflib
□ Trace

Select Makefile: Makefile.tau-mpi-pdt



PAPI Counters

Select the PAPI counters to use with TAU

⊟ PAPI_L1_DCM
⊟ PAPI_L1_ICM
☑ PAPI_L2_DCM
□ PAPI_L2_ICM
⊟ PAPI_L1_TCM
□ PAPI_L2_TCM
⊟ PAPI_CA_SHR
⊟ PAPI_CA_CLN
⊟ PAPI_CA_ITV
⊟ PAPI_TLB_IM
⊟ PAPI_L1_LDM

Create, manage, and run configurations

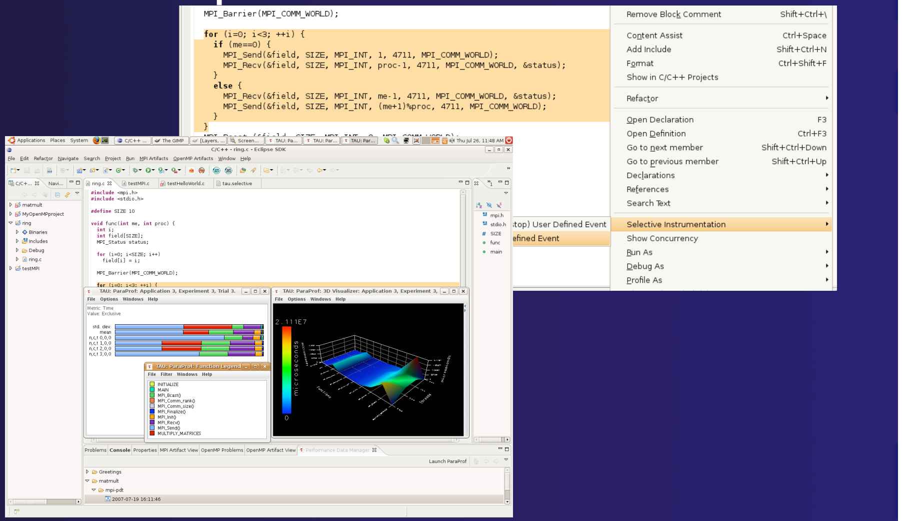
Create a configuration to launch a program to be instrumen

Name: ring (1)

☑ MPI
□ Callpath Profiling
☑ Phase Based Profiling
□ Memory Profiling
□ OPARI
□ OpenMP
□ Epilog
□ VampirTrace
☑ PAPI Select PAPI Counters ● Preset Counters ○ Native Counters
□ Perflib
□ Trace

Select Makefile: Makefile.tau-phase-multiplecounters-mpi-papi-pdt

Apply
Revert
Profile
Close



```
MPI_Barrier(MPI_COMM_WORLD);

for (i=0; i<3; ++i) {
  if (me==0) {
    MPI_Send(&field, SIZE, MPI_INT, 1, 4711, MPI_COMM_WORLD);
    MPI_Recv(&field, SIZE, MPI_INT, proc-1, 4711, MPI_COMM_WORLD, &status);
  }
  else {
    MPI_Recv(&field, SIZE, MPI_INT, me-1, 4711, MPI_COMM_WORLD, &status);
    MPI_Send(&field, SIZE, MPI_INT, (me+1)%proc, 4711, MPI_COMM_WORLD);
  }
}
```

Remove Block Comment Shift+Ctrl+\
Content Assist Ctrl+Space
Add Include Shift+Ctrl+N
Format Ctrl+Shift+F
Show in C/C++ Projects
Refactor
Open Declaration F3
Open Definition Ctrl+F3
Go to next member Shift+Ctrl+Down
Go to previous member Shift+Ctrl+Up
Declarations
References
Search Text
Selective Instrumentation
Show Concurrency
Run As
Debug As
Profile As
top) User Defined Event
efined Event
C/C++ - ring.c - Eclipse SDK
TAU: ParaProf: 3D Visualizer: Application 3, Experiment 3,
TAU: ParaProf: Function Legend
2.111E7
seconds

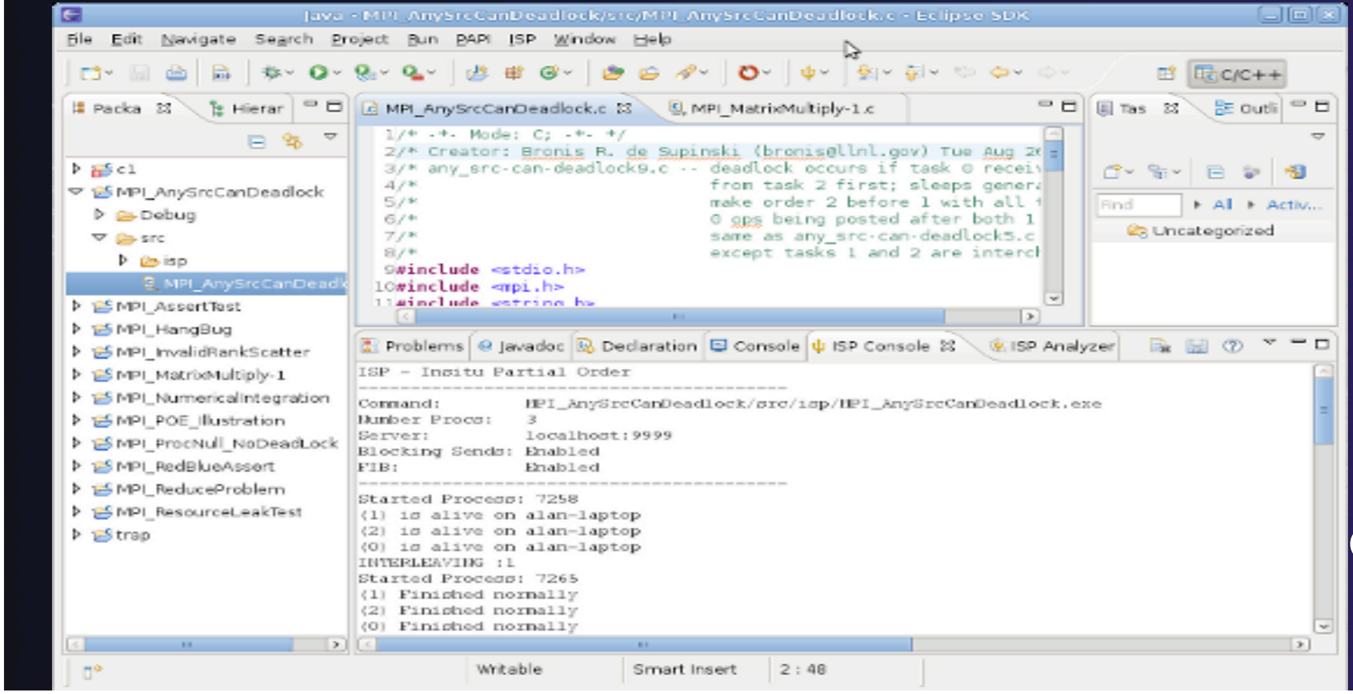
ISP – In-situ Partial Order

- Being contributed to PTP by U. Utah
  - Hope to make available in PTP 3.0 (late Oct.)
- Analyses MPI code dynamically for deadlocks, etc.
- Can match sends and recieves
- Can work with several different MPI implementations

ISP – Formal Dynamic Verification of MPI Applications

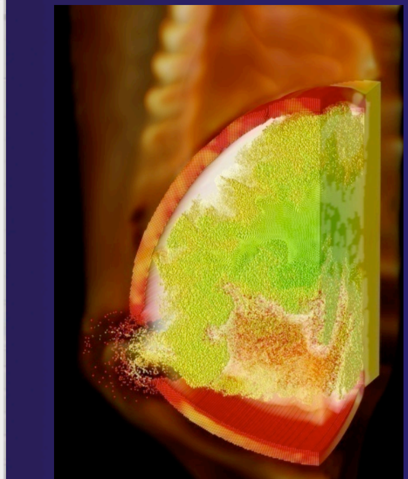
(BlueGene/L - Image courtesy of IBM / LLNL)

- Verifies MPI User Applications, generating only the *Relevant Process Interleavings*
- Detects all Deadlocks, Assert Violations, MPI object leaks, and Default Safety Properties
- Works by Instrumenting MPI Calls Computing Relevant Interleavings, Replaying



```
/* -*- Mode: C; -*- */
/* Creator: Bronis R. de Supinski (bronis@llnl.gov) Tue Aug 2
/* any_src-can-deadlock9.c -- deadlock occurs if task 0 recei
/*                            from task 2 first; sleeps genera
/*                            make order 2 before 1 with all 
/*                            0 ops being posted after both 1
/*                            same as any_src-can-deadlock5.c
/*                            except tasks 1 and 2 are interch
#include <stdio.h>
#include <mpi.h>
#include <string.h>
```

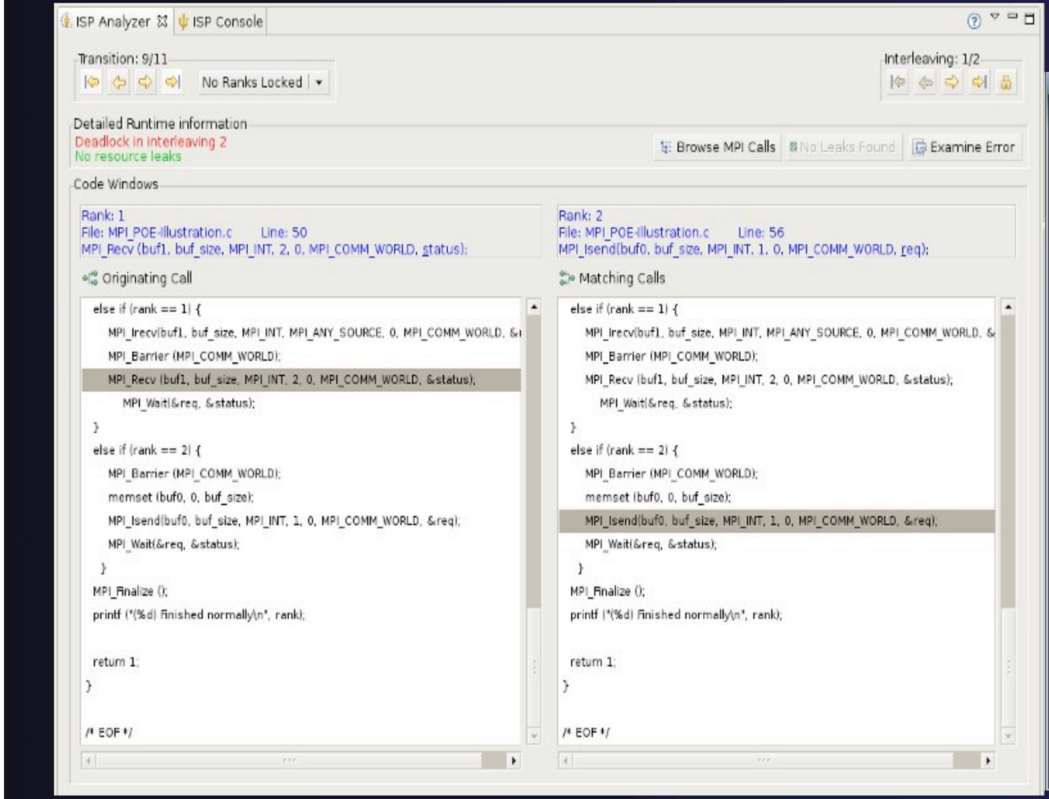
```
ISP - Insitu Partial Order
-----------------------------------------
Command:          MPI_AnySrcCanDeadlock/src/isp/MPI_AnySrcCanDeadlock.exe
Number Procs:     3
Server:           localhost:9999
Blocking Sends:   Enabled
FIB:              Enabled
-----------------------------------------
Started Process: 7258
(1) is alive on alan-laptop
(2) is alive on alan-laptop
(0) is alive on alan-laptop
INTERLEAVING :1
Started Process: 7265
(1) Finished normally
(2) Finished normally
(0) Finished normally
```



(Image courtesy of Steve Parker, U of Utah)

Eclipse CDT/PTP based ISP GUI

ISP Plug-in (trident icon) based on CDT and PTP allows PostVerification Review of Relevant Interleavings, and highlights bugs



Transition: 9/11

Interleaving: 1/2

Detailed Runtime information

Deadlock in interleaving 2

No resource leaks

Code Windows

Rank: 1
File: MPI_POE-Illustration.c Line: 50
MPI_Recv (buf1, buf_size, MPI_INT, 2, 0, MPI_COMM_WORLD, status);

Originating Call

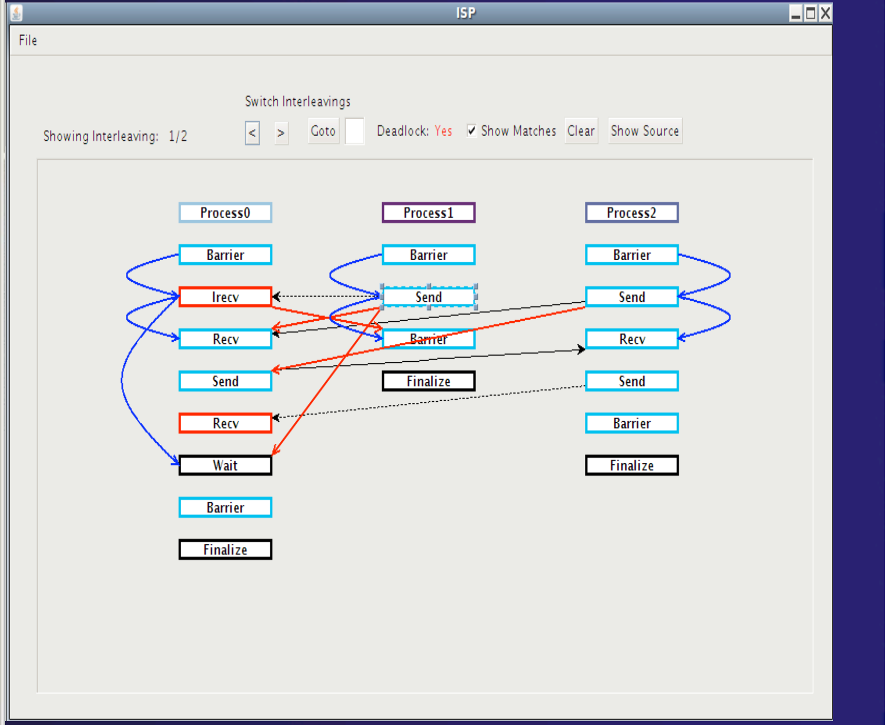
```
else if (rank == 1) {
    MPI_Irecv(buf1, buf_size, MPI_INT, MPI_ANY_SOURCE, 0, MPI_COMM_WORLD, &r
    MPI_Barrier (MPI_COMM_WORLD);
    MPI_Recv (buf1, buf_size, MPI_INT, 2, 0, MPI_COMM_WORLD, &status);
      MPI_Wait(&req, &status);
}
else if (rank == 2) {
    MPI_Barrier (MPI_COMM_WORLD);
    memset (buf0, 0, buf_size);
    MPI_Isend(buf0, buf_size, MPI_INT, 1, 0, MPI_COMM_WORLD, &req);
    MPI_Wait(&req, &status);
  }
MPI_Finalize ();
printf ("(%d) Finished normally\n", rank);

return 1;
}

/* EOF */
```

Rank: 2
File: MPI_POE-Illustration.c Line: 56
MPI_Isend(buf0, buf_size, MPI_INT, 1, 0, MPI_COMM_WORLD, req);

Matching Calls

```
else if (rank == 1) {
    MPI_Irecv(buf1, buf_size, MPI_INT, MPI_ANY_SOURCE, 0, MPI_COMM_WORLD, &
    MPI_Barrier (MPI_COMM_WORLD);
    MPI_Recv (buf1, buf_size, MPI_INT, 2, 0, MPI_COMM_WORLD, &status);
      MPI_Wait(&req, &status);
}
else if (rank == 2) {
    MPI_Barrier (MPI_COMM_WORLD);
    memset (buf0, 0, buf_size);
    MPI_Isend(buf0, buf_size, MPI_INT, 1, 0, MPI_COMM_WORLD, &req);
    MPI_Wait(&req, &status);
  }
MPI_Finalize ();
printf ("(%d) Finished normally\n", rank);

return 1;
}

/* EOF */
```

It also allows viewing of MPI Happens-Before Relation – a succinct summary of the required MPI orderings



Showing Interleaving: 1/2

Deadlock: Yes

For details, including Beta download, please visit
http://www.cs.utah.edu/formal_verification/ISP-Eclipse

Useful Eclipse Tools

- Python
  - http://pydev.sourceforge.net
- Ruby
  - http://sourceforge.net/projects/rubyeclipse
- Subversion (now an Eclipse project)
  - http://eclipse.org/subversive
- Git (now an Eclipse project)
  - http://www.eclipse.org/egit
- … and many more!

Future PTP Features

- Support for multicore development
  - Building on Cell IDE and other multicore tools
- Resource managers to support for PBS, LSF, and Blue Gene
- Transitioning debugger to Scalable Tools Communication Infrastructure (STCI)
- Enhancements to ETF to support compiler generated reports and optimization directives
- Scalability improvements
  - UI to support 1M processes
  - Optimized communication protocol
  - Very large application support

Information About PTP

- Main web site for downloads, documentation, etc.
  - http://eclipse.org/ptp
- Developers' wiki for designs, planning, meetings, etc.
  - http://wiki.eclipse.org/PTP
- Articles and other documents:
  - http://wiki.eclipse.org/PTP/articles

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 –
    - http://dev.eclipse.org/mailman/listinfo/photran-dev

Getting Involved

- See http://eclipse.org/ptp
- Read the developer documentation on the wiki
- Join the mailing lists
- Attend the monthly developer meetings
  - Teleconference each second Tuesday, 1:00 pm ET

- PTP will only succeed with your participation!

PTP Tutorial Feedback

- Please complete feedback form
- Your feedback is valuable!

Thanks for attending
We hope you found it useful