

Compile and Debug SWAT with GFortran and Eclipse

(Windows version)

Zhiqiang Yu, hawklorry@gmail.com

February, 2014

[Introduction](#)

[Downloading and Installing Eclipse Parallel Tools Platform \(PTP\)](#)

[Downloading and Installing GFortran and MSYS](#)

[Rtools](#)

[MinGW-w64](#)

[MinGW](#)

[Downloading SWAT Source Codes](#)

[Get Makefile](#)

[Downloading Makefile](#)

[Generating Makefile with Makefile Generator](#)

[Compiling SWAT in Command Line](#)

[Compiling and Debugging SWAT in Eclipse](#)

[Create a Fortran Project](#)

[Find Fortran Project... Menu](#)

[Create an Empty Makefile Project](#)

[Setup the Project](#)

[Build Command](#)

[Fortran Build Environment](#)

[Binary Parser](#)

[Error Parsers](#)

[Copy SWAT Source Codes and Makefile](#)

[Create Make Target](#)

[Build Make Target](#)

[Debug/Run SWAT](#)

[Create Run/Debug Settings](#)

[Debug/Run](#)

[Break Point](#)

[Debug Perspective](#)

[Variables and Expression View](#)

[Some Useful Settings](#)

[Change Text Font](#)

[Show Line Number](#)

[Save automatically before build](#)

[Change Keys](#)

Introduction

SWAT is initially developed and compiled with Intel Fortran, which is not free. To allow more users participate in SWAT development, it's important that SWAT could be compiled and debug in a totally free environment. GFortran plus Eclipse is the toolset perfect for this job.

This document would show you how to install and configure GFortran and Eclipse Parallel Tools Platfrom (PTP) and then compile and debug SWAT with them.

Downloading and Installing Eclipse Parallel Tools Platform (PTP)

PTP is the IDE where SWAT codes will be compiled and debugged. For more features, please visit Photran information page <http://www.eclipse.org/photran/>.

1. Go to <http://eclipse.org/downloads/> and select downloads for Eclipse for Parallel Application Developers based on Window version (32 bit or 64 bit).

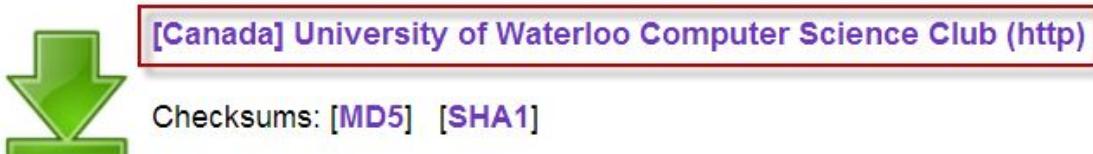
 Eclipse Modeling Tools , 292 MB Downloaded 177,665 Times This package contains framework and tools to leverage models : an Ecore graphical modeler (class-like diagram), Java code generation utility for.	 Windows 32 Bit  Windows 64 Bit
 Eclipse for RCP and RAP Developers , 236 MB Downloaded 171,036 Times A complete set of tools for developers who want to create Eclipse plug-ins, Rich Client or Rich Ajax Applications (RCP+RAP), plus.	 Windows 32 Bit  Windows 64 Bit
 Eclipse for Parallel Application Developers , 212 MB Downloaded 163,342 Times Tools for C, C++, Fortran, and UPC, including MPI, OpenMP, OpenACC, a parallel debugger, and remotely building, running and monitoring applications...	 Windows 32 Bit  Windows 64 Bit
 Eclipse for Scout Developers , 288 MB Downloaded 156,865 Times This package contains framework and tools to develop Eclipse Scout applications.	 Windows 32 Bit  Windows 64 Bit
 Eclipse IDE for Automotive Software Developers (includes Incubating components) , 195 MB Downloaded 155,768 Times This package contains frameworks and tools used for the development of embedded automotive software: In addition to Eclipse Platform, Java Development...	 Windows 32 Bit  Windows 64 Bit

2. The download link would be give based on your location. Click the download link to start the download process. The name of download file would like **eclipse-parallel-kepler-SR1-win32-x86_64.zip**.

Eclipse downloads - mirror selection

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

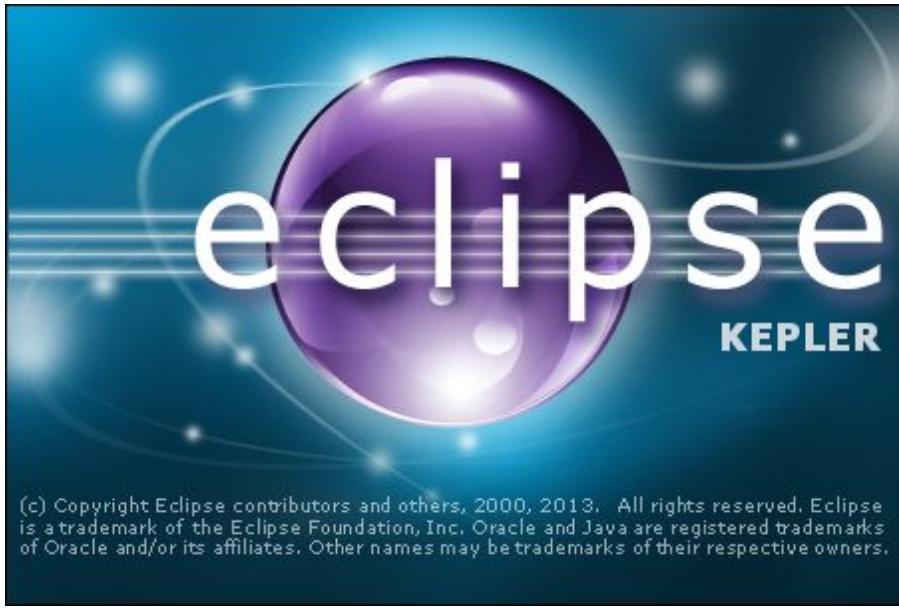
Download [eclipse-parallel-kepler-SR1-win32-x86_64.zip](#) from:



...or pick a mirror site below.

3. After the download process is done, extract the zip file to a folder. There is a file named `eclipse.exe` in the extracted folder. This is the main program of Eclipse. Create a shortcut on the desktop if necessary. Double-click this file to start Eclipse.

Name	Date modified	Type	Size
configuration	1/24/2014 10:56 AM	File folder	
dropins	9/19/2013 6:37 AM	File folder	
features	1/24/2014 10:58 AM	File folder	
p2	1/24/2014 10:58 AM	File folder	
plugins	1/24/2014 11:03 AM	File folder	
readme	1/24/2014 11:03 AM	File folder	
.eclipseproduct	1/24/2014 11:03 AM	ECLIPSEPRODUCT...	1 KB
artifacts.xml	1/24/2014 11:03 AM	XML Document	156 KB
eclipse.exe	1/24/2014 11:03 AM	Application	305 KB
eclipse.ini	1/24/2014 11:03 AM	Configuration sett...	1 KB
eclipsesec.exe	1/24/2014 11:03 AM	Application	18 KB
epl-v10.html	1/24/2014 11:03 AM	Chrome HTML Do...	17 KB
notice.html	1/24/2014 11:03 AM	Chrome HTML Do...	10 KB



(c) Copyright Eclipse contributors and others, 2000, 2013. All rights reserved. Eclipse is a trademark of the Eclipse Foundation, Inc. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Note:

Eclipse requires a Java Runtime Environment (JRE) in order to run. JRE should be already installed in most of the Windows system. In case it's not available, please download it from Oracle website

<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>.

Downloading and Installing GFortran and MSYS

[GFortran](#) is the GNU Fortran compiler, part of GCC. It's developed in Linux. To use it in Windows, there are two options: MinGW and Cygwin. As the program compiled with Cygwin is not native Windows program and requires Cygwin installed on the target machine, MinGW is the better choice. MinGW, a contraction of "Minimalist GNU for Windows", is a minimalist development environment for native Microsoft Windows application. gfortran is part of MinGW.

[MSYS](#) is a collection of GNU utilities such as bash, make, gawk and grep to allow building of applications and programs which depend on traditionally UNIX tools to be present. It is intended to supplement MinGW and the deficiencies of the cmd shell. Three of these utilities are used in SWAT Makefile: **make**, **rm** and **mkdir**, where make is used to execute the Makefile, rm is used to clear the project and mkdir is used to create the subfolder for object (*.o) files.

There are different versions (32-bit and 64-bit) of MinGW and different ways to install MinGW and MSYS. Three methods are demonstrated in this document and **we recommend to use MinGW**, which is easiest way for beginners although it doesn't support 64-bit build.

Rtools

[R](#) is a free software environment for statistical computing and graphics. Rtools is the toolset to compile R source codes, most of which are C and Fortran. It combines some files from Cygwin, MinGW-w64 and MSYS and has everything we need for SWAT compilation.

1. Visit <http://cran.r-project.org/bin/windows/Rtools/> to download the latest Rtools.

Building R for Windows

This document is a collection of resources for building packages for R under Microsoft Windows, or for building R itself (version 1.9.0 or later). The original collection was put together by Prof. Brian Ripley; it is currently being maintained by Duncan Murdoch.

The authoritative source of information for tools to work with the current release of R is the "R Administration and Installation" manual. In particular, please read the ["Windows Toolset" appendix](#).

Rtools Downloads

With the change to gcc 4.2.1, some of the tools for 32 bit compiles became incompatible with obsolete versions of R. Since then we have been maintaining one actively updated version of the tools, and other "frozen" snapshots of them. We recommend that users use the latest release of Rtools with the latest release of R.

The current version of this file is recorded here: [VERSION.txt](#)

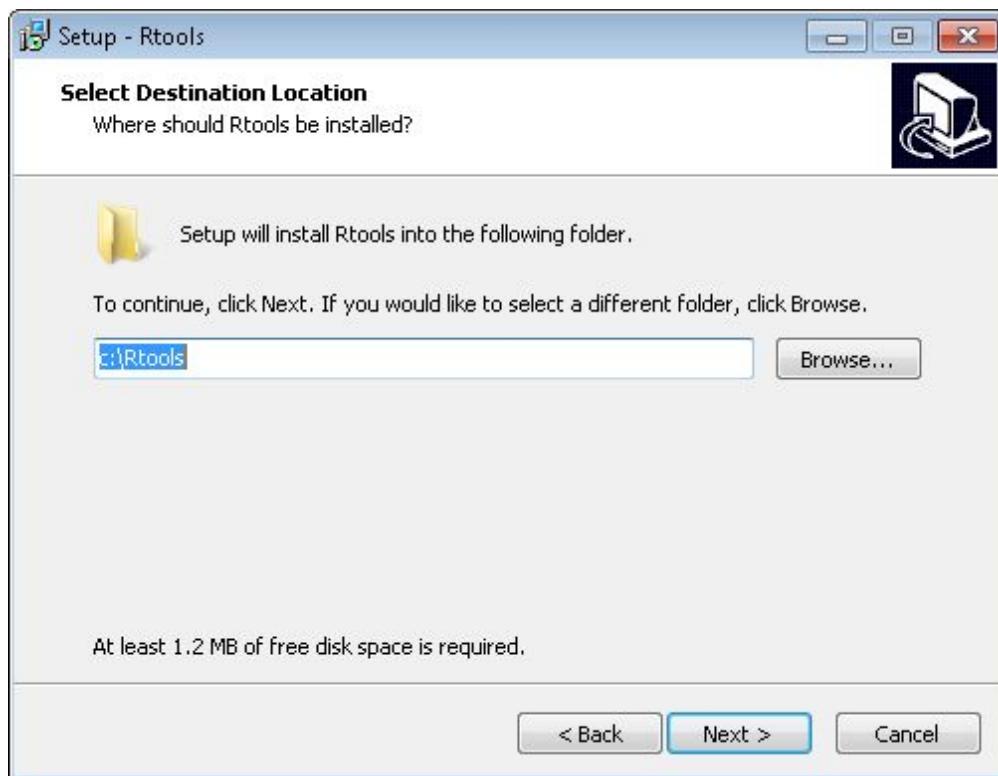
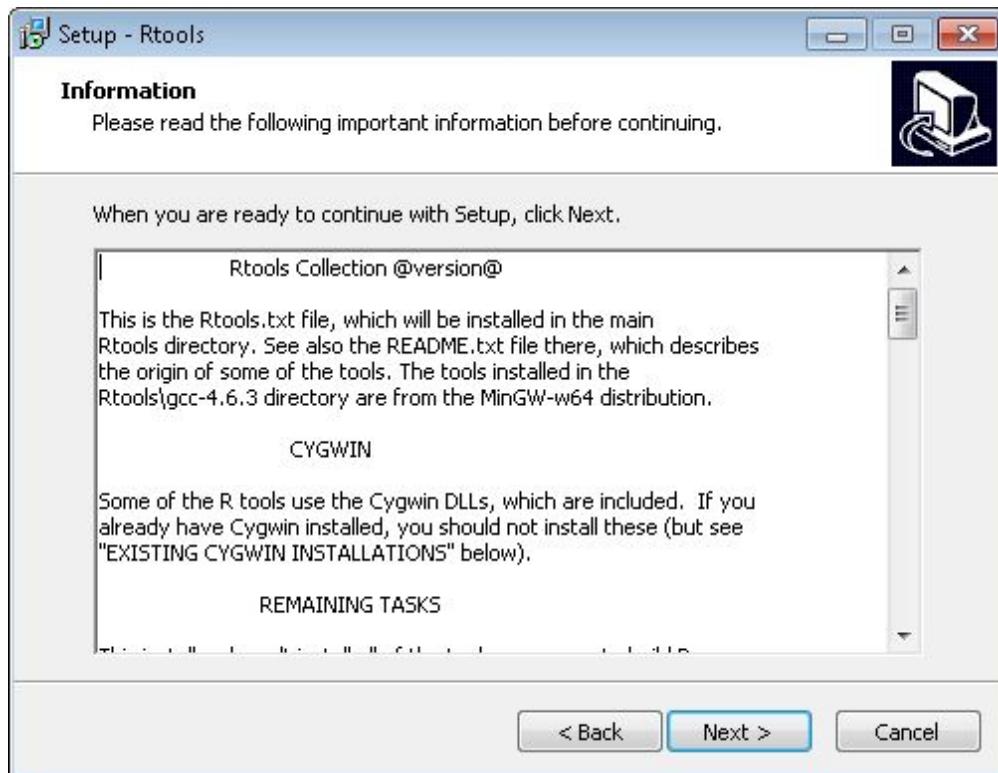
Download	R compatibility	Frozen?
Rtools31.exe	R 3.0.x to 3.1.x	No
Rtools30.exe	R >2.15.1 to R 3.0.x	Yes
Rtools215.exe	R >2.14.1 to R 2.15.1	Yes
Rtools214.exe	R 2.13.x or R 2.14.x	Yes
Rtools213.exe	R 2.13.x	Yes
Rtools212.exe	R 2.12.x	Yes

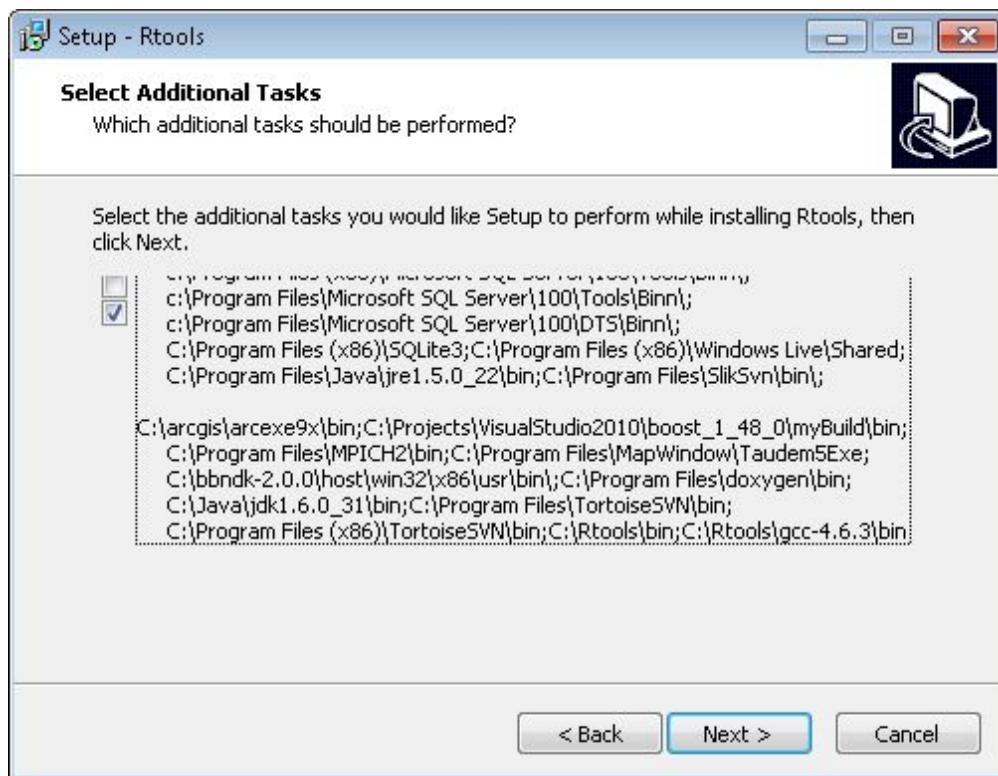
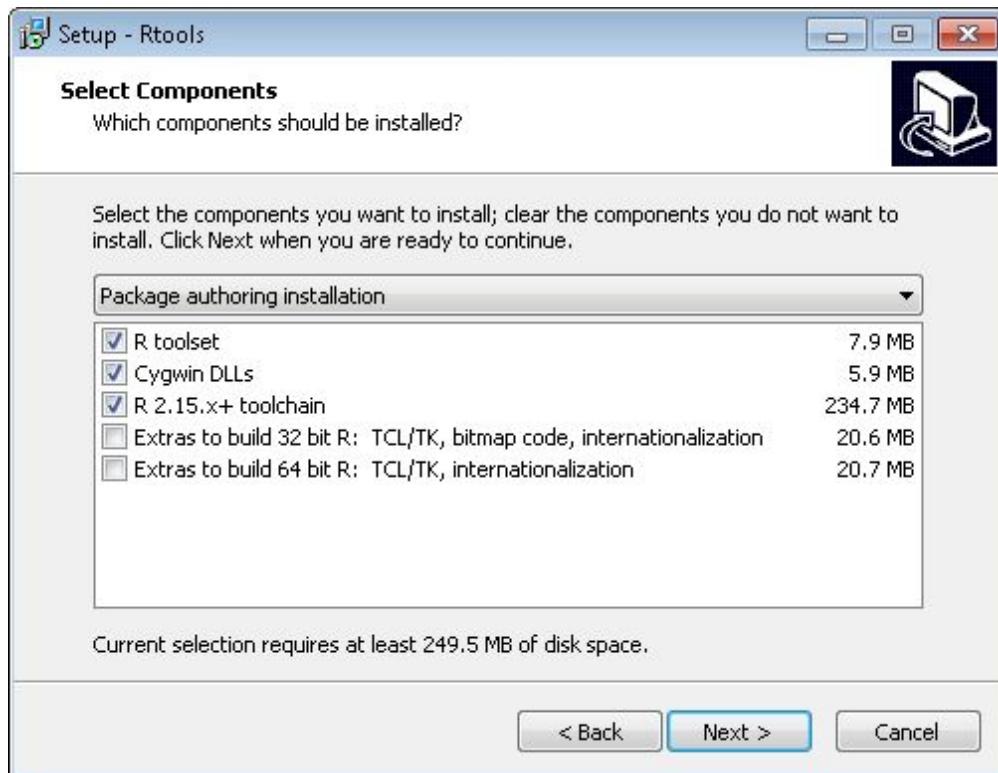
2. Run the installation file and select language.

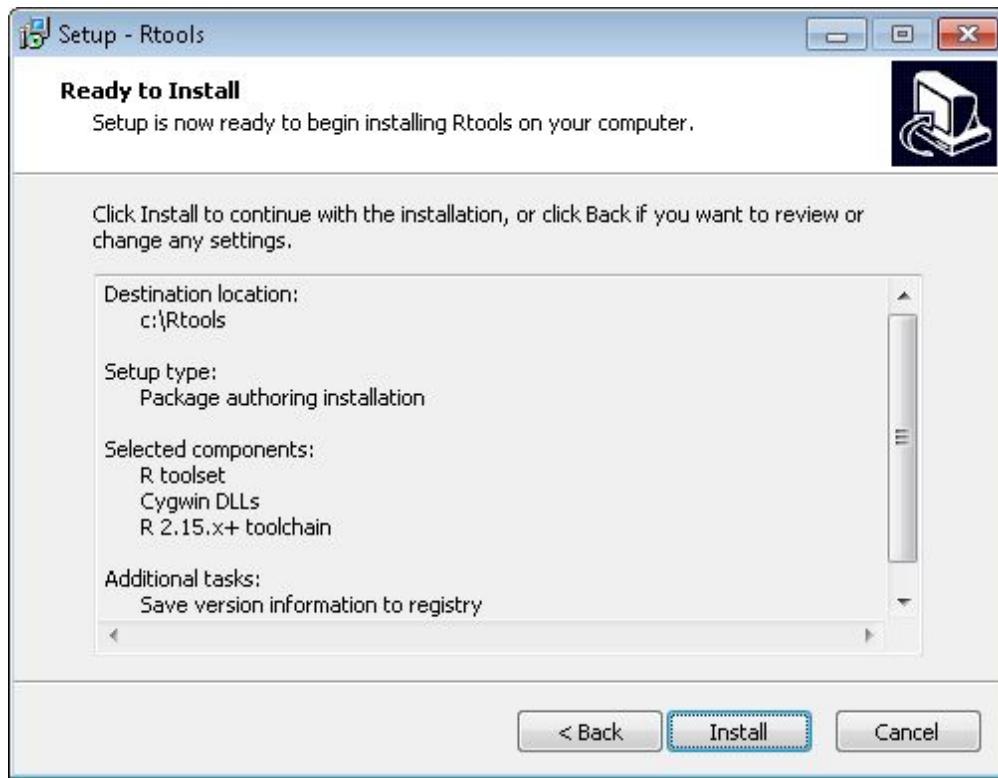


3. Click Next to accept all the default settings until the install button comes up.

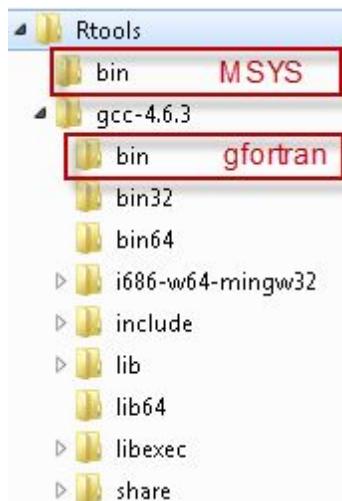








4. In the Rtools folder, MSYS is in bin folder and gfortran is located in gcc-4.6.3/bin (the folder may be different for other version).



5. Add C:\Rtools\bin and C:\Rtools\gcc-4.6.3\bin to PATH. This is optional. It's only required when SWAT is compiled in command line.

MinGW-w64

[Mingw-w64](http://sourceforge.net/projects/mingwbuilds/files/mingw-builds-install/mingw-builds-install.exe/download) delivers runtime, headers and libs for developing both 64 bit (x64) and 32 bit (x86) windows applications using GCC and other free software compilers.

Please note MinGW and MinGW-w64 are two separated projects and MinGW doesn't support 64 bit.

1. Visit

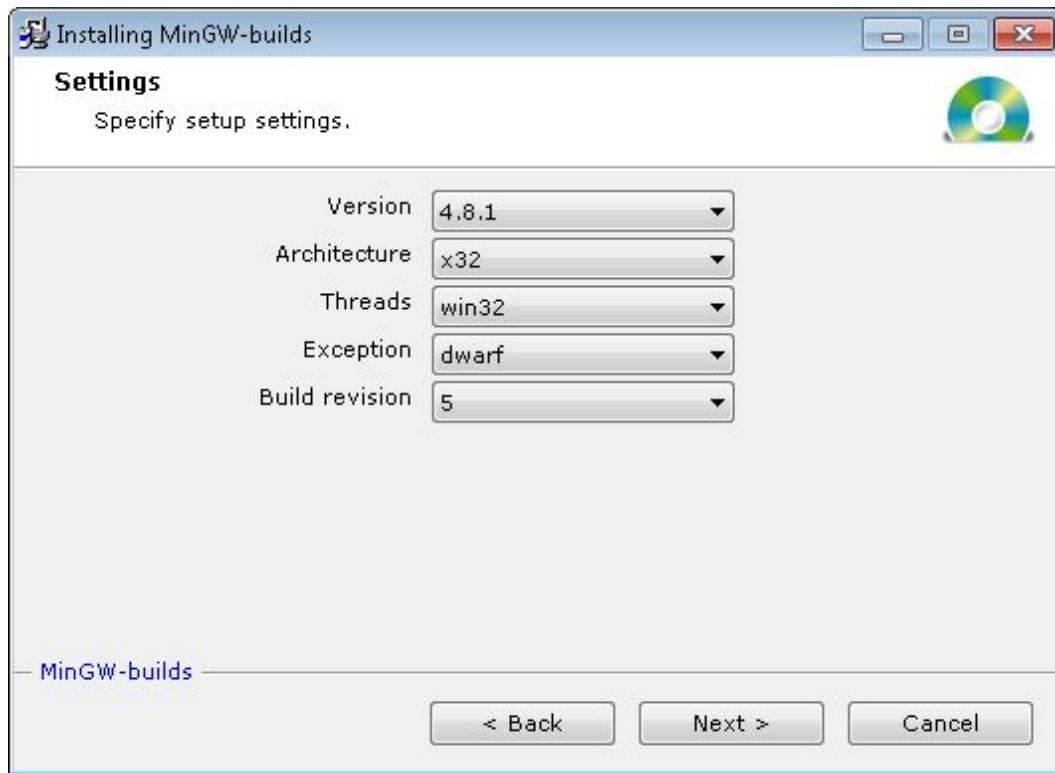
<http://sourceforge.net/projects/mingwbuilds/files/mingw-builds-install/mingw-builds-install.exe/download> to download MinGW Build Installer.



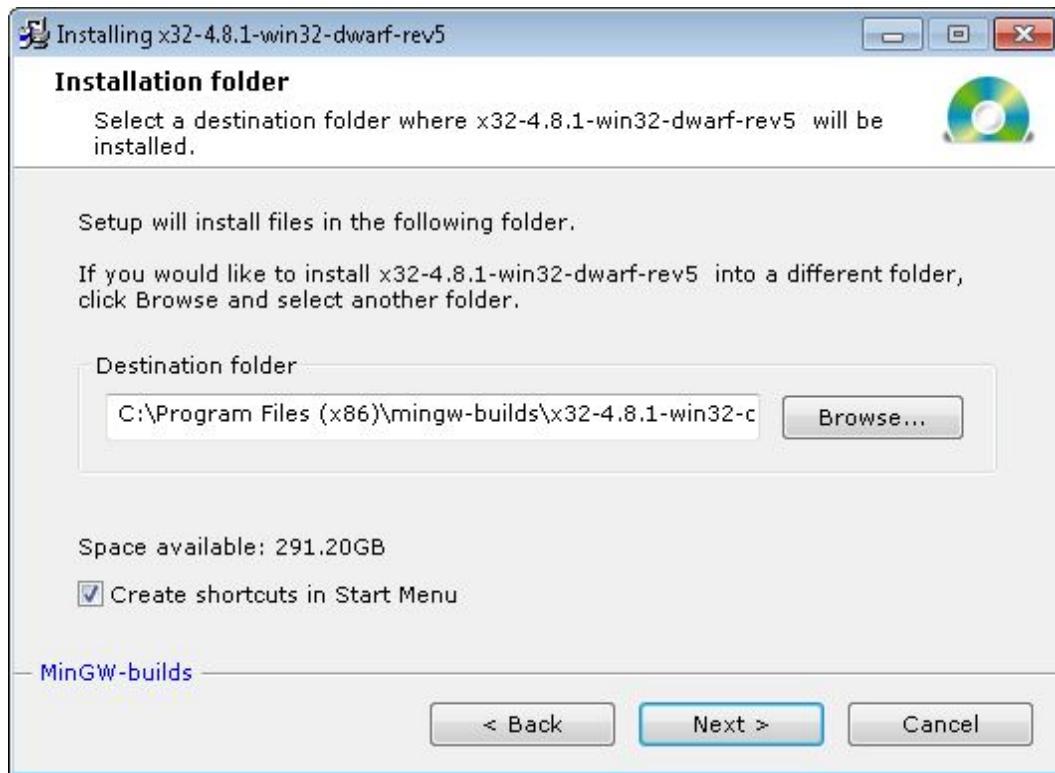
2. Run the installer and click next.



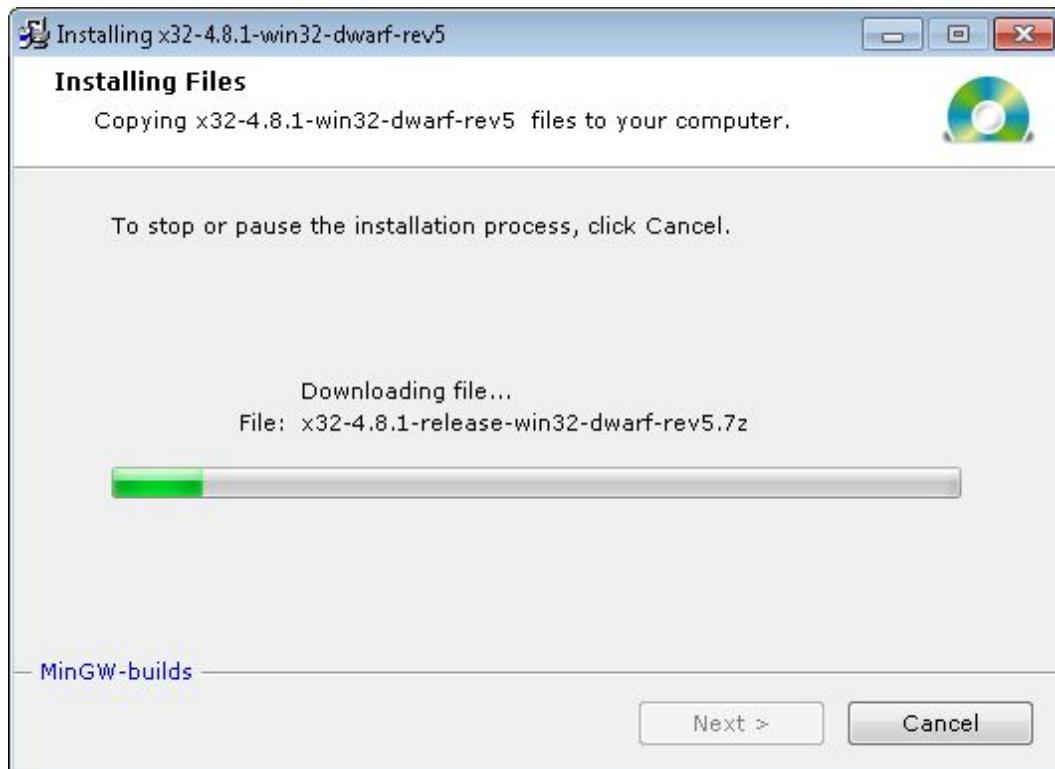
3. In the Settings window, select x32 for Architecture (even if you are working on 64-bit machine) , win32 for Threads and dwarf for Exception. Click next.

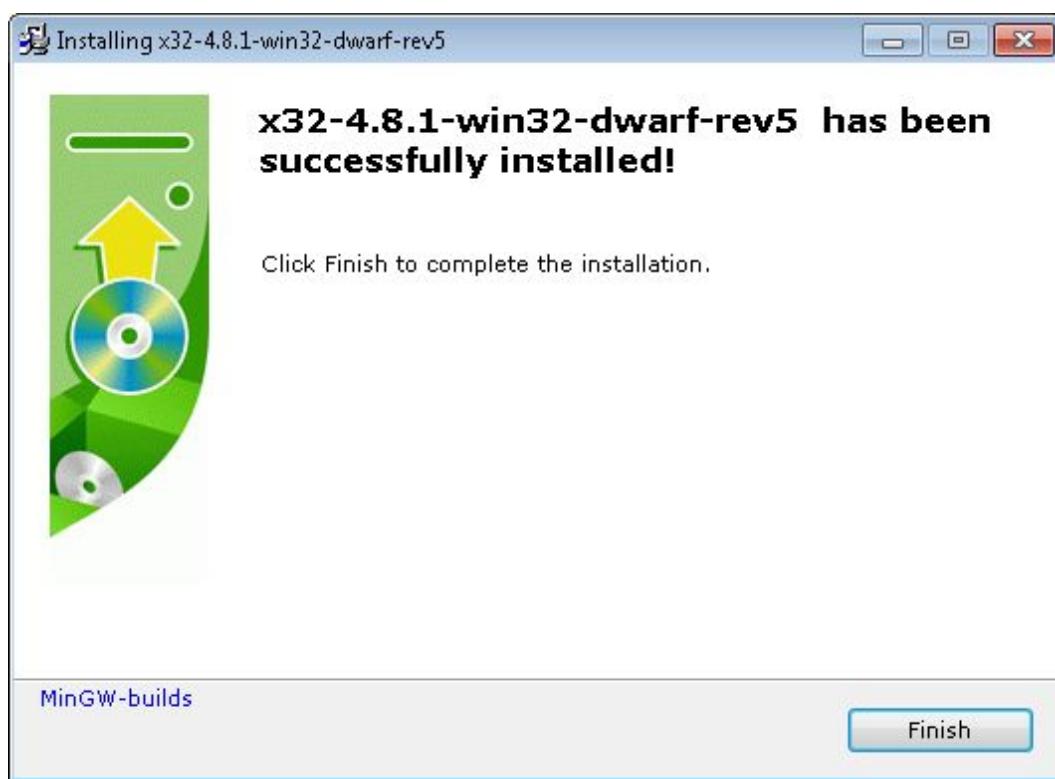
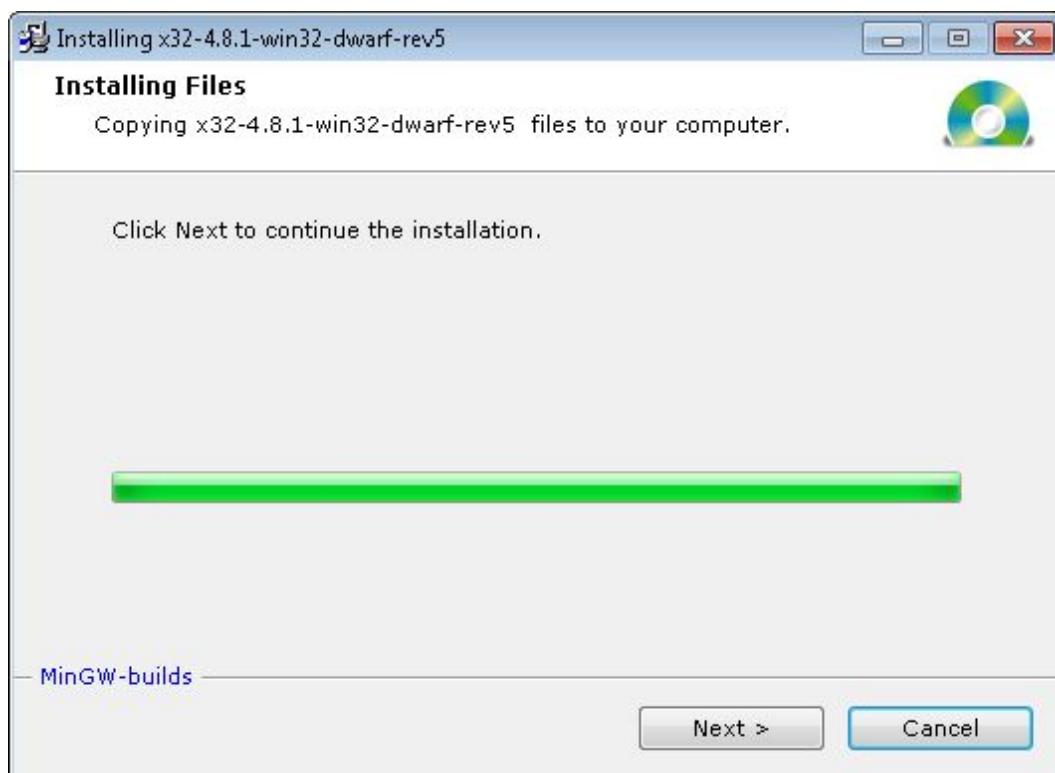


4. Change the Destination Folder if necessary.

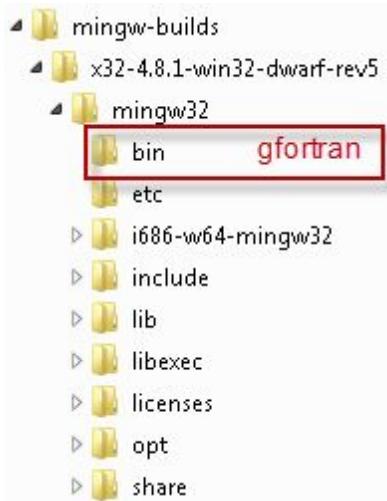


5. Click Next to start download and extract selected package.





The structure of the installation folder (default is C:\Program Files (x86)\mingw-builds\x32-4.8.1-win32-dwarf-rev5\mingw32) is shown in the graph below.



6. GFortran is installed and next is MSYS. Visit <http://sourceforge.net/projects/mingw-w64/files/External%20binary%20packages%20%20Win64%20hosted%29/MSYS%20%2832-bit%29/> and download the latest zip file.

MinGW-w64 - for 32 and 64 bit Windows

Brought to you by: [jon_y](#), [ktietz70](#), [mookmoz](#), [nightstrike](#)

[Summary](#) | [Files](#) | [Reviews](#) | [Support](#) | [Wiki](#) | [Mailing Lists](#) | [Hosted Apps](#) | [Tickets](#) | [News](#)

Looking for the latest version? [Download mingw-w64-v3.1.0.tar.bz2 \(7.0 MB\)](#)

[Home](#) / [External binary packages \(Win64 hosted\)](#) / MSYS (32-bit) 

Name	Modified	Size	Downloads / Week
Parent folder			
MSYS-20111123-src.7z	2011-11-23	109.2 MB	15  
MSYS-20111123.zip	2011-11-23	51.0 MB	223  
MSYS-20110526-src.7z	2011-05-26	103.4 MB	3  
MSYS-20110526.zip	2011-05-26	43.5 MB	4  
MSYS-20110309-src.7z	2011-03-09	92.8 MB	4  
MSYS-20110309.zip	2011-03-09	43.5 MB	4  
Totals: 6 Items		443.4 MB	253

7. Exact the zip file into a folder. It's highly to extract it to the MinGW-w64 installation folder (Default is C:\Program Files (x86)\mingw-builds\x32-4.8.1-win32-dwarf-rev5\mingw32).
8. Add **C:\Program Files (x86)\mingw-builds\x32-4.8.1-win32-dwarf-rev5\mingw32\bin** and **C:\Program Files (x86)\mingw-builds\x32-4.8.1-win32-dwarf-rev5\mingw32\msys\bin** to PATH. This is optional. It's only required when SWAT is compiled in command line.

MinGW

Please note MinGW doesn't support 64-bit. To build SWAT as 64-bit, please use Rtools or MinGW-w64.

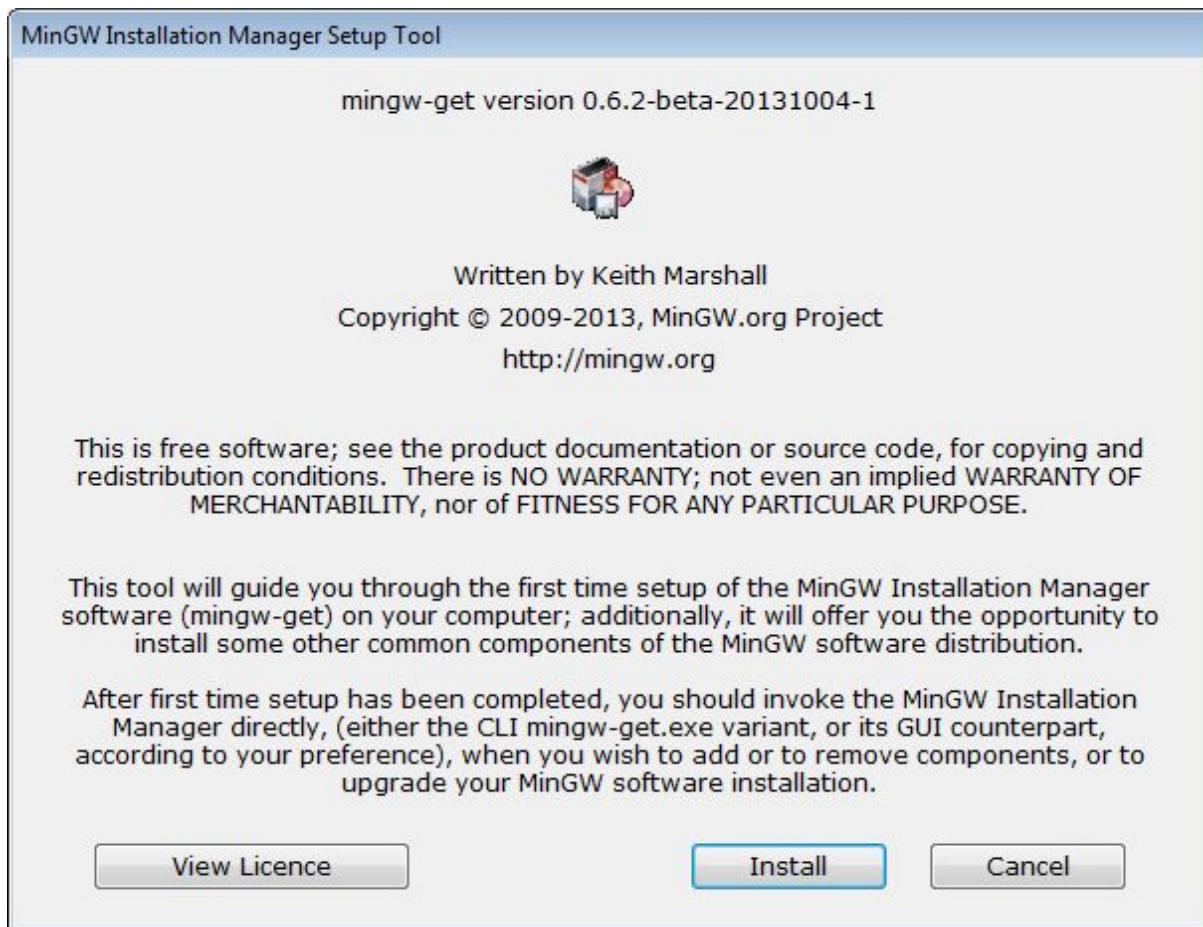
1. Download the automated GUI installer mingw-get-setup.exe at <https://sourceforge.net/projects/mingw/files/latest/download>.



- Double-click mingw-get-setup.exe and accept all the default settings. A shortcut for MinGW installer would be created on desktop once the process is done.

Note:

It's highly recommended to use the default installation directory C:\MinGW. If it's not possible, make sure the installation directory doesn't have any space.



MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1



Step 1: Specify Installation Preferences

Installation Directory

C:\MinGW **No Space in the directory, recommend to use default**

Change

If you elect to change this, you are advised to avoid any choice of directory which includes white space within the absolute representation of its path name.

User Interface Options

Both command line and graphical options are available. The command line interface is always supported; the alternative only if you choose the following option to ...

... also install support for the graphical user interface.

Program shortcuts for launching the graphical user interface should be installed ...

... just for me (the current user), or for all users * ...

... in the start menu, and/or on the desktop.

* selection of this option requires administrative privilege.

View Licence

Continue

Cancel

MinGW Installation Manager Setup Tool

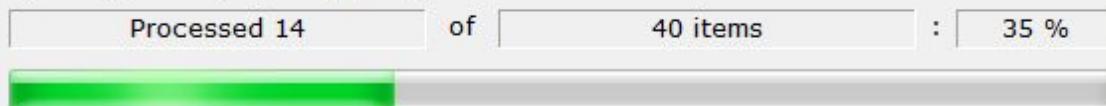
mingw-get version 0.6.2-beta-20131004-1



Step 2: Download and Set Up MinGW Installation Manager

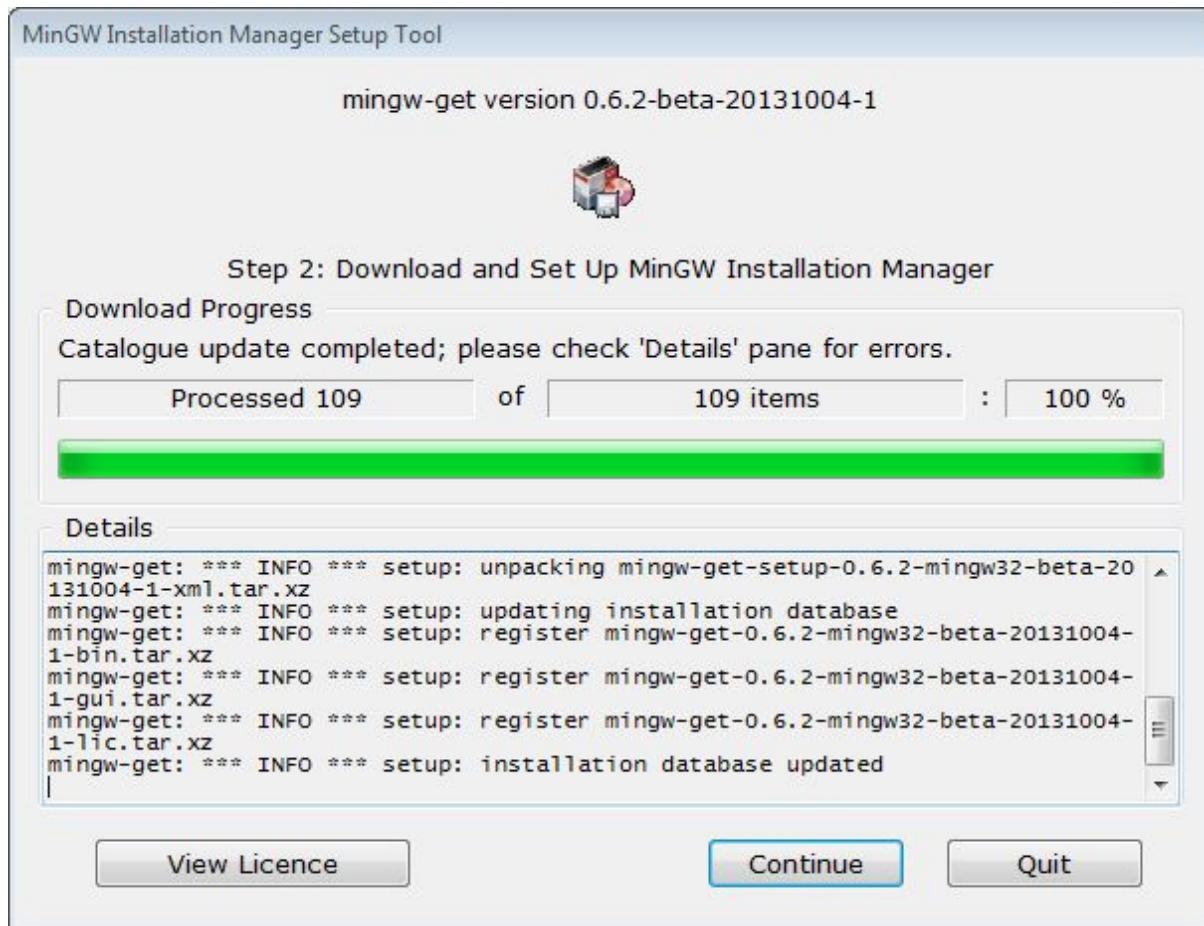
Download Progress

Updating catalogue: mingw32-gettext.xml

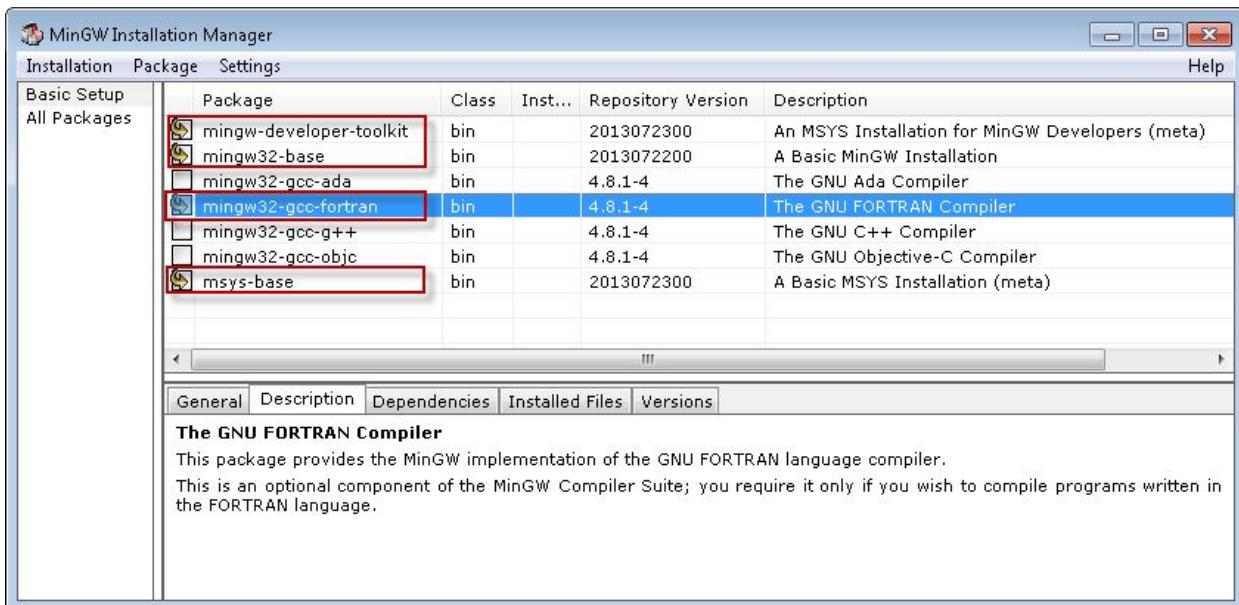


Details

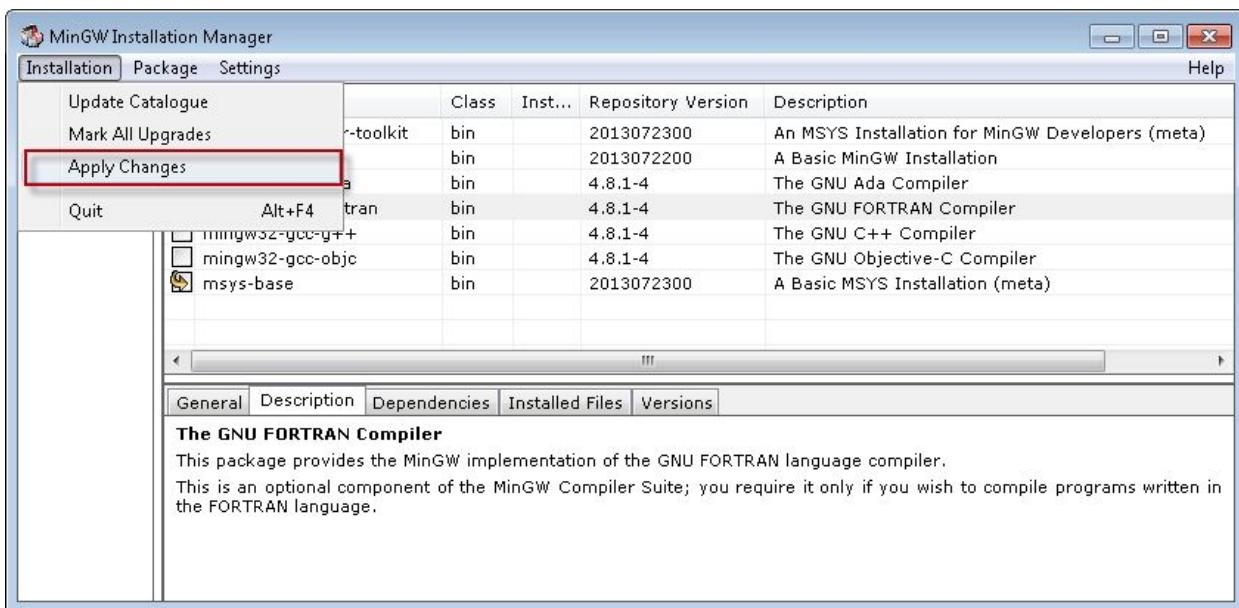
```
mingw-get: *** INFO *** setup: unpacking mingw-get-setup-0.6.2-mingw32-beta-20131004-1-xml.tar.xz
mingw-get: *** INFO *** setup: updating installation database
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-bin.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-gui.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-1-lic.tar.xz
mingw-get: *** INFO *** setup: installation database updated
```



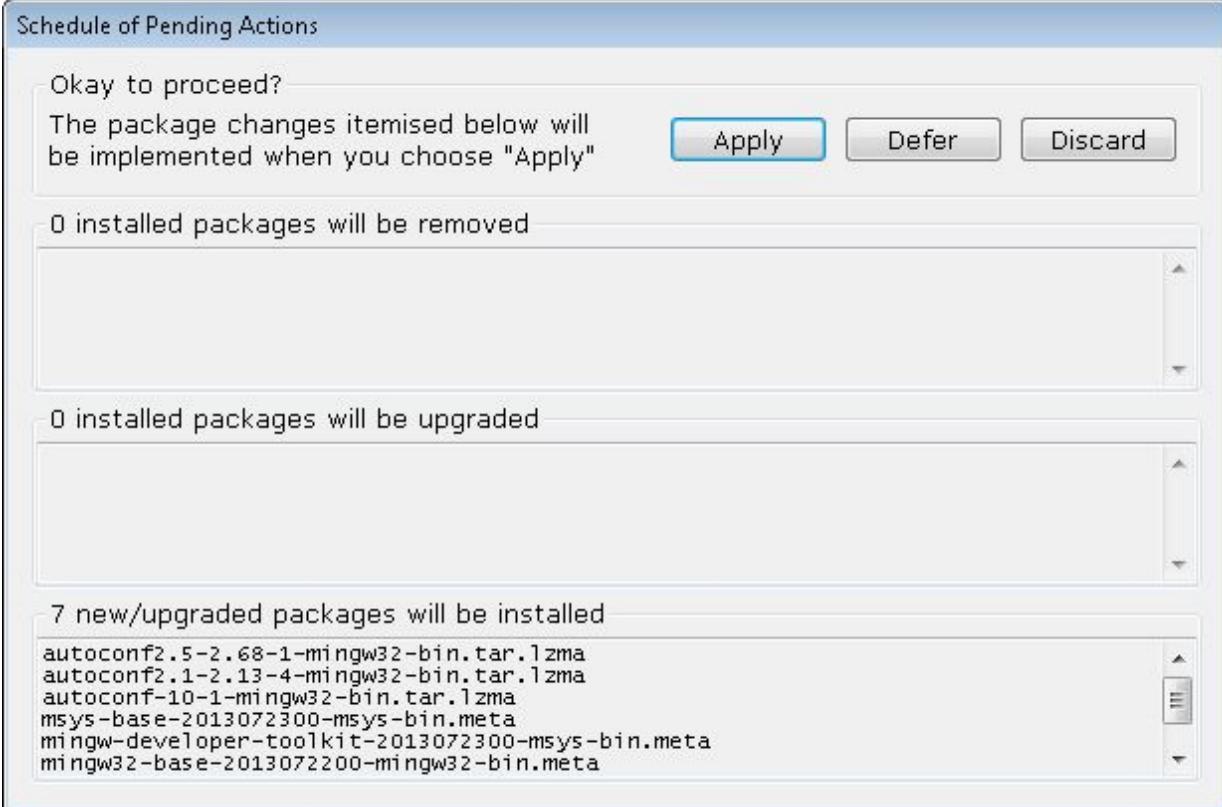
3. Open MinGW Installation Manager on the desktop, select **Basic Setup** in the left panel and choose package **mingw32-base**, **mingw32-gcc-fortran**, **mingw-developer-toolkit** and **msys-base**.

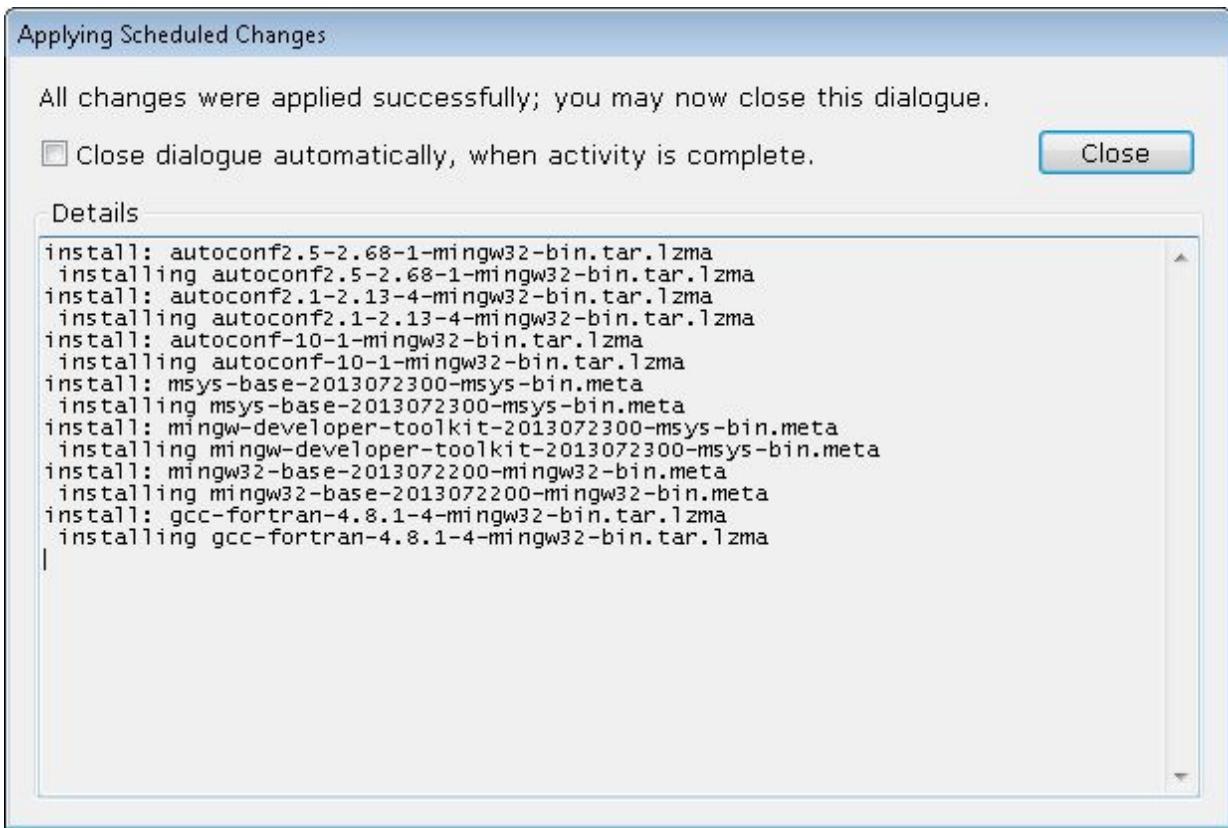


4. Select menu **Installation -> Apply Changes** to start installation.

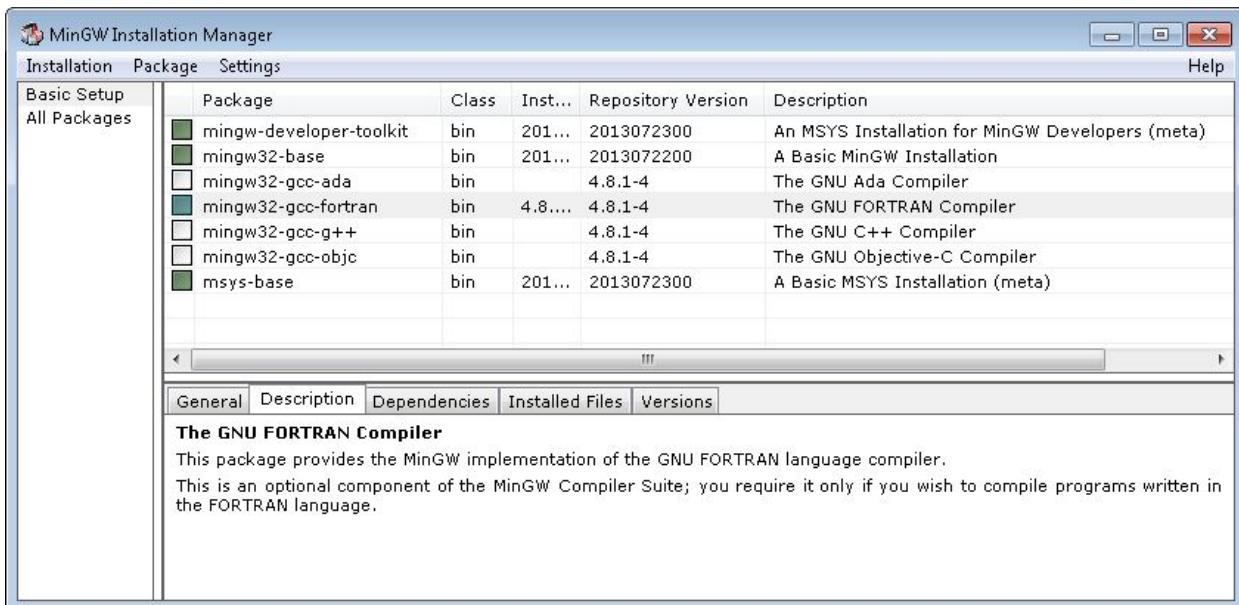


5. Click Apply in the coming window to start downloading selected packages and then extracting them into the installation folder.

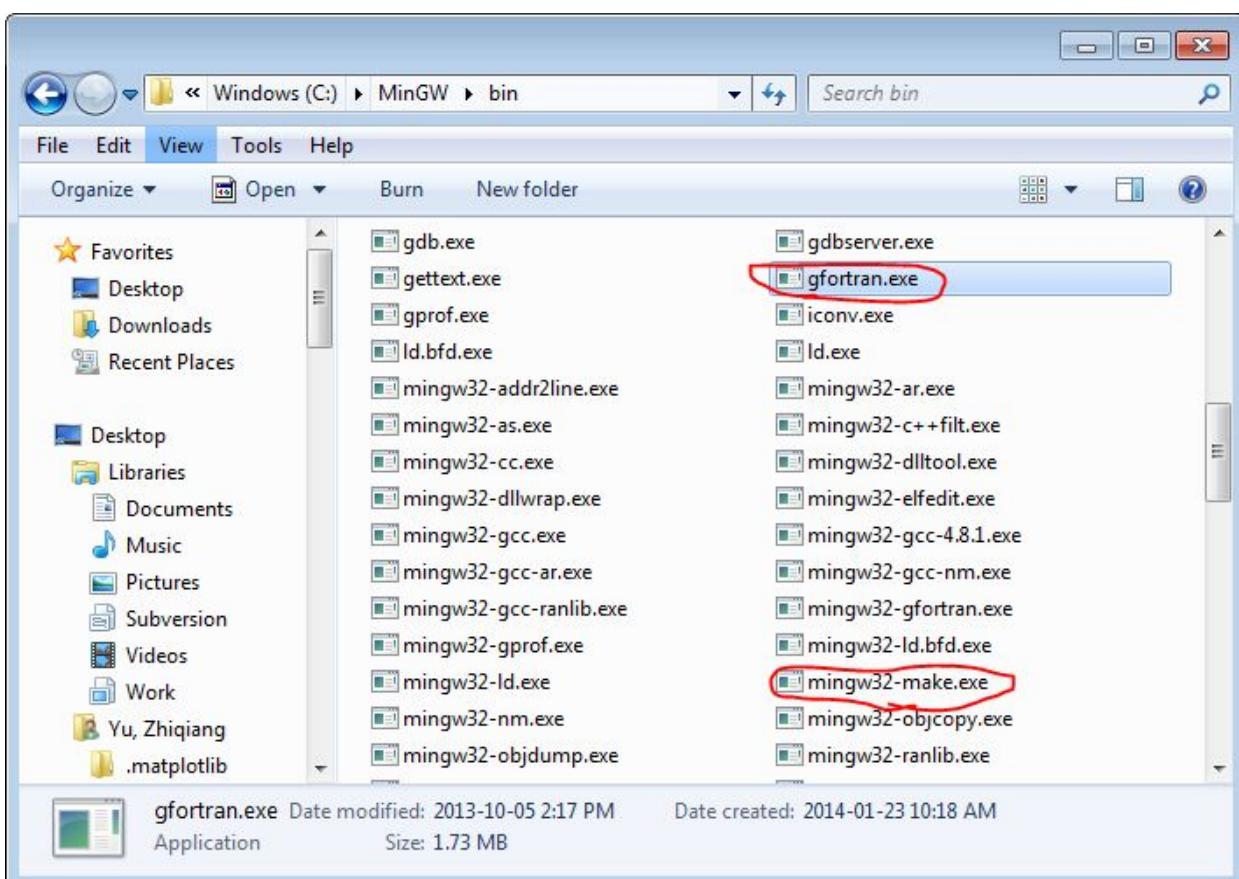




6. The four packages will be selected after the installation is done.



7. Go to the installation folder (C:\MinGW\bin) to check mingw32-make.exe and gfortran.exe.



8. Add C:\MinGW\bin and C:\MinGW\msys\1.0\bin to PATH. This is optional. It's only

required when SWAT is compiled in command line.

Downloading SWAT Source Codes

Visit SWAT website <http://swat.tamu.edu/software/swat-executables/> to download the latest SWAT source codes.

SWAT2012 rev. 627 was released June 24, 2014. Read the [revision history](#).

DOWNLOAD EXECUTABLES

The SWAT2012 [source code](#) and [input/output documentation](#) is also available.

Download the [SWAT executable for Linux](#) (rev. 583, January 17, 2013).

Get Makefile

Makefile defines the rules for compiling source codes. The Makefile for SWAT could be downloaded directly or generated with Makefile Generator.

Downloading Makefile

Visit my blog using following link to download the makefile directly.

<http://zhiqiangyu.wordpress.com/2014/02/10/makefile-updated-1-no-need-to-modify-main-f-any-more-2-a-single-makefile-3-64-bit-build/>.

Generating Makefile with Makefile Generator

Makefile Generator is a .NET program which will generate the Makefile automatically based on the version of SWAT source codes. It supports SWAT 2009 and SWAT 2012. The source code of this program is host in Google Code and the executable could be downloaded using following link.

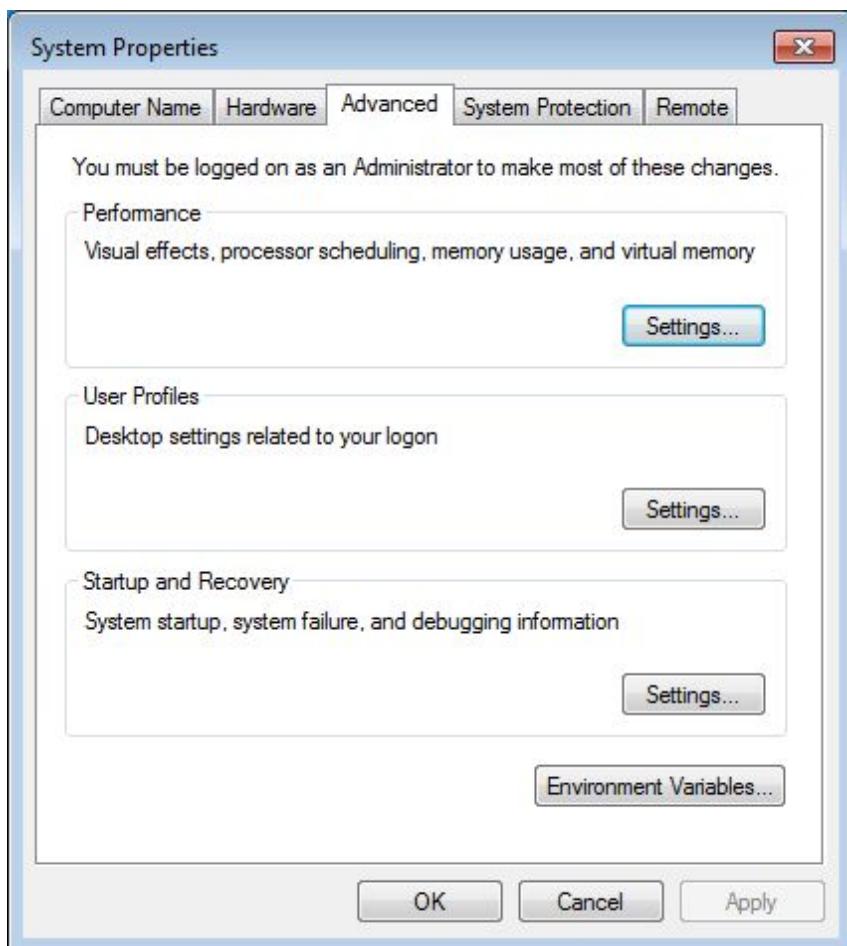
<https://swat-eclipse.googlecode.com/svn/trunk/GenerateMakefile/GenerateMakefile/bin/Release/GenerateMakefile.exe>

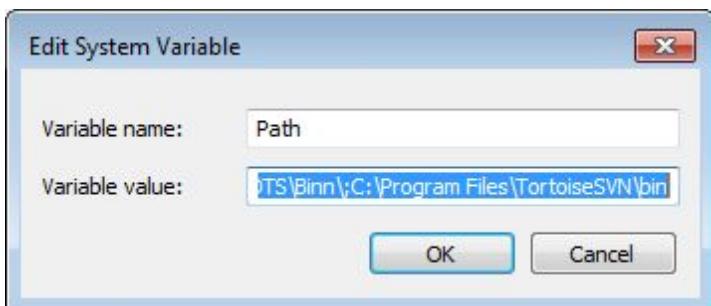
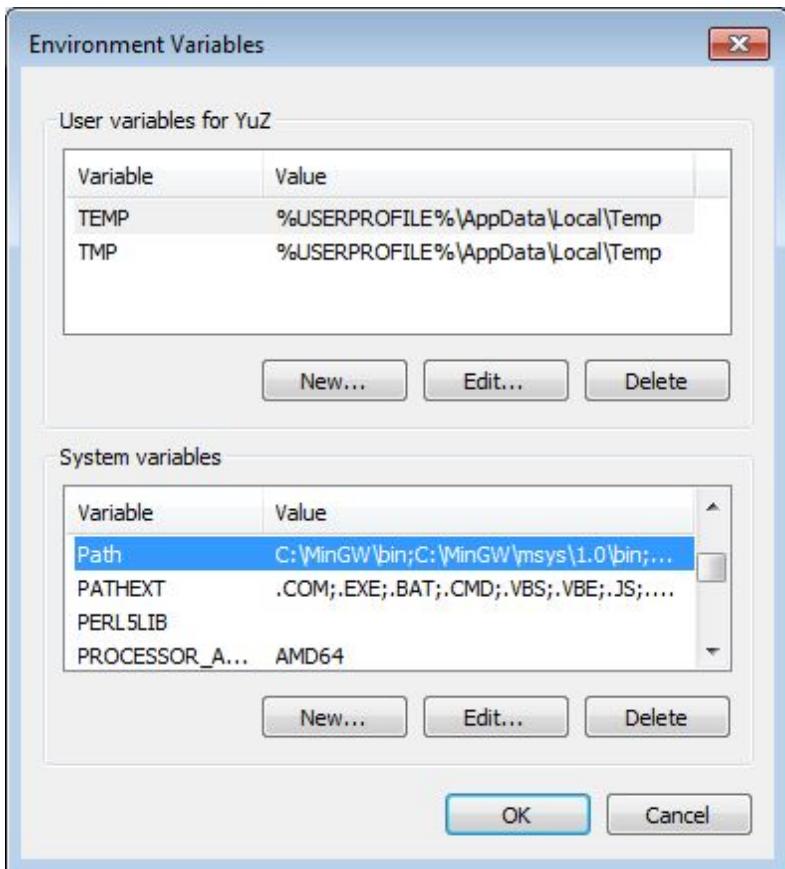
Download this program and copy the file into the SWAT source code folder. Double-click to run the program and the Makefile will be generated in the same folder. Please note that the existing Makefile will be overwritten.

Compiling SWAT in Command Line

With Makefile, the SWAT source codes could be compiled using GFortran in command line.

1. Locate gfortran.exe and make.exe. The default locations would be as following.
 - **Rtools:** C:\Rtools\gcc-4.6.3\bin, C:\Rtools\bin
 - **MinGW-w64:** C:\Program Files (x86)\mingw-builds\x32-4.8.1-win32-dwarf-rev5\mingw32\bin, C:\Program Files (x86)\mingw-builds\x32-4.8.1-win32-dwarf-rev5\mingw32\msys\bin
 - **MinGW:** C:\MinGW\bin, C:\MinGW\msys\1.0\bin
2. Add locations found in step 1 into system variable PATH.





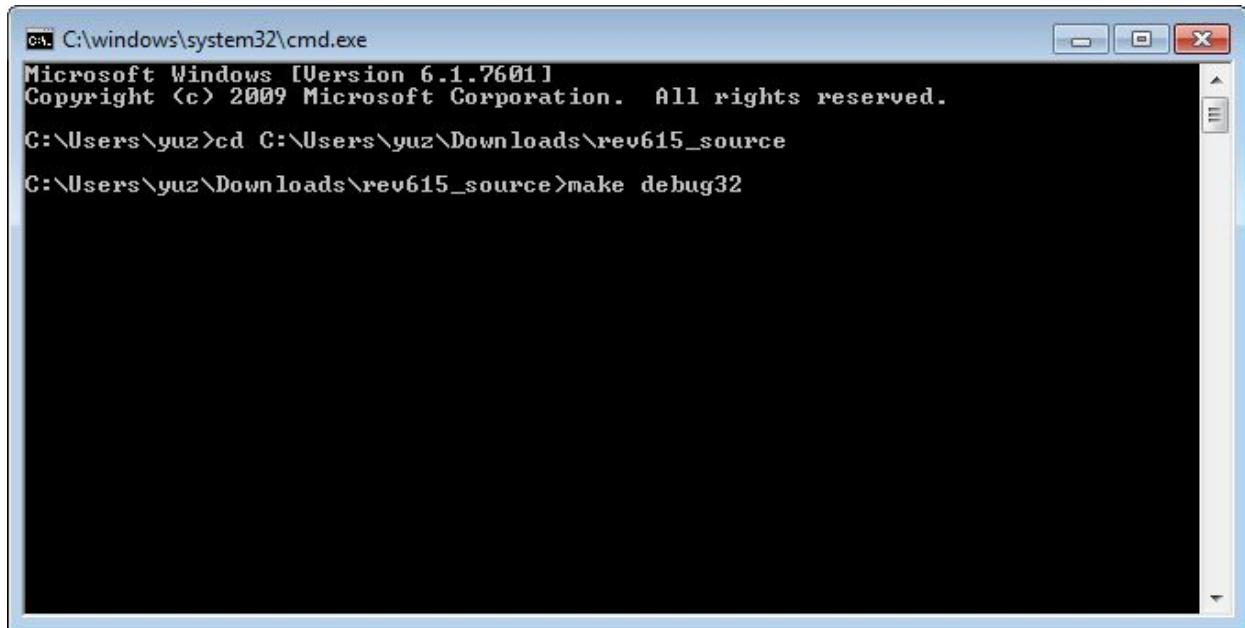
3. Open command line by typing cmd in start menu and navigate to SWAT source codes folder. Type one of the four commands in following table to start compiling SWAT. The whole compiling process would take around 15 minutes to finish.

Command	32-bit or 64-bit	debug or release	executable name	Comment
make debug32	32	debug	swat_debug32	
make debug64	64	debug	swat_debug64	Doesn't support by MinGW

make rel32	32	release	swat_rel32	
make rel64	64	release	swat_rel64	Doesn't support by MinGW

Note:

The **make** command also could be replaced by **mingw32-make** if MinGW and MinGW-w64 is used.



A screenshot of a Microsoft Windows Command Prompt window titled "C:\windows\system32\cmd.exe". The window shows the following text output:

```
C:\ C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\yuz>cd C:\Users\yuz\Downloads\rev615_source
C:\Users\yuz\Downloads\rev615_source>make debug32
```

```
C:\ C:\windows\system32\cmd.exe
.o debug32/routes.o debug32/rsedaa.o debug32/rseday.o debug32/rsedmon.o debug32
/rsedyr.o debug32/rtbact.o debug32/rtday.o debug32/rteinit.o debug32/rthmusk.o d
ebug32/rthpest.o debug32/rthr.o debug32/rthsed.o debug32/rtmusk.o debug32/rtout.
.o debug32/rtpest.o debug32/rtsed.o debug32/rtsed_bagnold.o debug32/rtsed_kodatie
.o debug32/rtsed_molinasmwu.o debug32/rtsed_yangsand.o debug32/sat_excess.o debu
g32/save.o debug32/saveconc.o debug32/schedule_ops.o debug32/sched_mgt.o debug32
/simulate.o debug32/sim_initday.o debug32/sim_inityr.o debug32/slrgen.o debug32/
smeas.o debug32/snom.o debug32/soil_chem.o debug32/soil_par.o debug32/soil_phys.
o debug32/soil_write.o debug32/solp.o debug32/solt.o debug32/std1.o debug32/std2
.o debug32/std3.o debug32/stdaa.o debug32/storeinitial.o debug32/structure.o deb
ug32/subaa.o debug32/subbasin.o debug32/subday.o debug32/submon.o debug32/substo
r.o debug32/subwg.o debug32/subyr.o debug32/subbasin.o debug32/sumhyd.o debu
g32/sumv.o debug32/surface.o debug32/surfstor.o debug32/surfst_h2o.o debug32/sur
q_daycn.o debug32/surq_greenampt.o debug32/swbl.o debug32/sweep.o debug32/swu.o
debug32/tair.o debug32/tgen.o debug32/theta.o debug32/tillfactor.o debug32/tillm
ix.o debug32/tmeas.o debug32/tran.o debug32/transfer.o debug32/tstr.o debug32/tt
coef.o debug32/ttcoef_wway.o debug32/urban.o debug32/urbanhr.o debug32/urb_bmp.o
debug32/varinit.o debug32/vbl.o debug32/virtual.o debug32/volq.o debug32/washp.
o debug32/watbal.o debug32/water_hru.o debug32/watqual.o debug32/watqual2.o debu
g32/wattable.o debug32/watuse.o debug32/weatgn.o debug32/wetlan.o debug32/wmeas.
o debug32/wndgen.o debug32/writea.o debug32/writeaa.o debug32/writed.o debug32/w
ritem.o debug32/xmon.o debug32/ysed.o debug32/zero0.o debug32/zero1.o debug32/ze
ro2.o debug32/zeroini.o debug32/zero_urbn.o -m32 -static -o swat_debug32
C:\Users\yuz\Downloads\rev615_source>
```

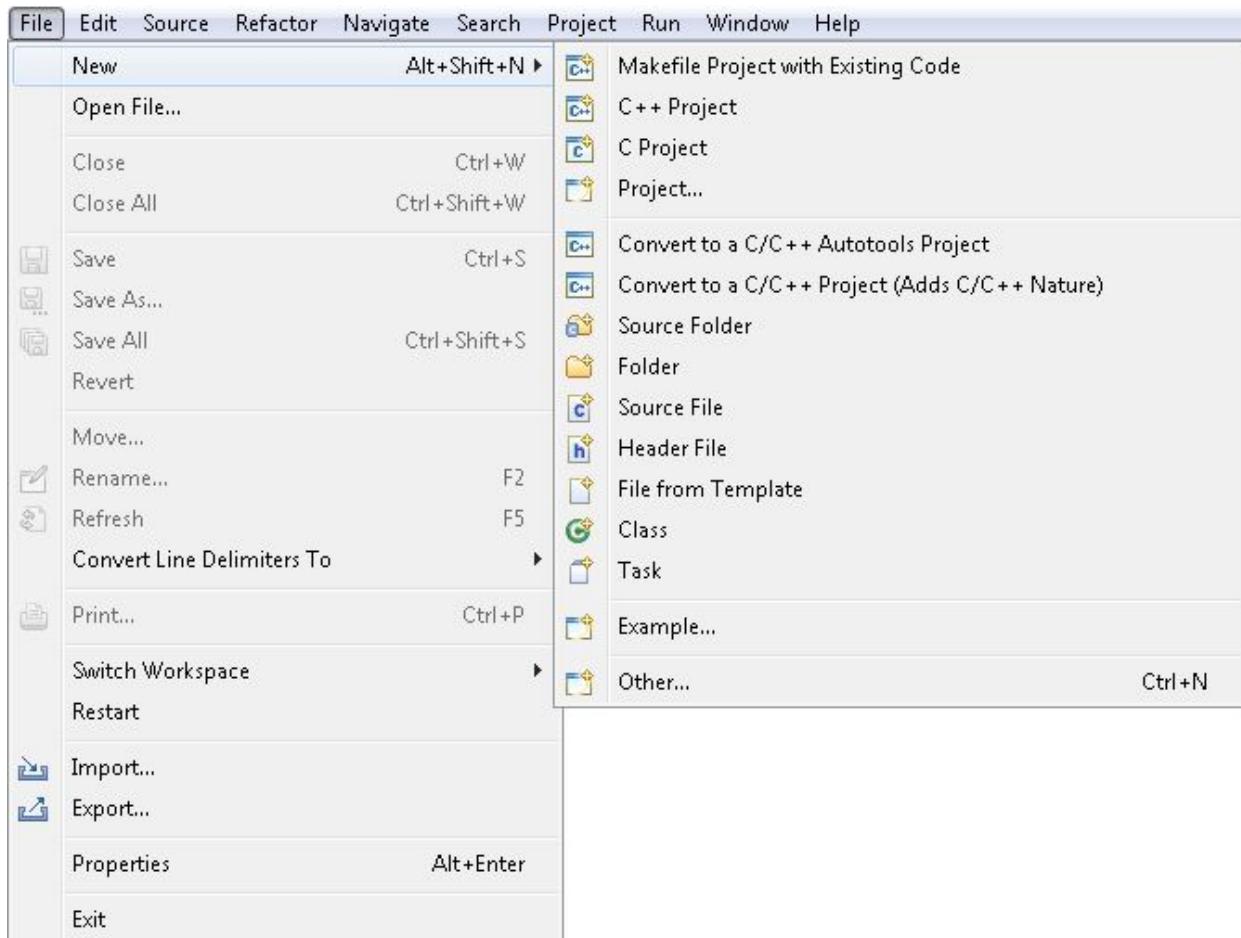
Compiling and Debugging SWAT in Eclipse

To debug SWAT, e.g set breakpoints and check variable values, a project is needed to be created in Eclipse.

Create a Fortran Project

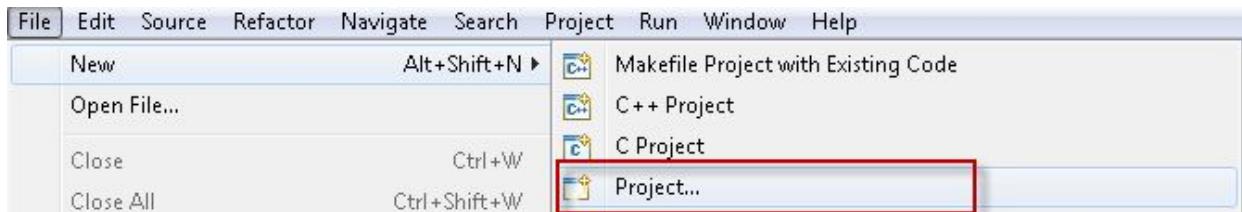
Find Fortran Project... Menu

The Fortran Project option is not listed in menu **File -> New** when PTP is used the first time.

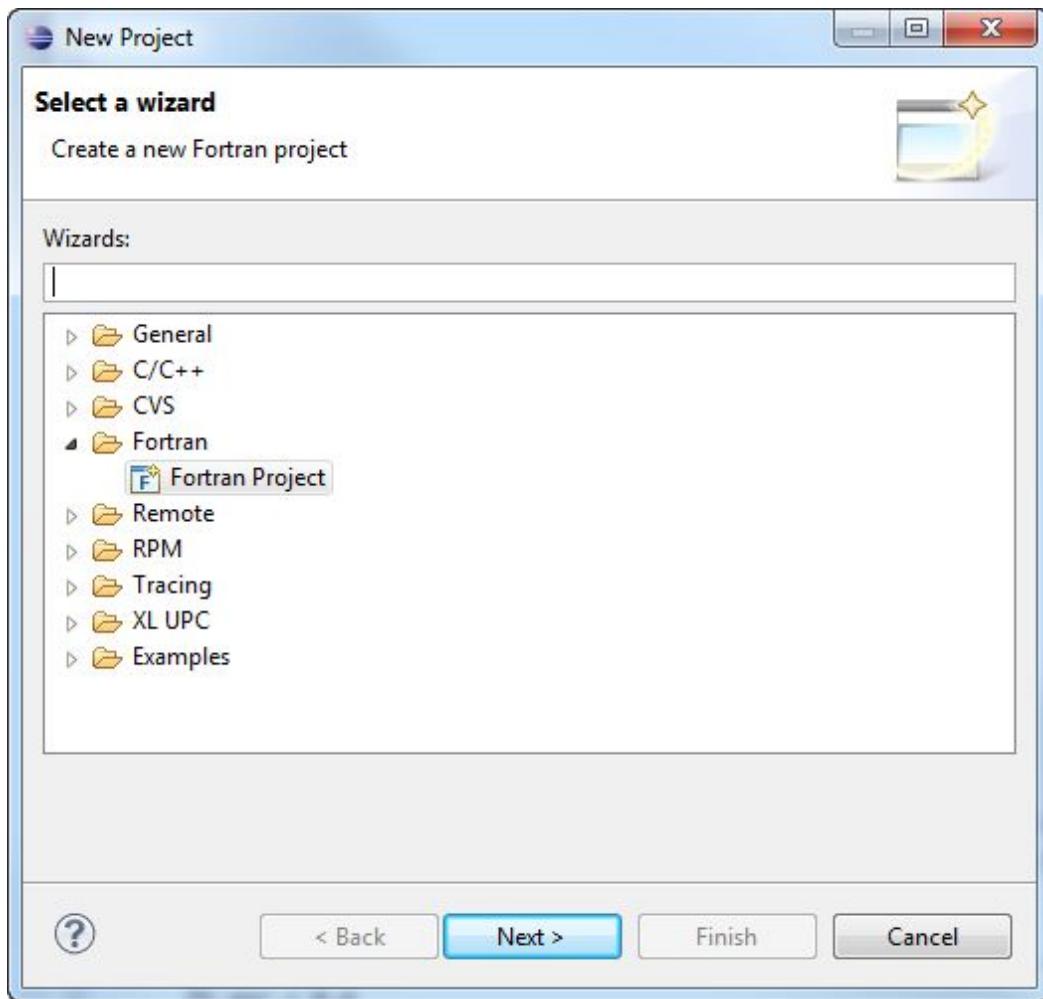


To add Fortran Project in the menu, follow steps below.

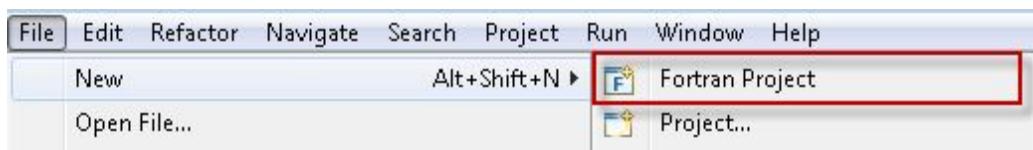
1. select **Projects...** to open **New Project** window.



2. Find Fortran project under Fortran folder. Select it and click Next to create the project.

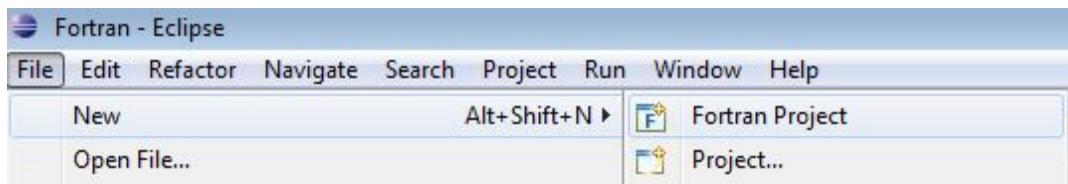


3. When the project is created, you will be asked if you want to open Associated perspective. Check Remember my decision and click Yes. The **Fortran Project** option would be list in menu **File -> New**.

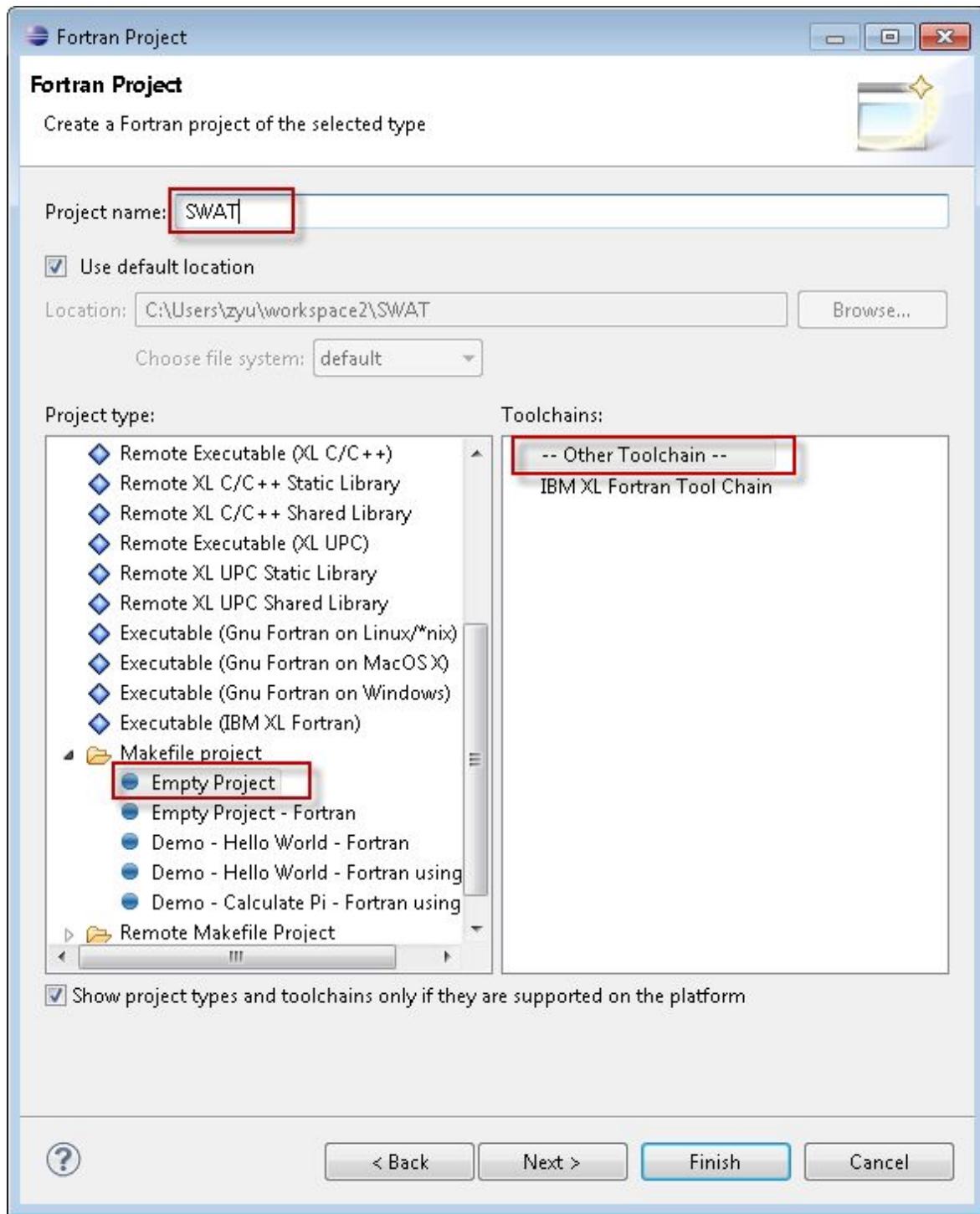


Create an Empty Makefile Project

1. Select **File** -> **New** -> **Fortran Project**



2. Give the project a name, e.g. SWAT. Select **Empty Project** under Makefile project in the lower left panel and select -- **Other Toolchain** -- in the lower right panel. Click finish to create the project.

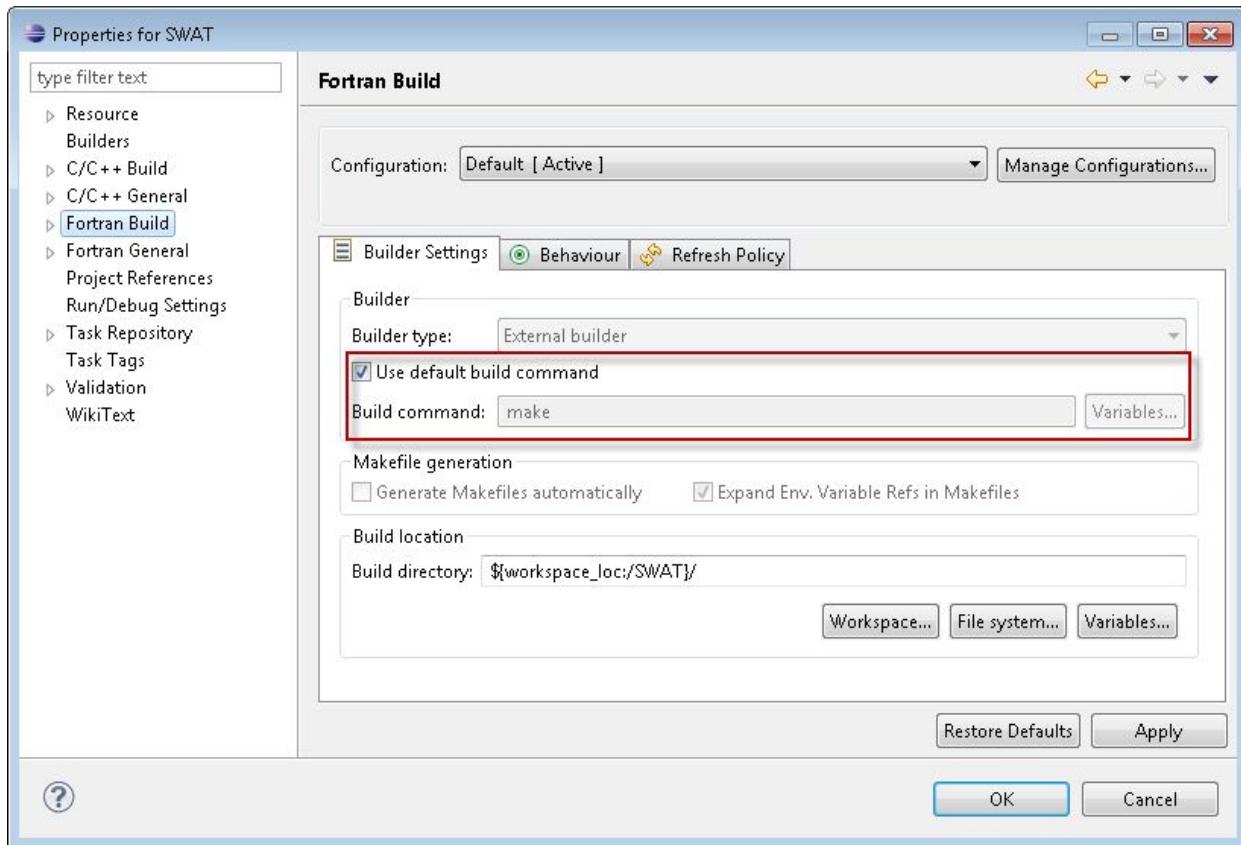


Setup the Project

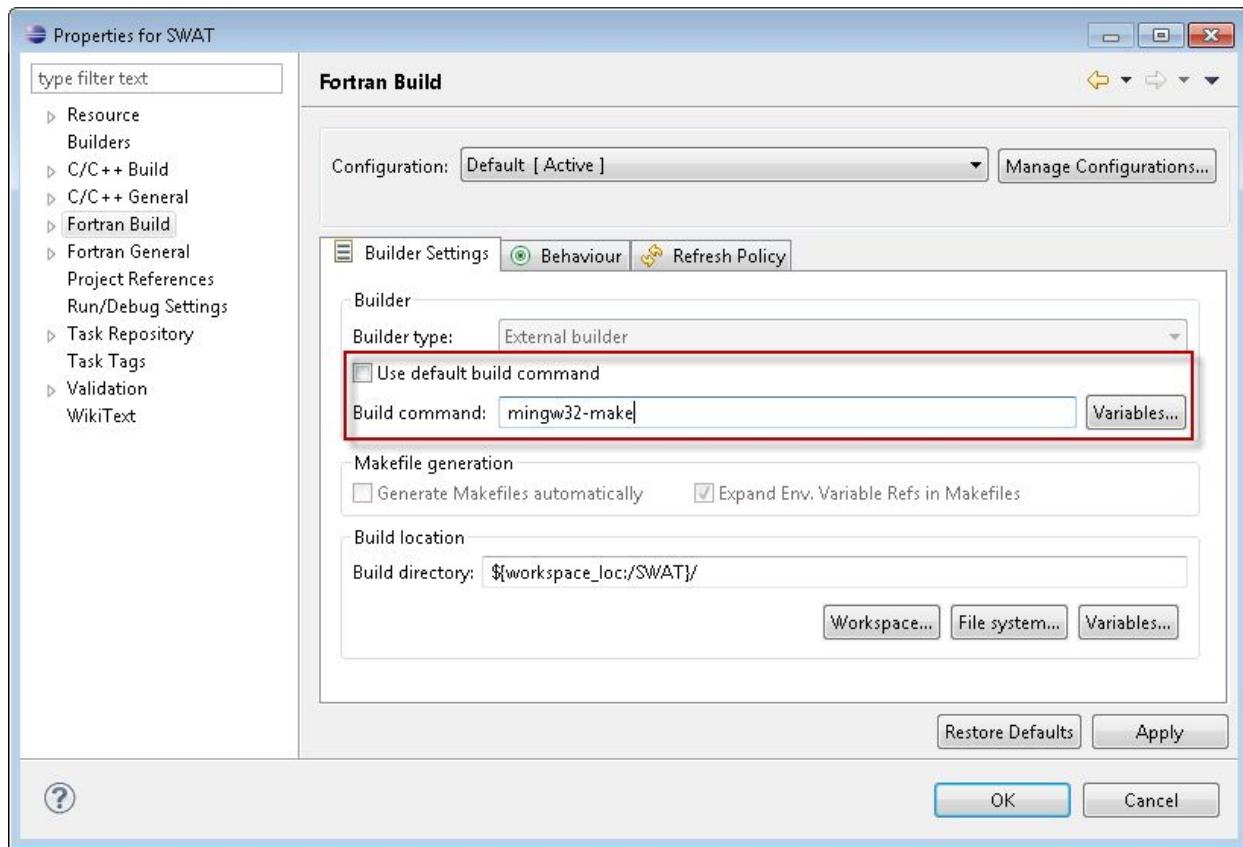
To setup the project, open the **Property** window through right click on the project and select **Properties**.

Build Command

Select **Fortran Build** in the left panel, and change the build command if necessary. The default build command is make, which would work for all MinGW platforms.



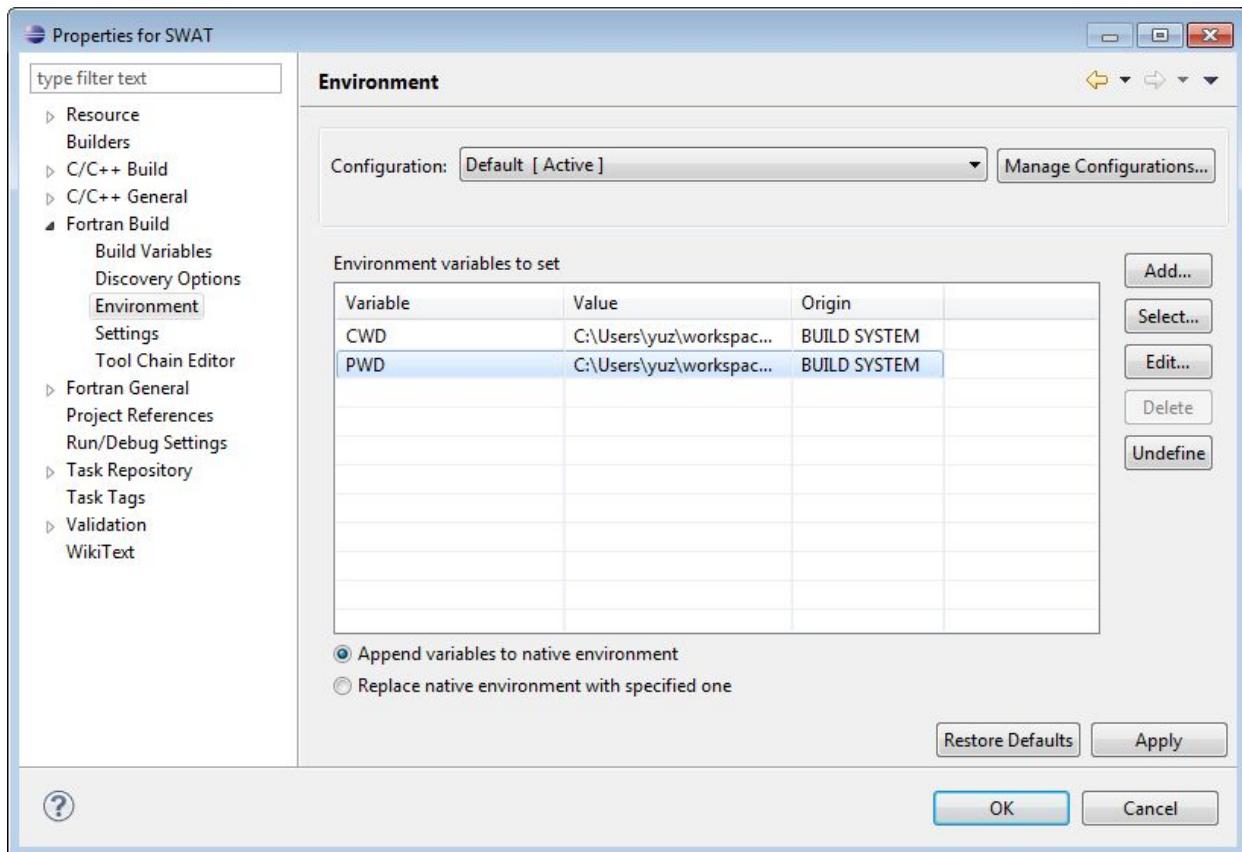
If prefer using mingw32-make, un-check **Use default build command** and change **Build Command** to **mingw32-make**.



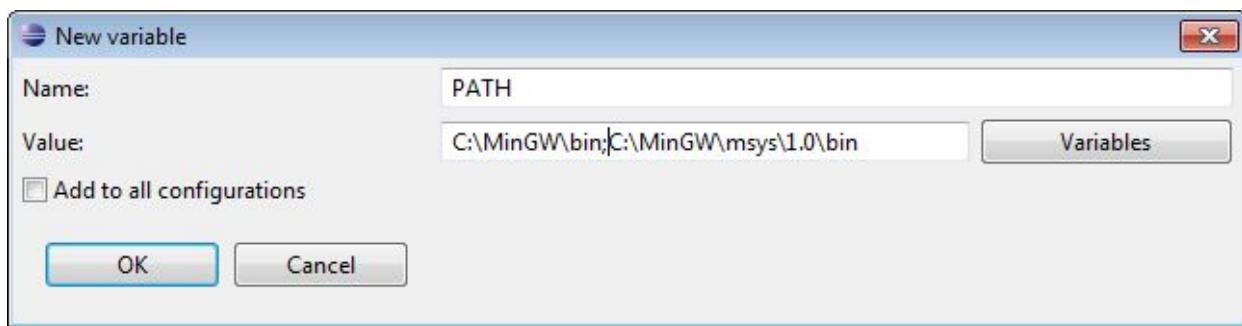
Fortran Build Environment

If the gfortran and msys folder is not added to the PATH system variable (which is not possible for users without administrative access), they needs to be added to the PATH variable in Fortran build environment. That's where Eclipse will search for **gfortran**, **make** (or **mingw32-make**) and **gdb**.

1. Select **Fortran Build -> Environment** in the left panel, and click **Add..** to popup the New Variable window.



- Input PATH as the variable name and the locations of gfortran and msys as the value. Click OK to add this variable.

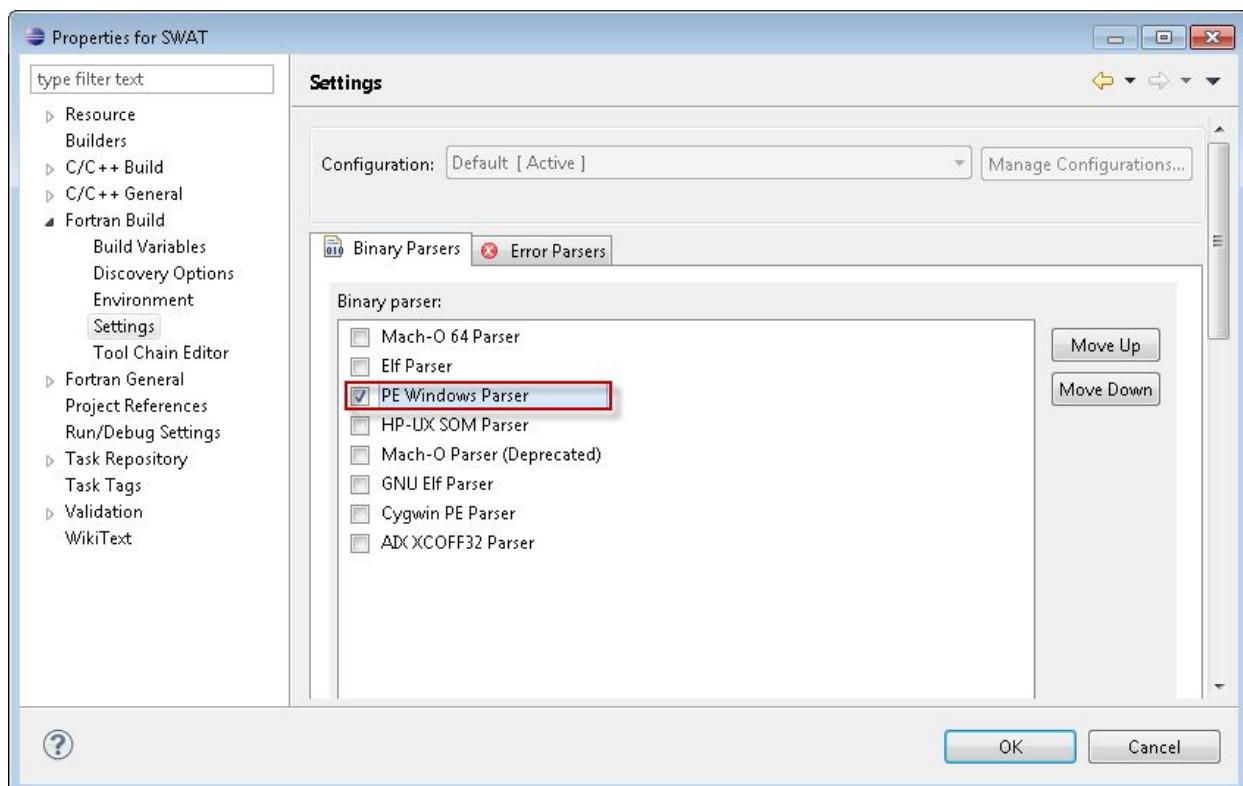
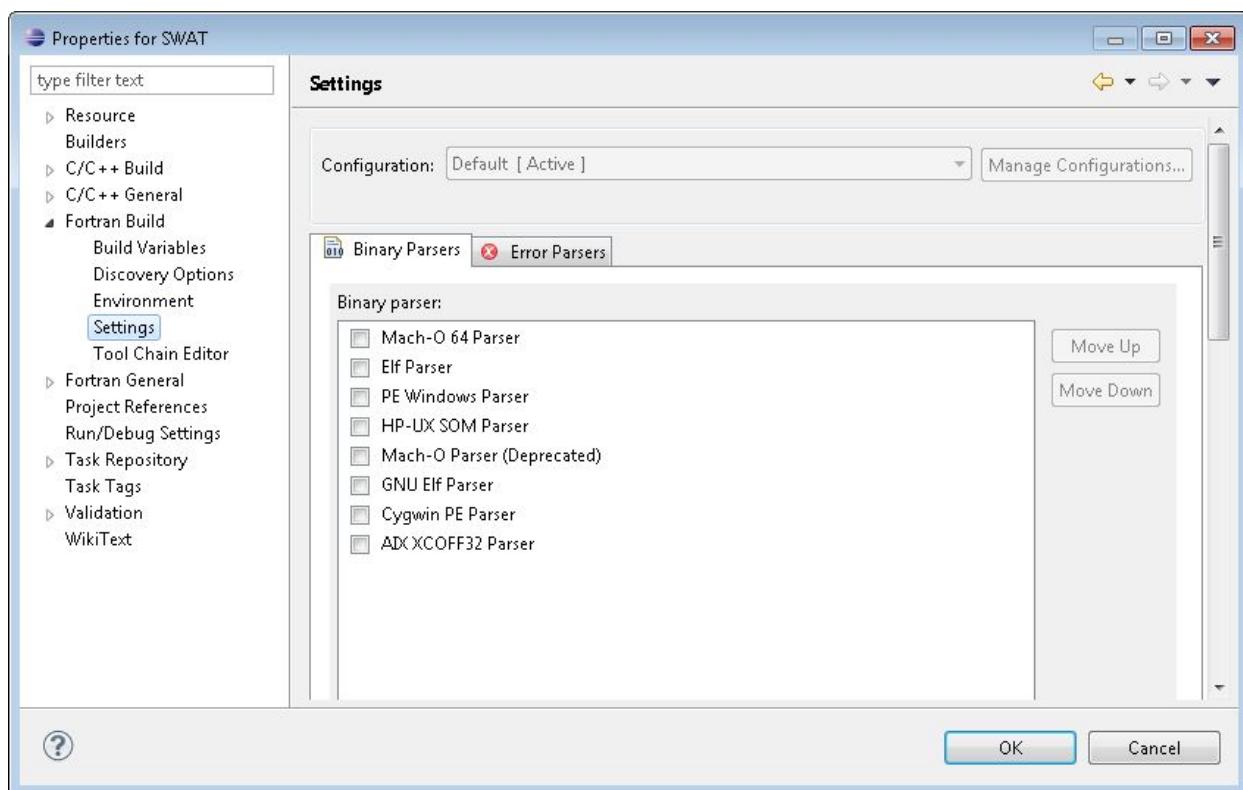


Note:

The location of gfortran and msys depends on which MinGW version would be used (Rtools, MinGW-w64 or MinGW).

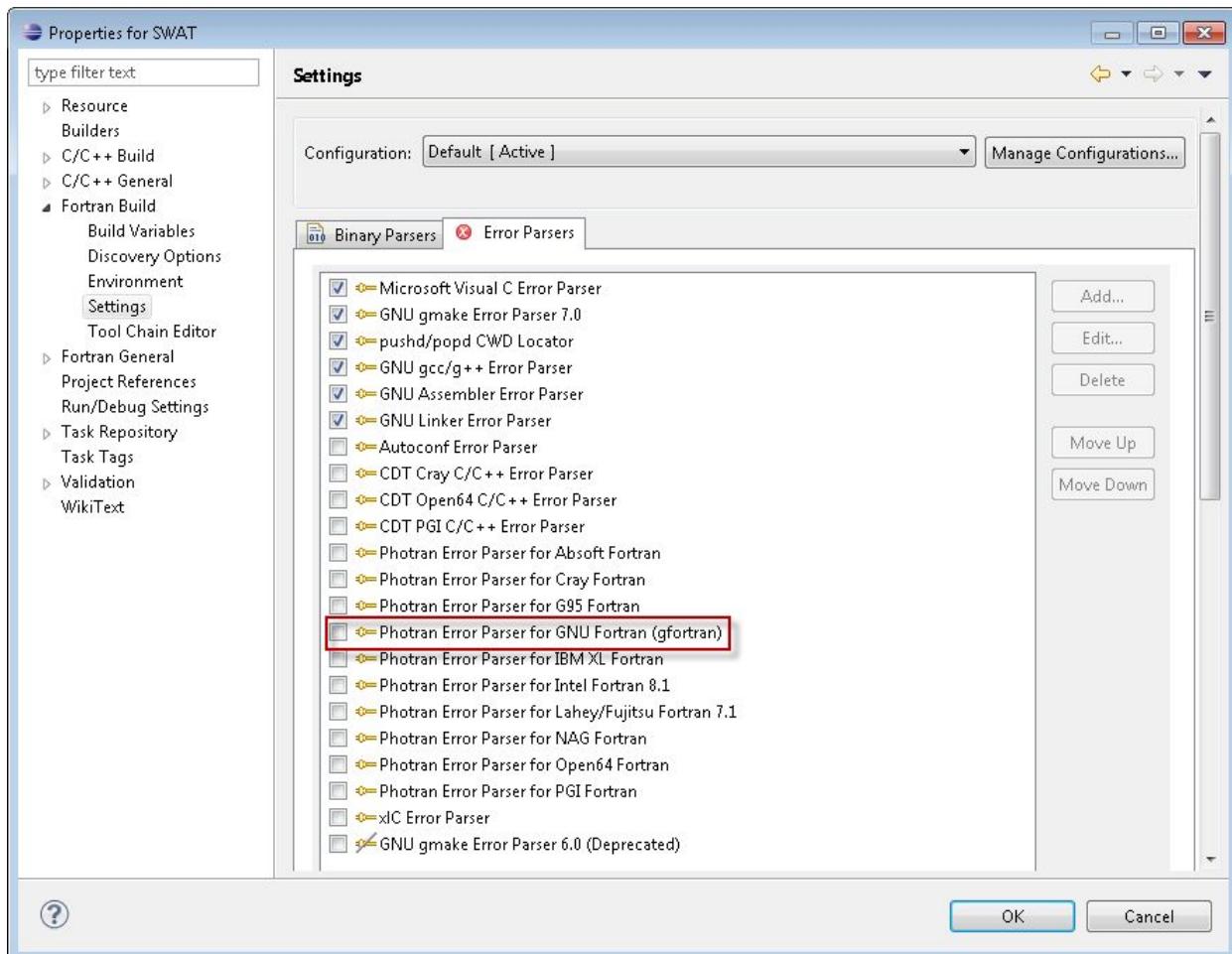
Binary Parser

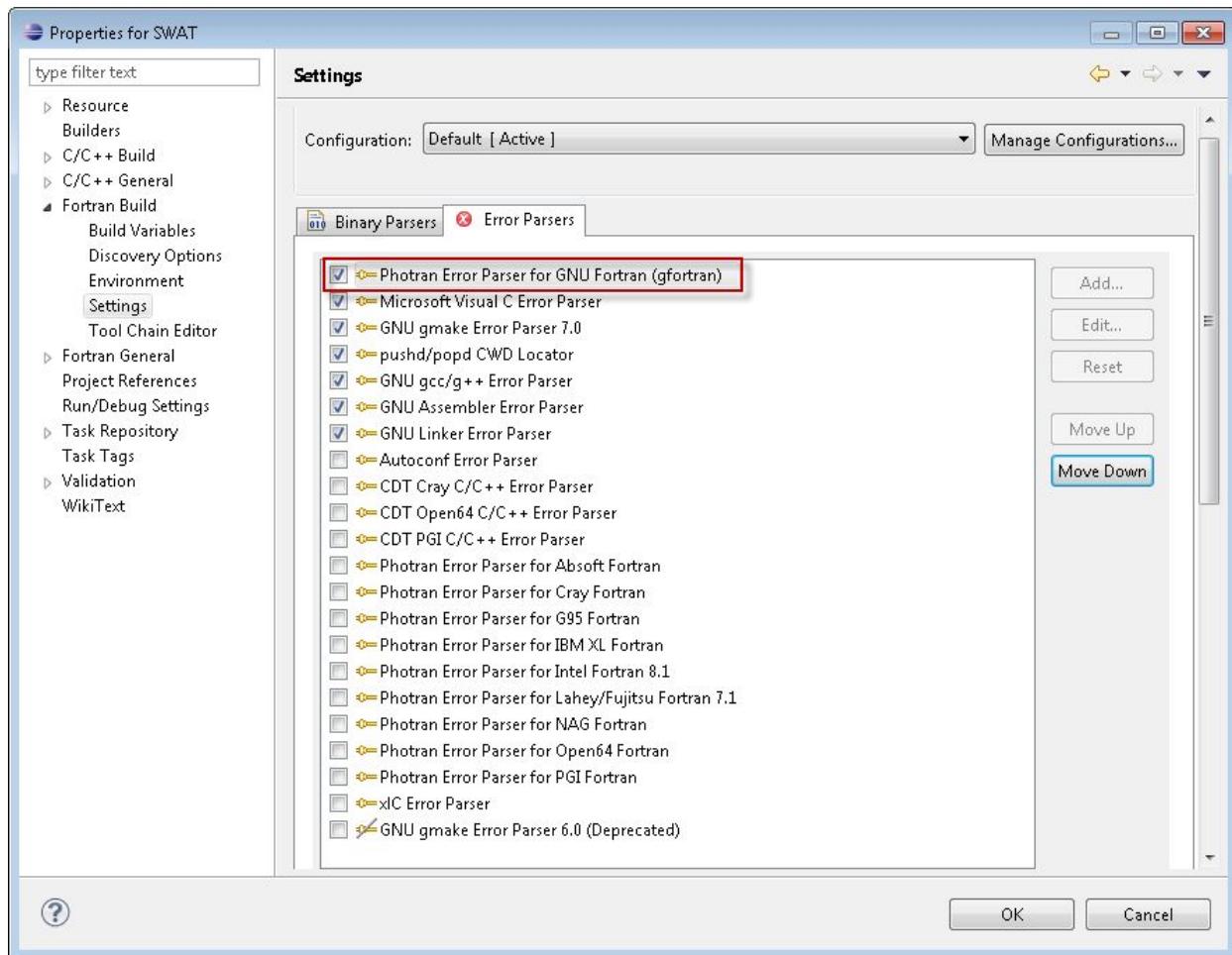
Select **Fortran Build -> Setting** in the left panel and then select **PE Windows Parser** for **Binary Parsers** in the right panel.



Error Parsers

Select **Photran Error Parser for GNU Fortran (gfortran)** for Error Parsers and move it to the top. This option is not selected when the project is created.

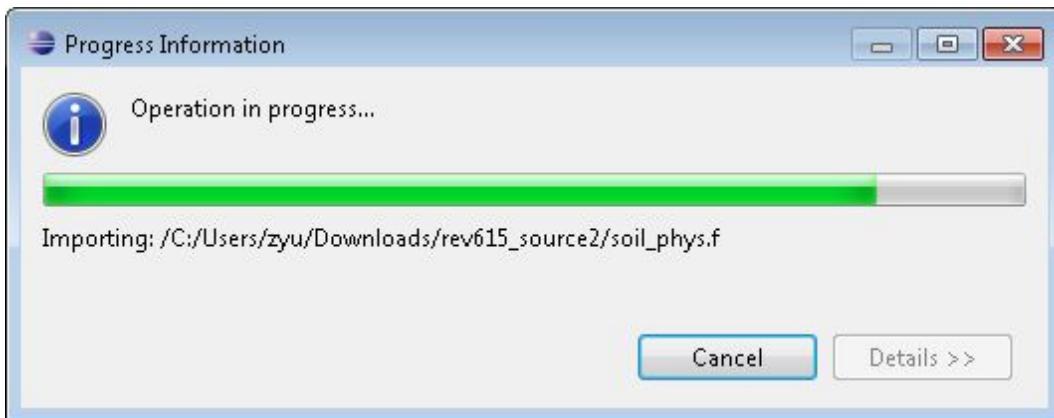
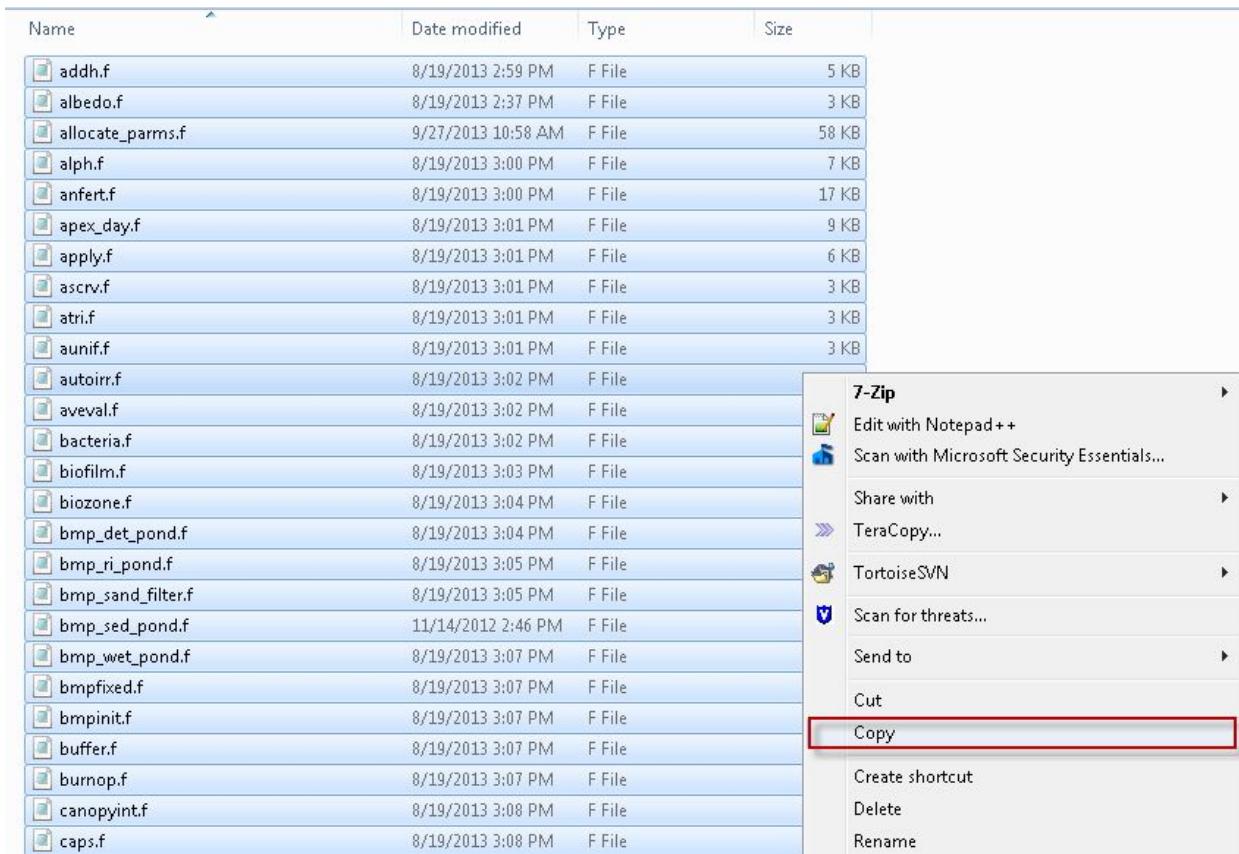




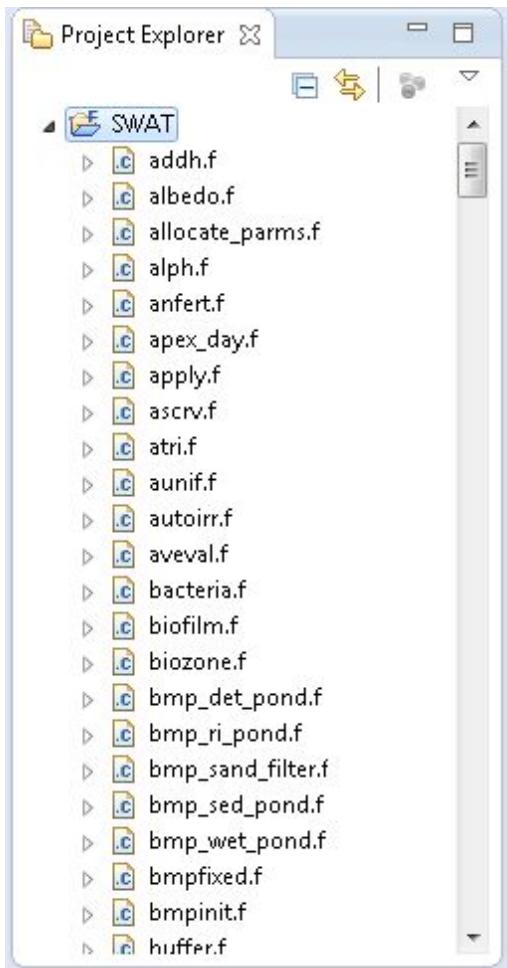
Click OK to close the properties window and save all the settings.

Copy SWAT Source Codes and Makefile

Copy SWAT source codes in Windows Explorer and right click the Fortran project and select paste to copy all the sources codes files into the project.



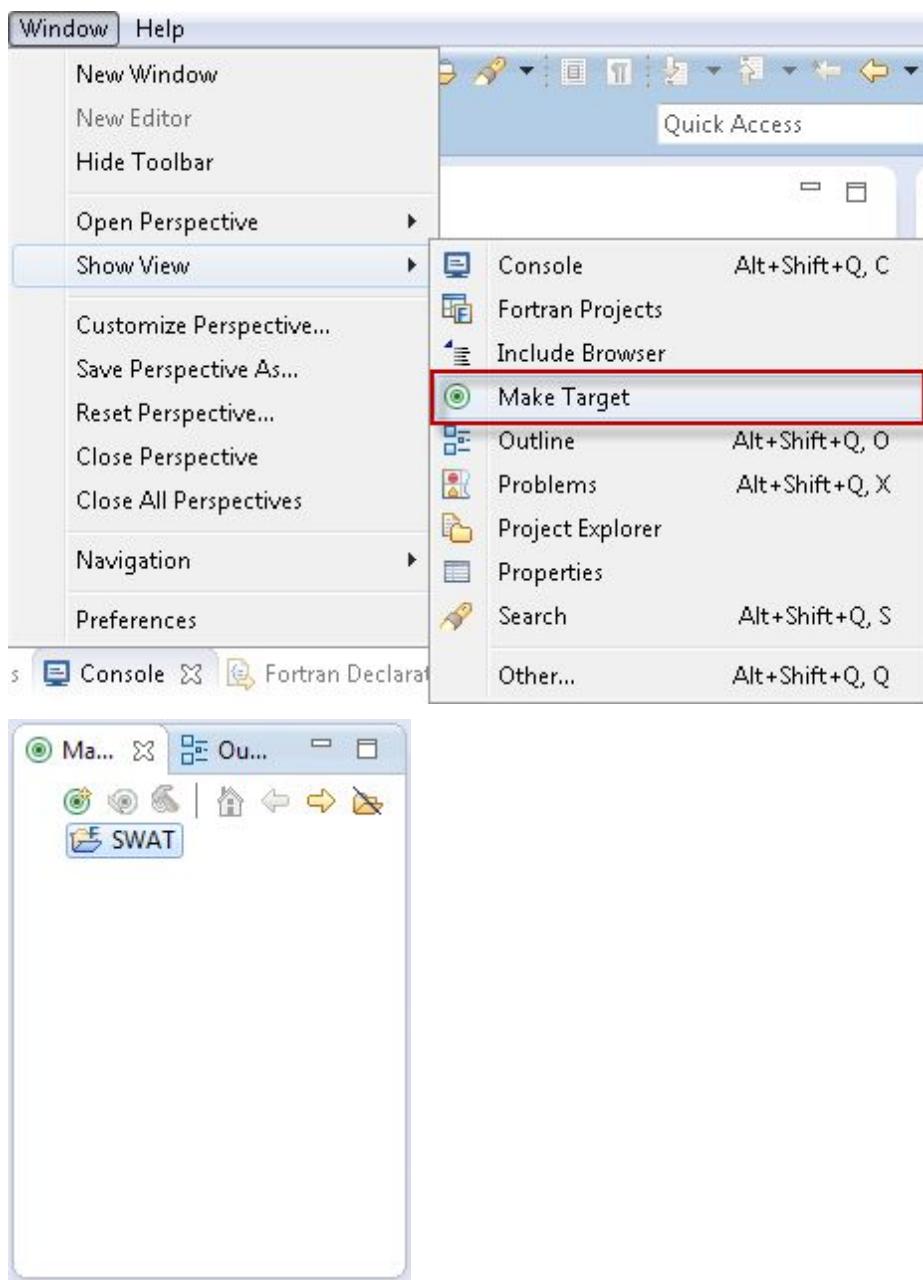
The source codes would be listed in the Project Explorer window.



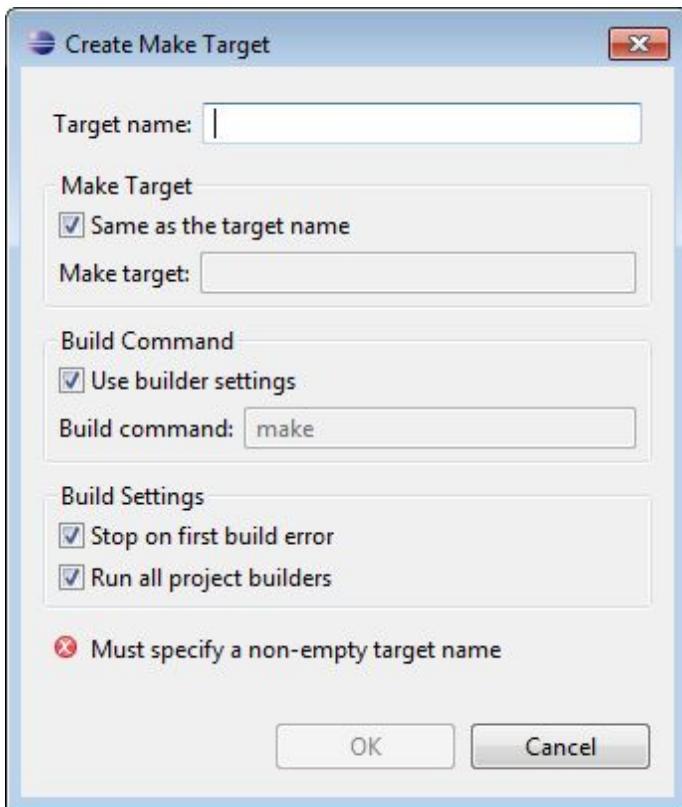
▷ zero_urbn.f
▷ zero0.f
▷ zero1.f
▷ zero2.f
▷ zeroini.f
[Makefile]

Create Make Target

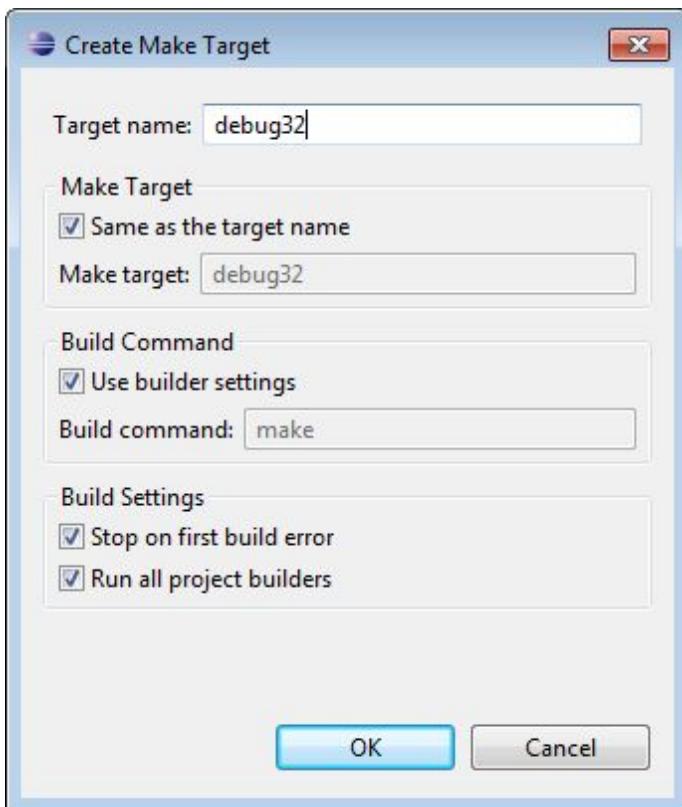
1. Open Make Target view through menu **Window -> Show View -> Make Target**.



2. Click to popup the Create Make Target window. Notice that here the build command is used the same build command set in project properties.



3. Input debug32 in the Target name text box and click OK.



4. The new Make target would be shown in Make Target window.



5. Repeat step 2-3 to create another three make targets using name **debug64**, **rel32** and **rel64**. Noticed that the target name is same as the parameter used in make command when SWAT is compiled in the command line. They are corresponding to different version of SWAT.



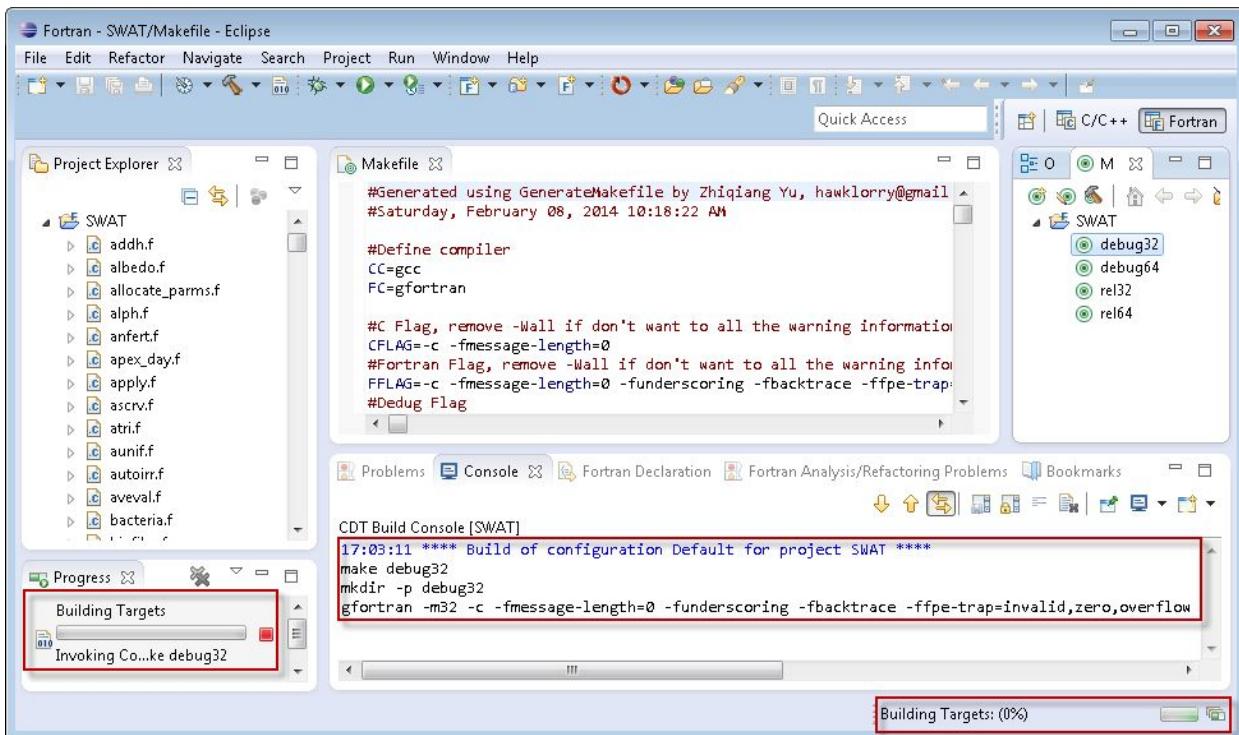
Build Make Target

Now, it's ready to compile SWAT, i.e. generate SWAT executable from source codes.

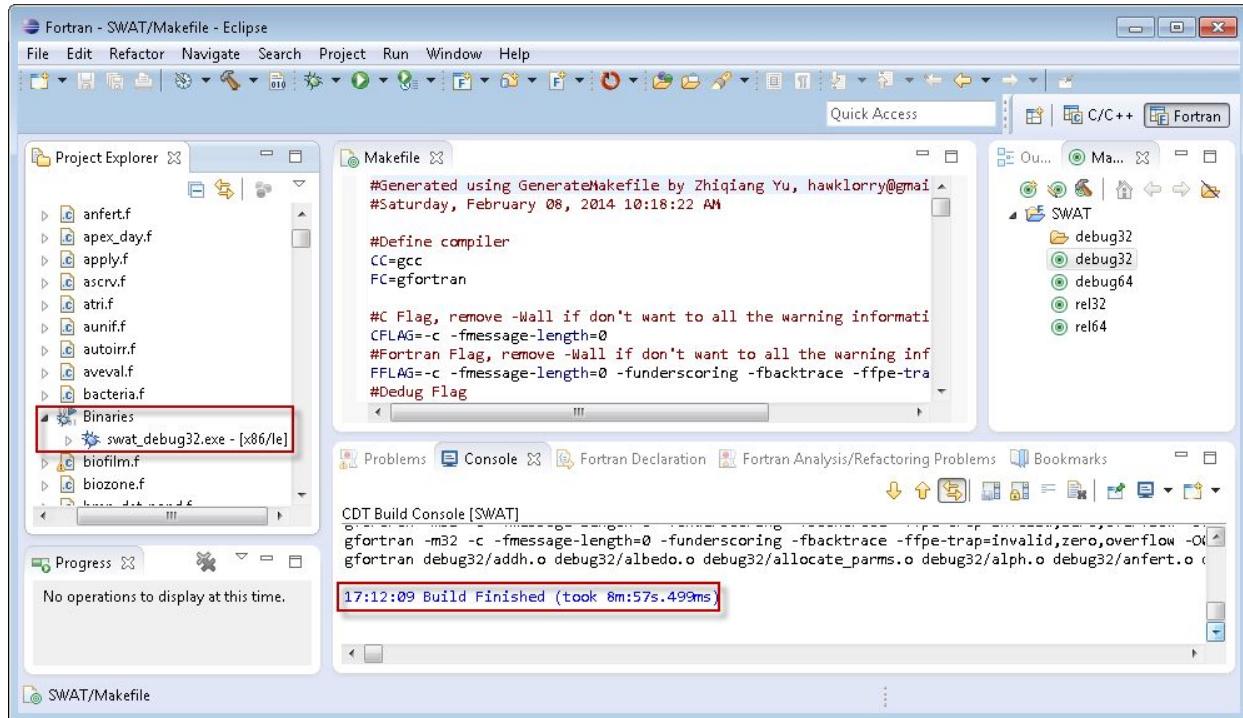
1. In Make Target view, double-click one of the four targets or select one of the make target in Make Target view, e.g. debug32, and then click to start building this target, i.e. compile SWAT for specific version.



2. The building process and possible warnings and errors would be shown in the **Console** window in the bottom.



3. When the build finished, the total time used would be shown in the Console window. Other three targets could be built using the same methods. At this point, the SWAT executables (swat_debug32.exe, swat_debug64.exe, rel32.exe and rel64.exe) are generated in the project folder, which will be shown in the **Project Explorer** view under the **Binaries** folder. It's ready to run run/debug SWAT.

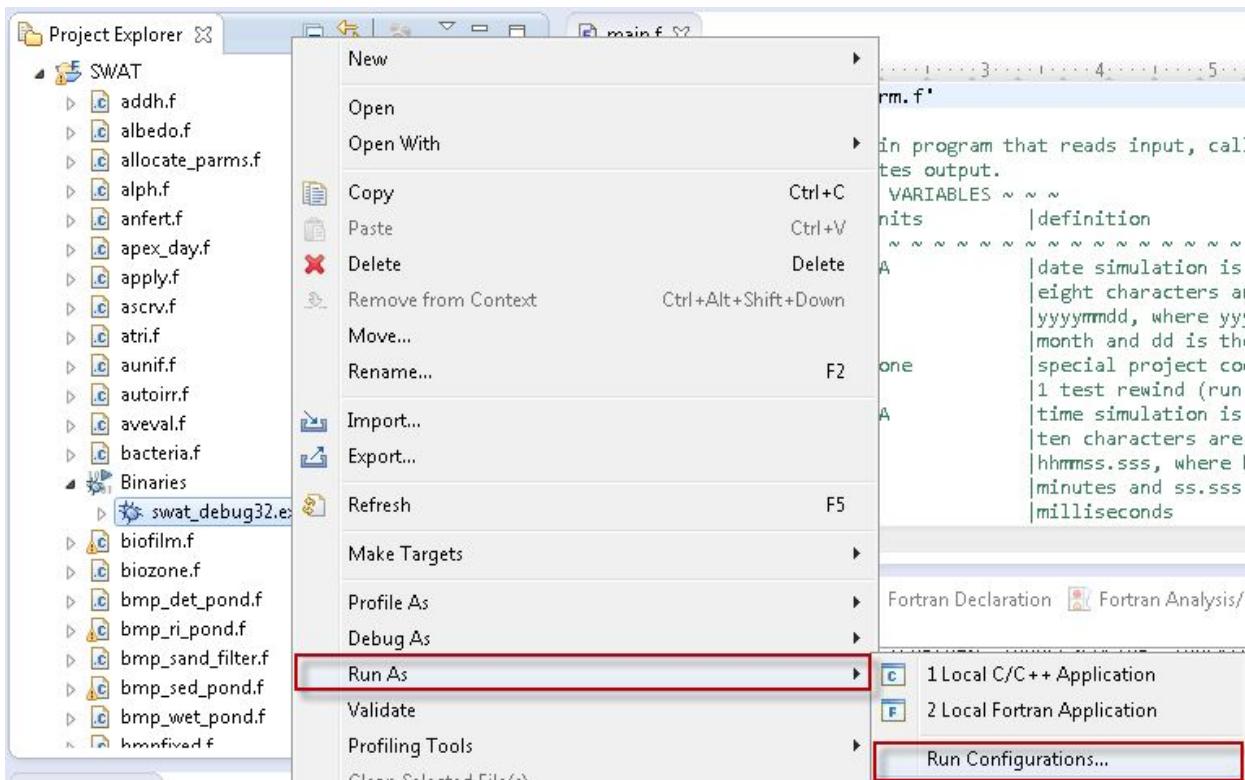


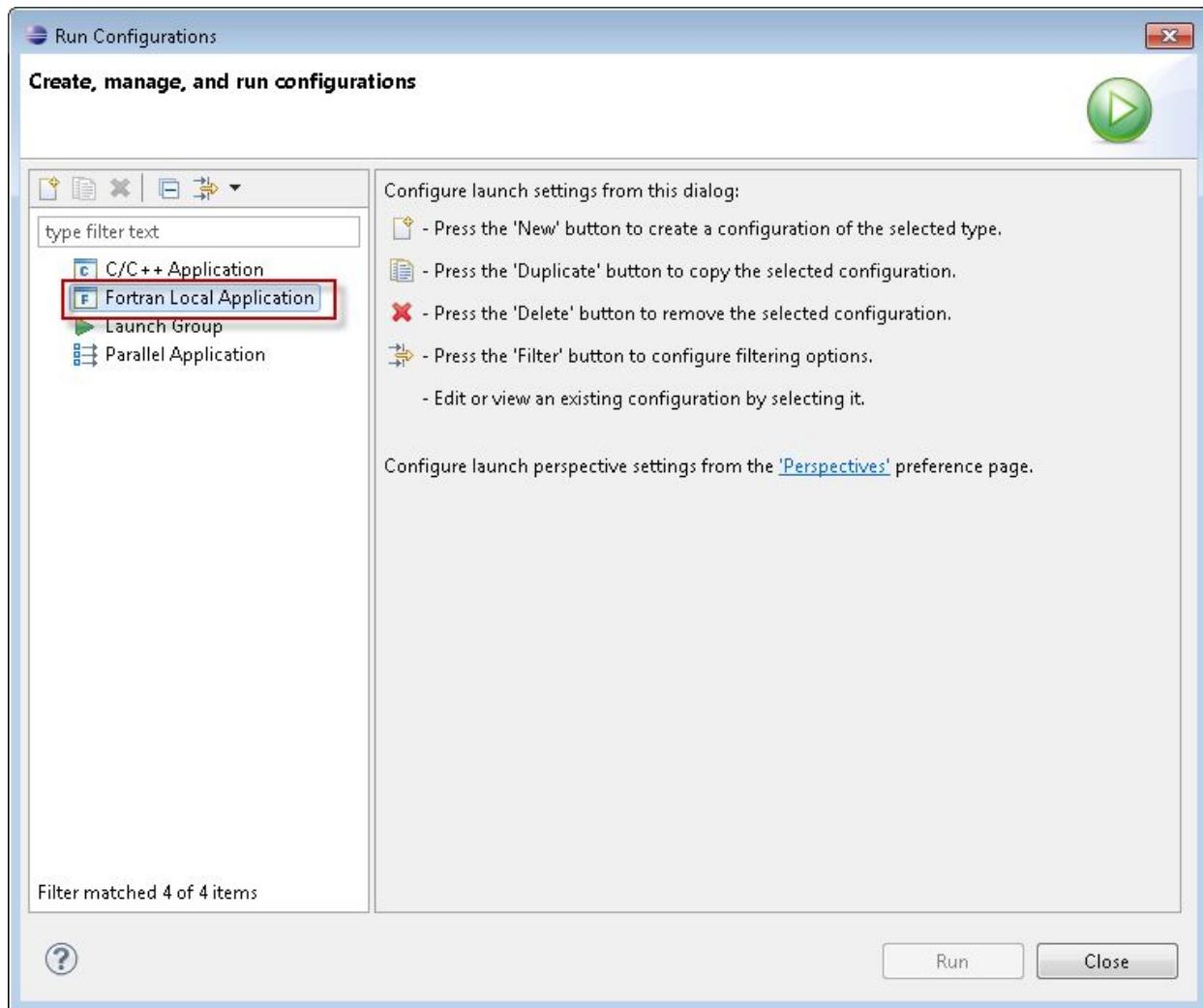
Debug/Run SWAT

Since the SWAT executables has been generated, it's time to run and debug SWAT. Some settings needs to be changed before SWAT could run successfully in Eclipse.

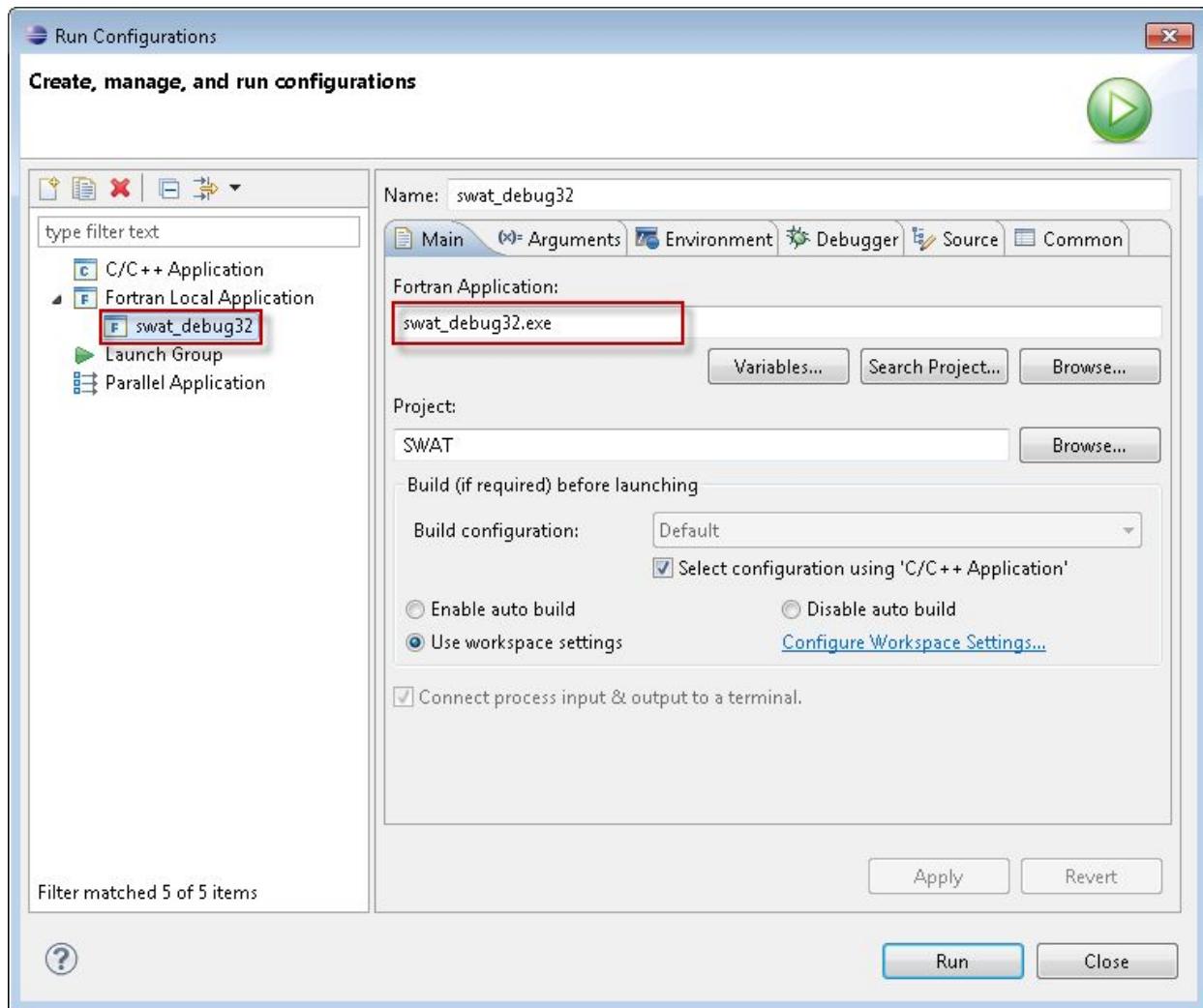
Create Run/Debug Settings

1. Select generated SWAT executable in the **Project Explorer** view under **binaries** folder, e.g. swat_debug32.exe. Right-click and select **Run As -> Run Configurations...** to open **Run Configurations** window. For debug, select **Debug As** instead of **Run As**.

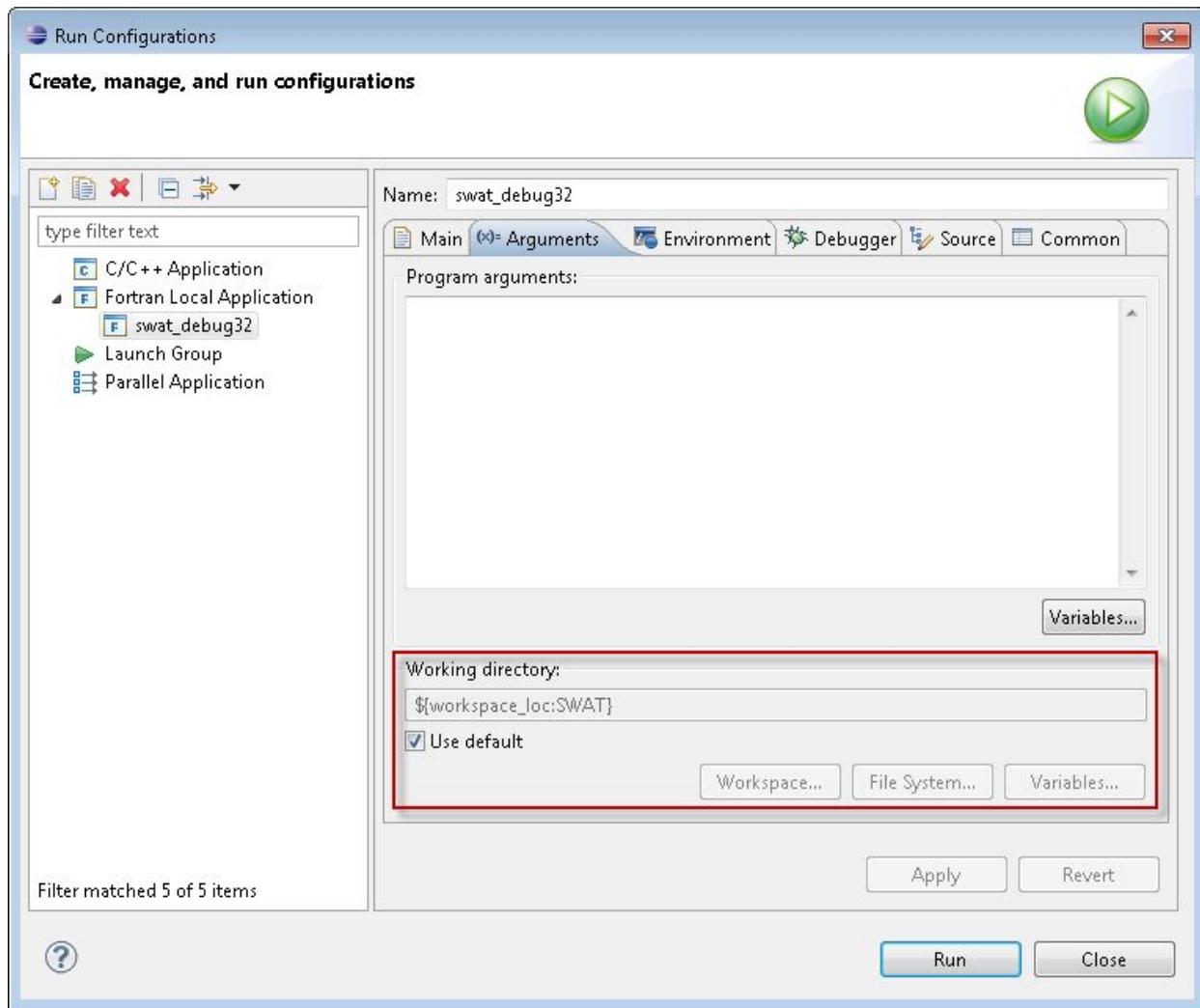


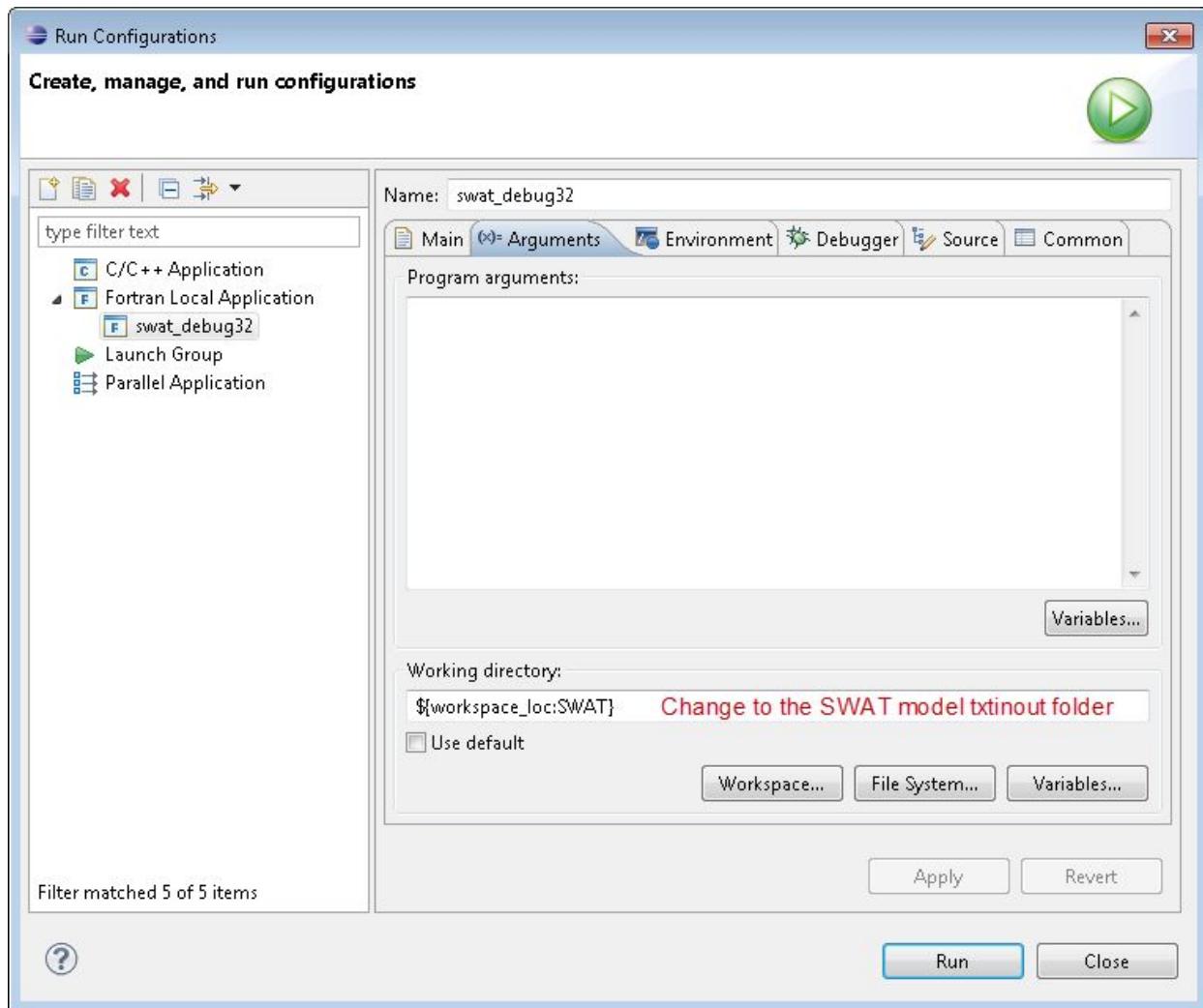


2. Select **Local Fortran Application** in the left panel and click to create a new configuration. The executable file name would be automatically set as the name of Fortran Applications.



3. Select **Arguments** tab in the right panel and change the Working Directory to the SWAT model folder, which is usually located in **Scenarios -> Default -> txtinout** folder in a ArcSWAT project.

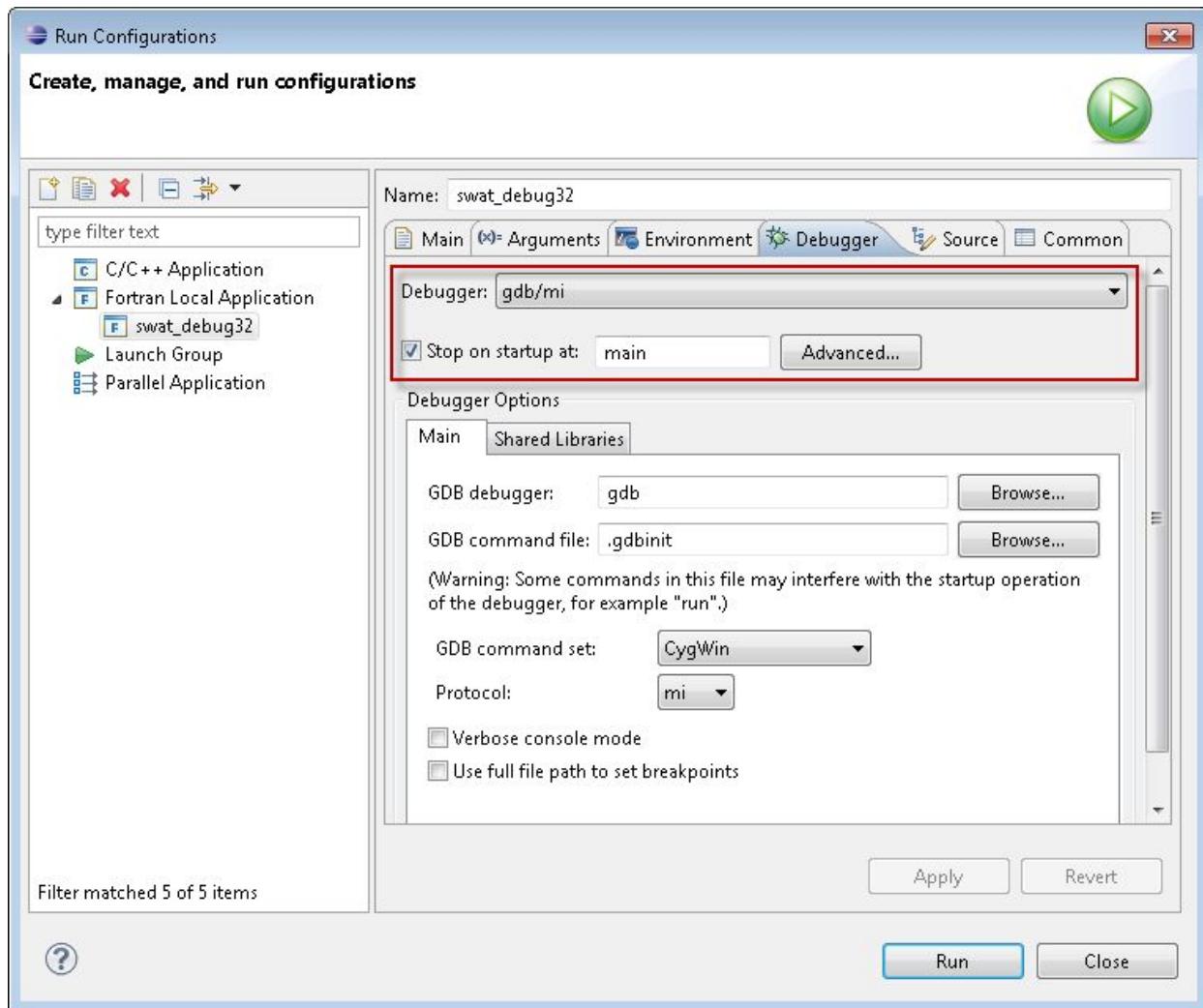


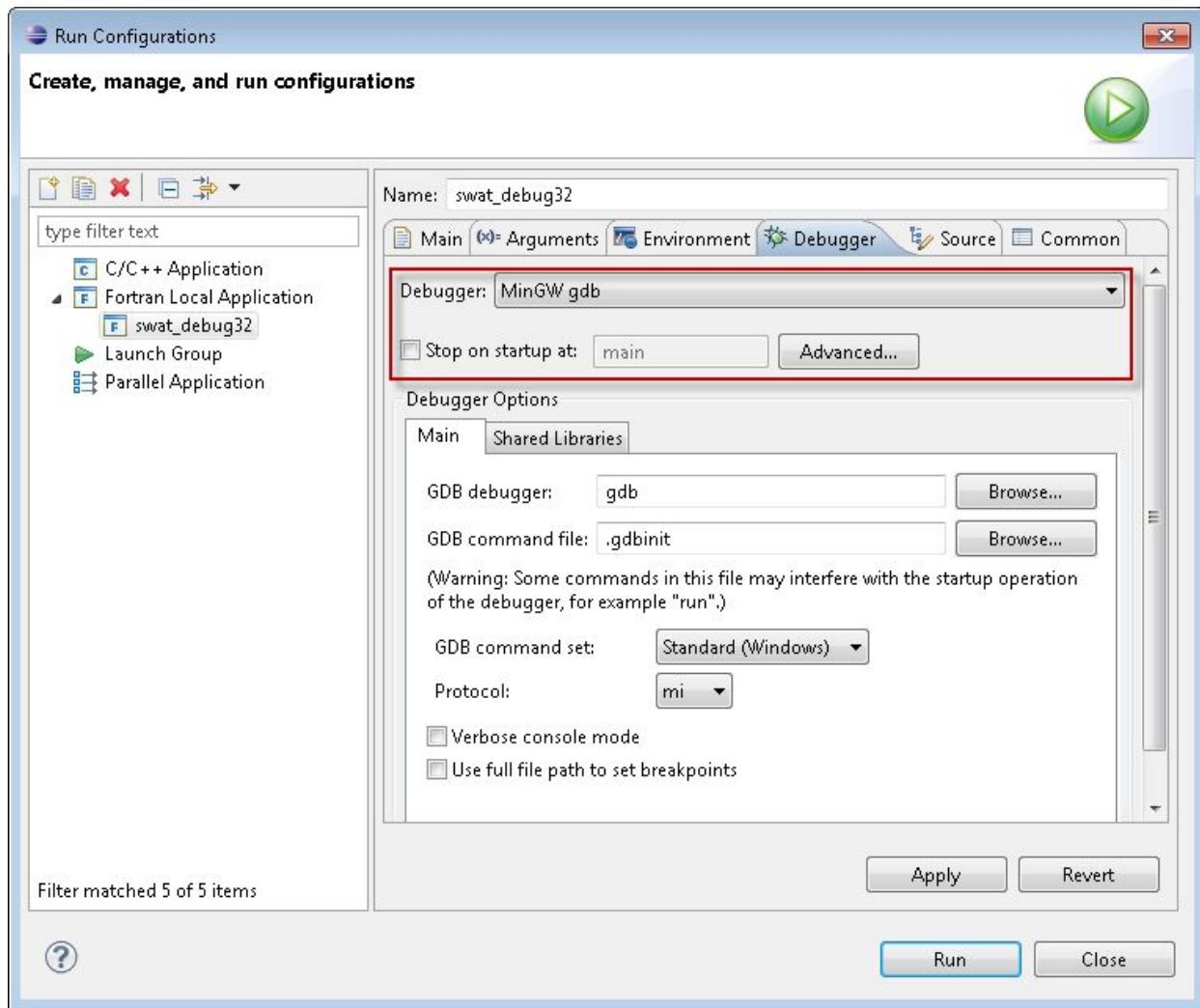


Note:

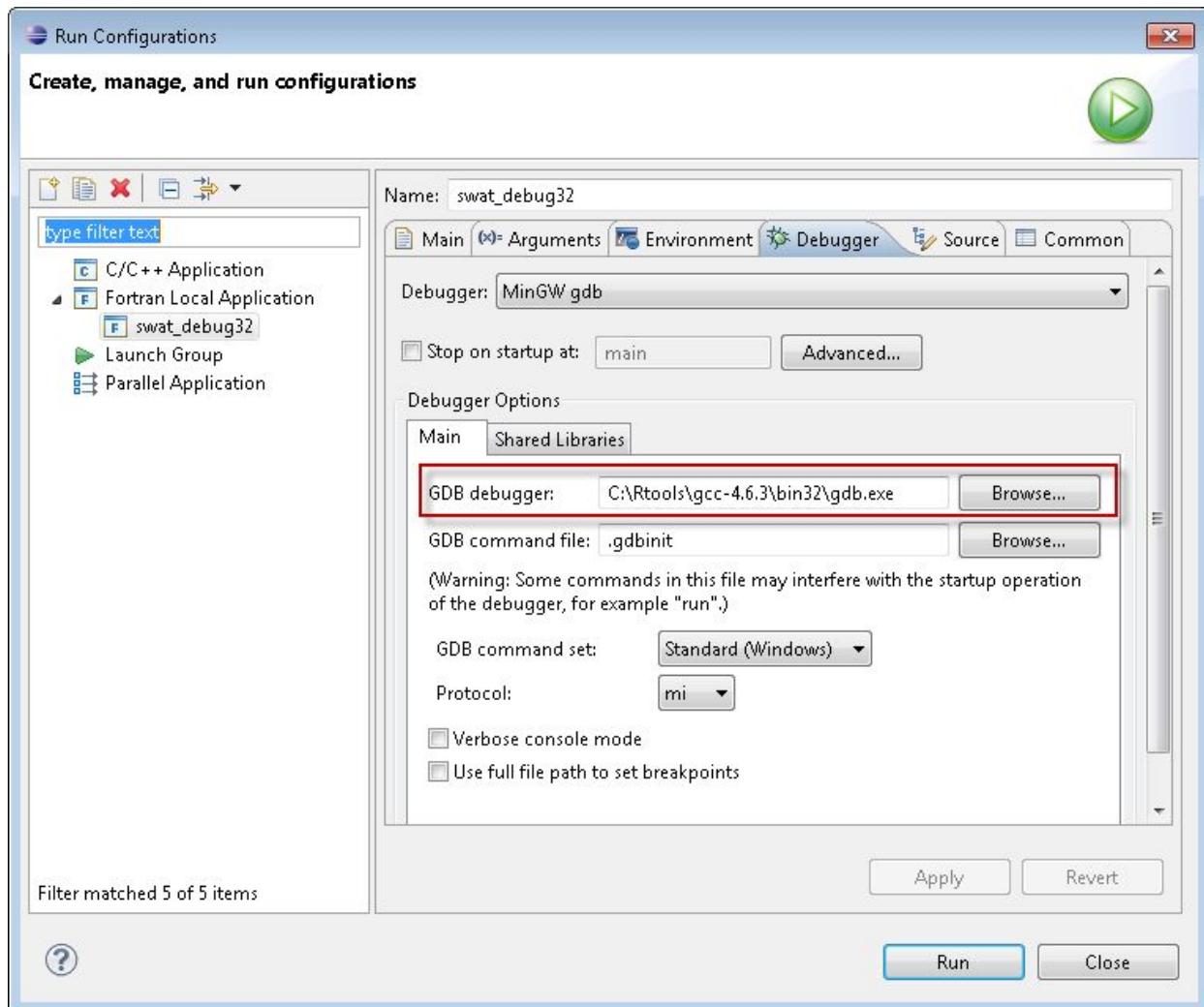
The working directory would be changed frequently if you are working on several models or scenarios.

4. Select **Debugger** tab in the right panel, change the Debugger to **MinGW gdb** and uncheck **Stop on startup at..**.





5. For Rtools, the gdb.exe is located at C:\Rtools\gcc-4.6.3\bin32 (for 32-bit) and C:\Rtools\gcc-4.6.3\bin64 (for 64-bit). These two folders are not added to the PATH variable by default. There is need to change the GDB debugger. Click the **Browse..** button to find gdb.exe and click OK.

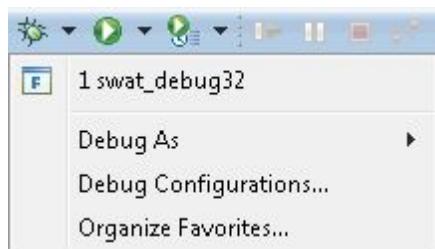


6. Click Apply button to save the configuration.
7. Repeat step 1-6 to create configurations for other three SWAT executables.

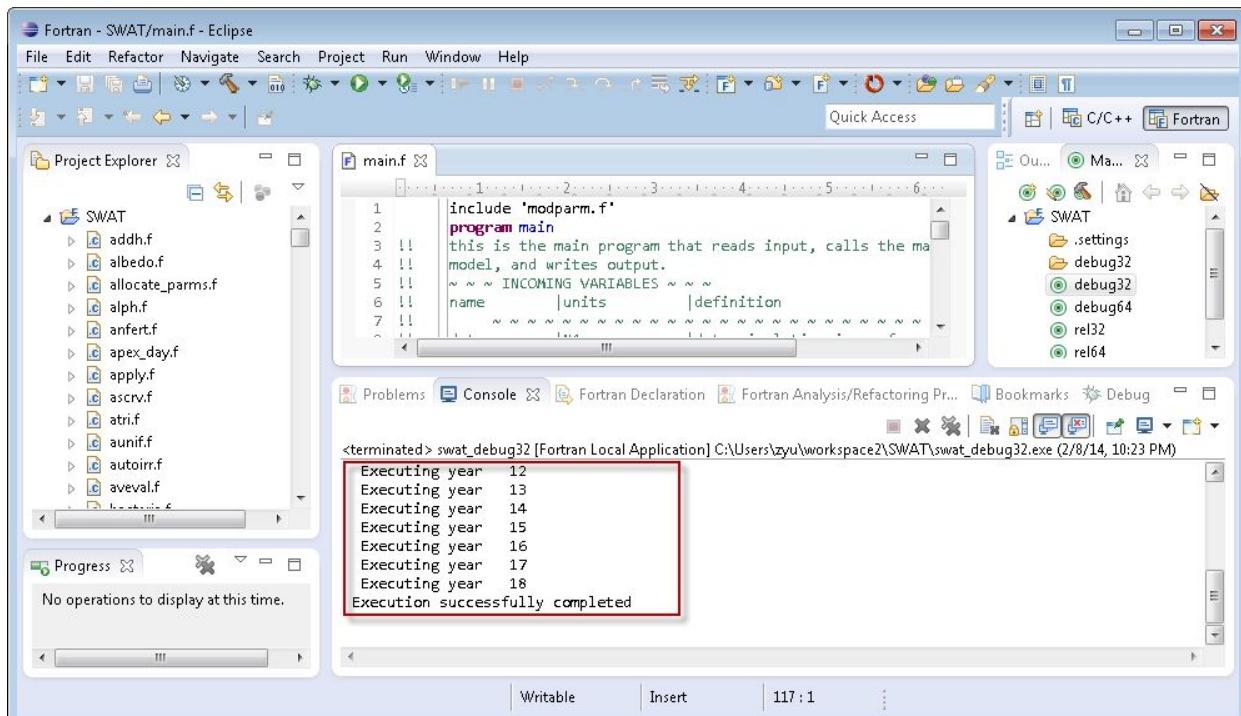
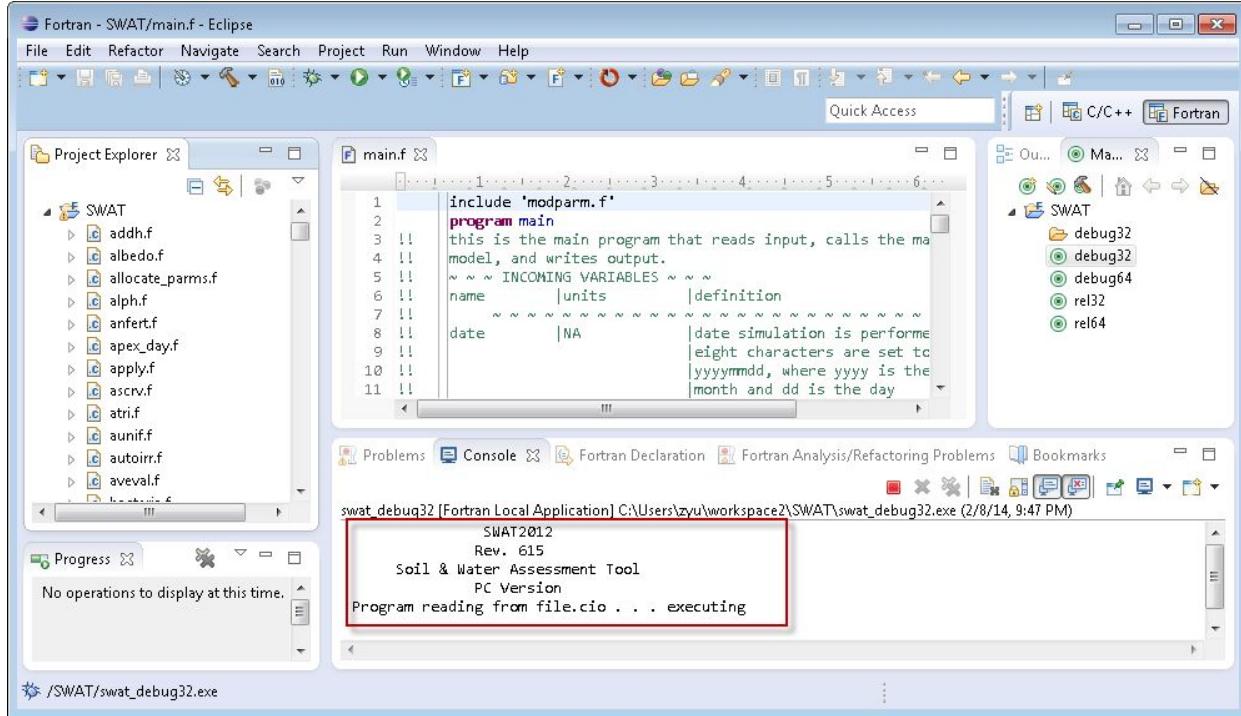
Debug/Run

After the configurations are created, SWAT could be run/debug though the toolbar

, where is debug and is run. When there are more than one executables, click the arrow after the button to select the executables.



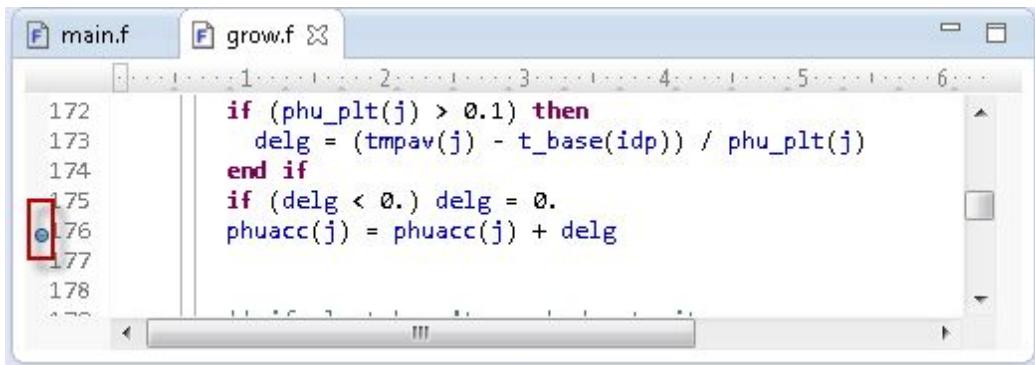
Click  or  start running or debugging SWAT. The regular SWAT output would be shown in the **Console** window.



Break Point

Breakpoints could be added in the editor by double-click the left edge of the line where the

breakpoint would be.

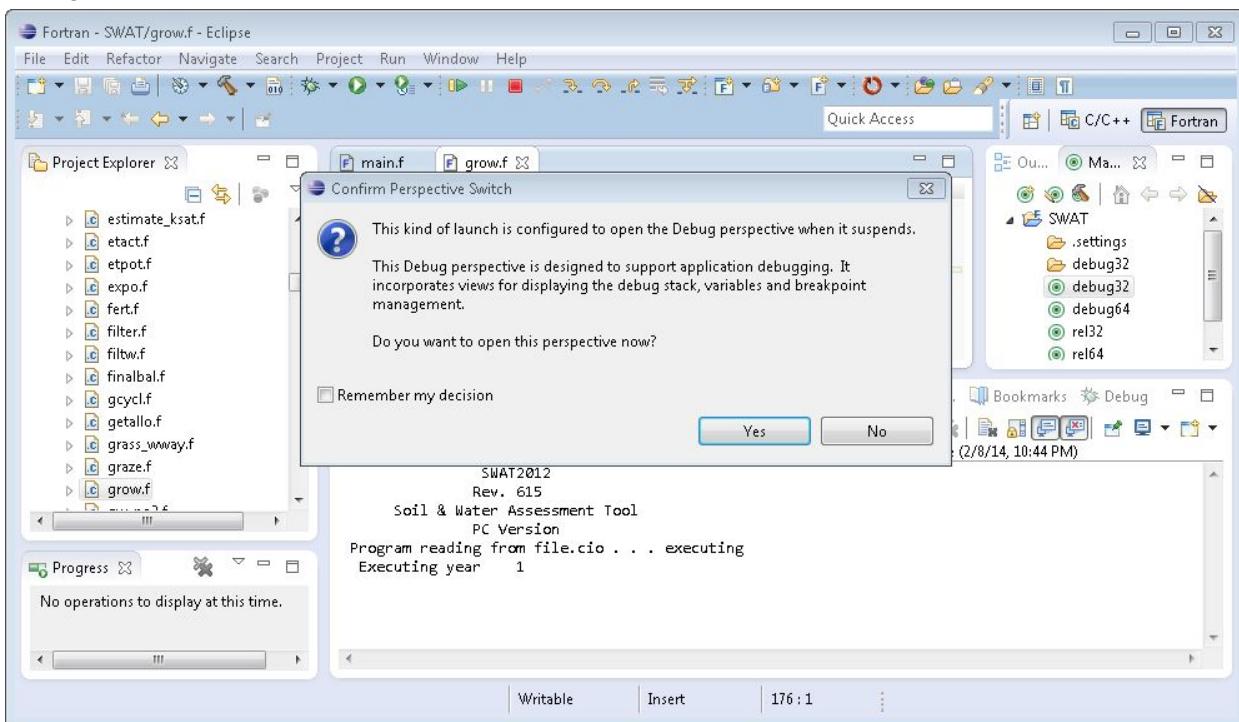


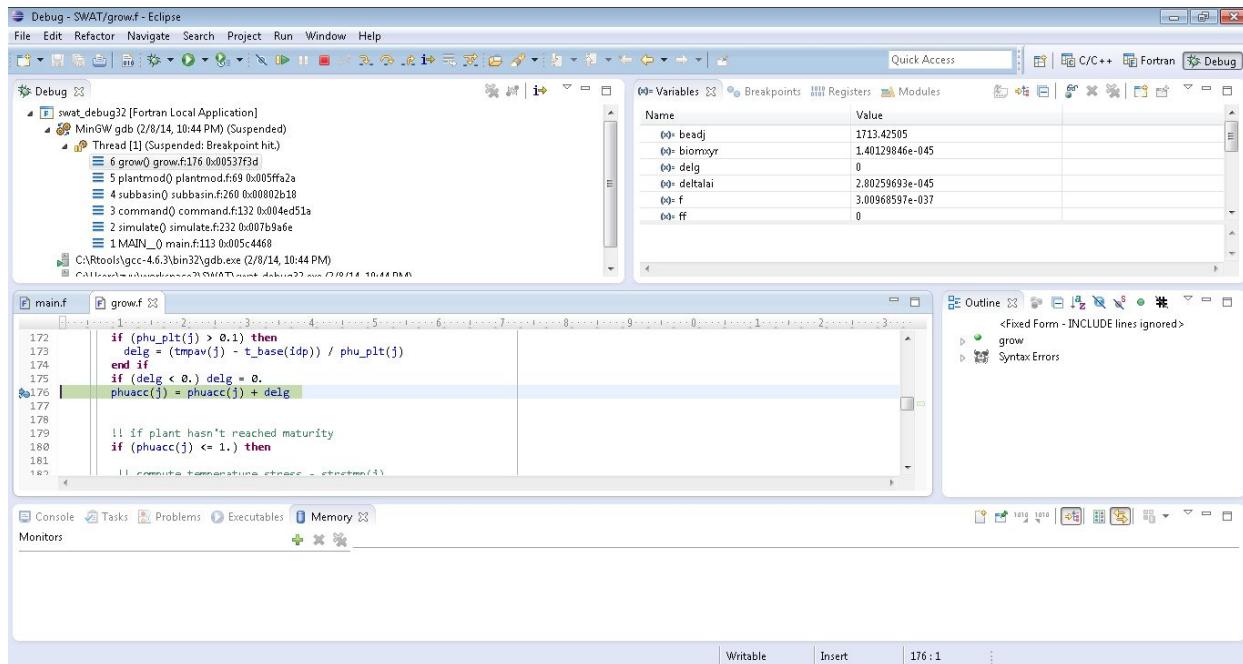
The screenshot shows the Eclipse IDE interface with two tabs open: "main.f" and "grow.f". The "grow.f" tab is active, displaying Fortran code. A red box highlights the line number 176, which contains a breakpoint indicator (a blue circle with a dot) followed by the code:

```
172 if (phu_plt(j) > 0.1) then
173     delg = (tmpav(j) - t_base(idp)) / phu_plt(j)
174 end if
175 if (delg < 0.) delg = 0.
176 phuacc(j) = phuacc(j) + delg
177
```

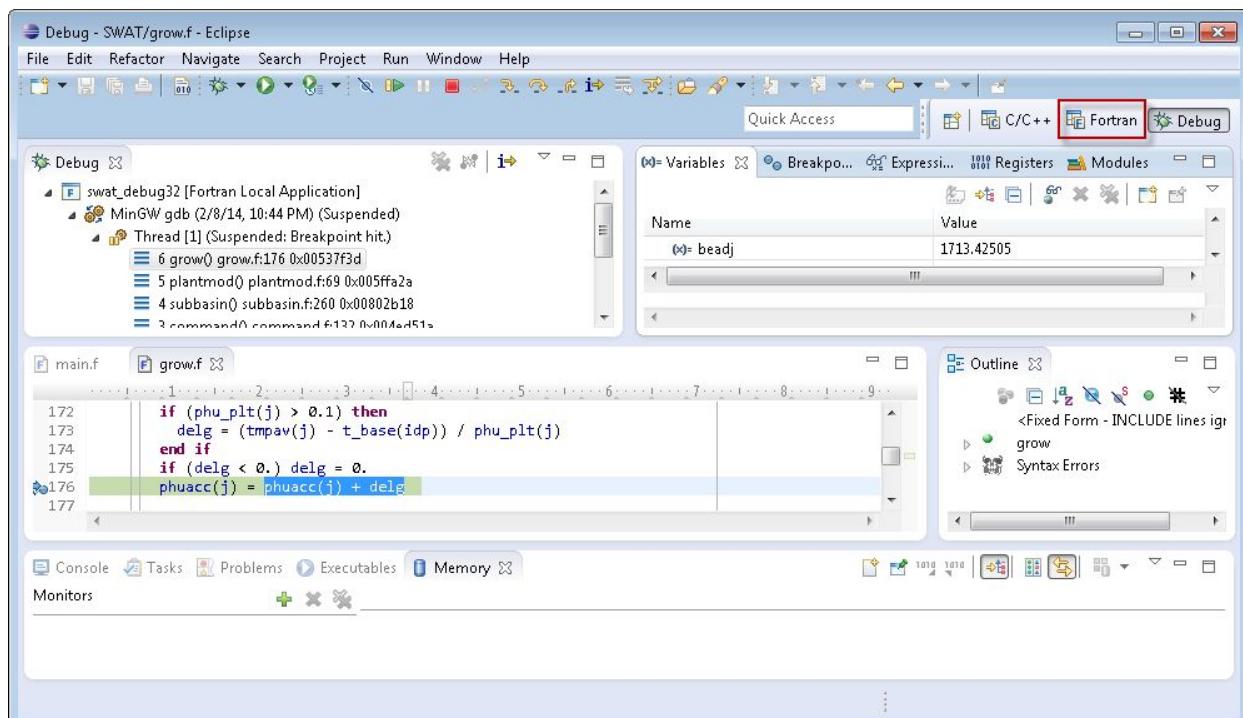
Debug Perspective

When breakpoint is hit, you would be asked if want to change to Debug Perspective, whose layout is designed to help debugging, especially the Variables and Expression view. It's recommended to change to this perspective and check Remember my decisions to go into debug perspective next time when a breakpoint is hit.

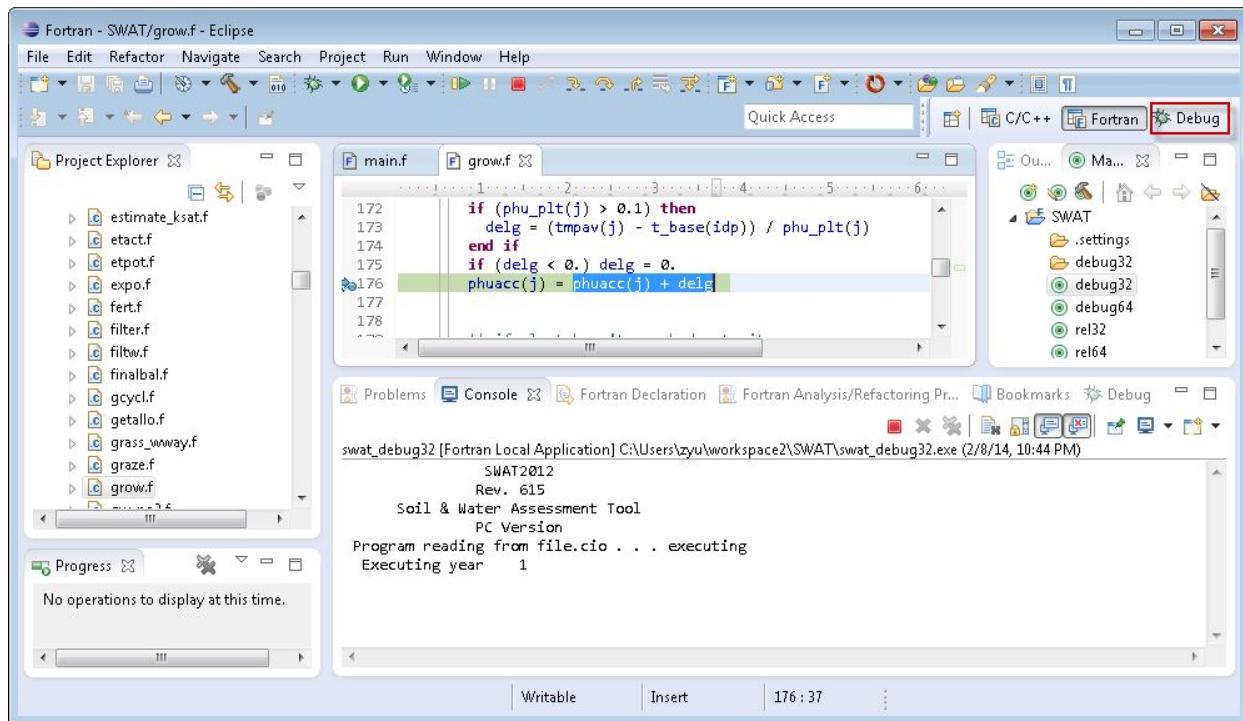




Click **Fortran** button on the up right corner, Eclipse would go back to Fortran perspective, which is designed for coding.



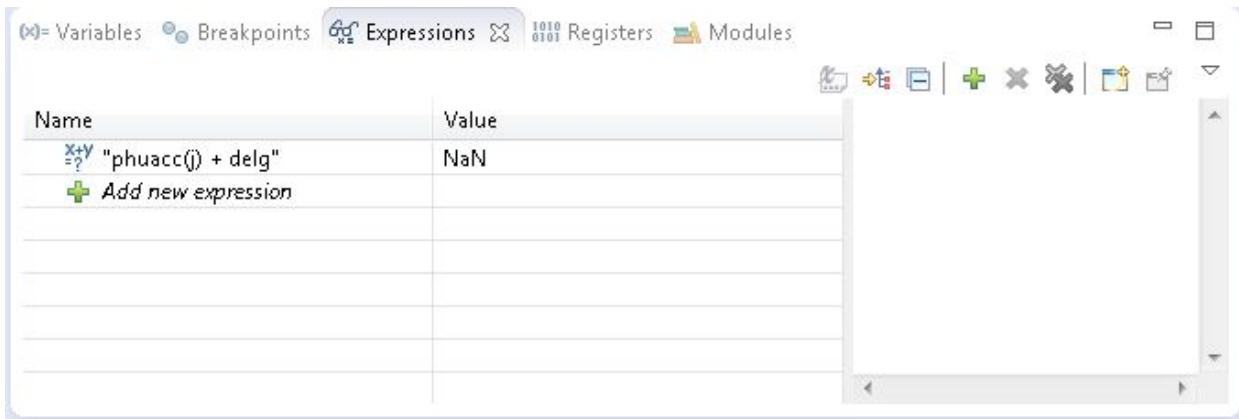
Click the **Debug** button in the up right corner to return to Debug perspective.



Variables and Expression View

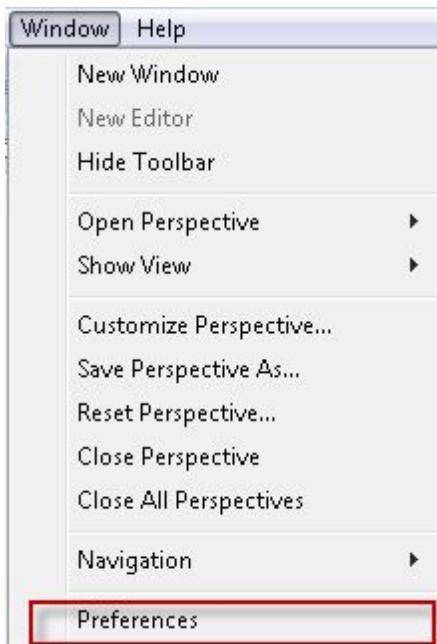
Variables view would show the value of local variables and Expression view would show values of any expression.

Variables		Breakpoints	Registers	Modules
Name	Value			
(x)= beadj	1713.42505			
(x)= biomxyr	1.40129846e-045			
(x)= delg	0			
(x)= deltalai	2.80259693e-045			
(x)= f	3.00968597e-037			
(x)= ff	0			



Some Useful Settings

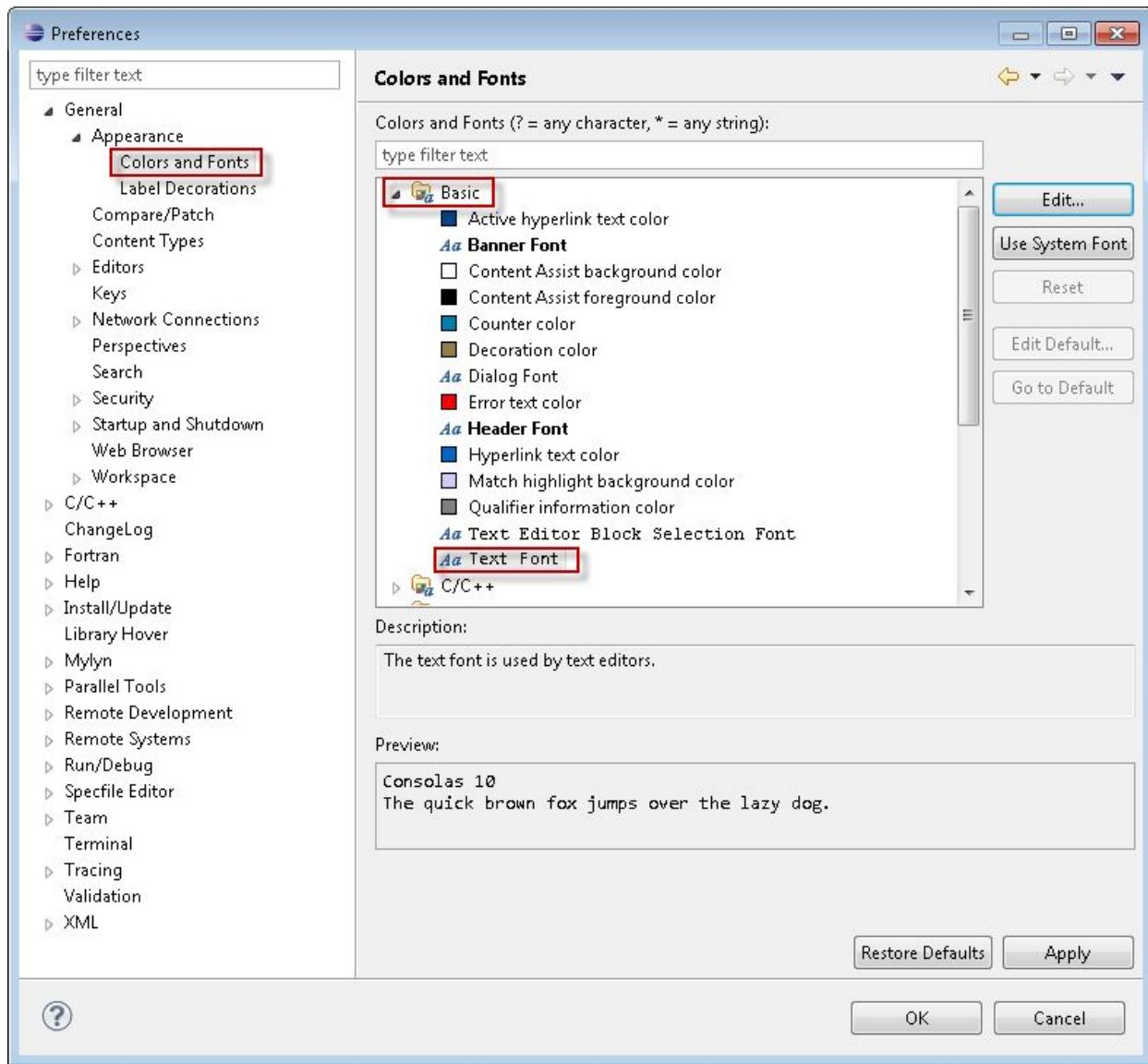
All these settings are located in Preferences window, which could be opened through menu Windows -> Preferences.



Change Text Font

The default text font is Consolas 10, which may be too small.

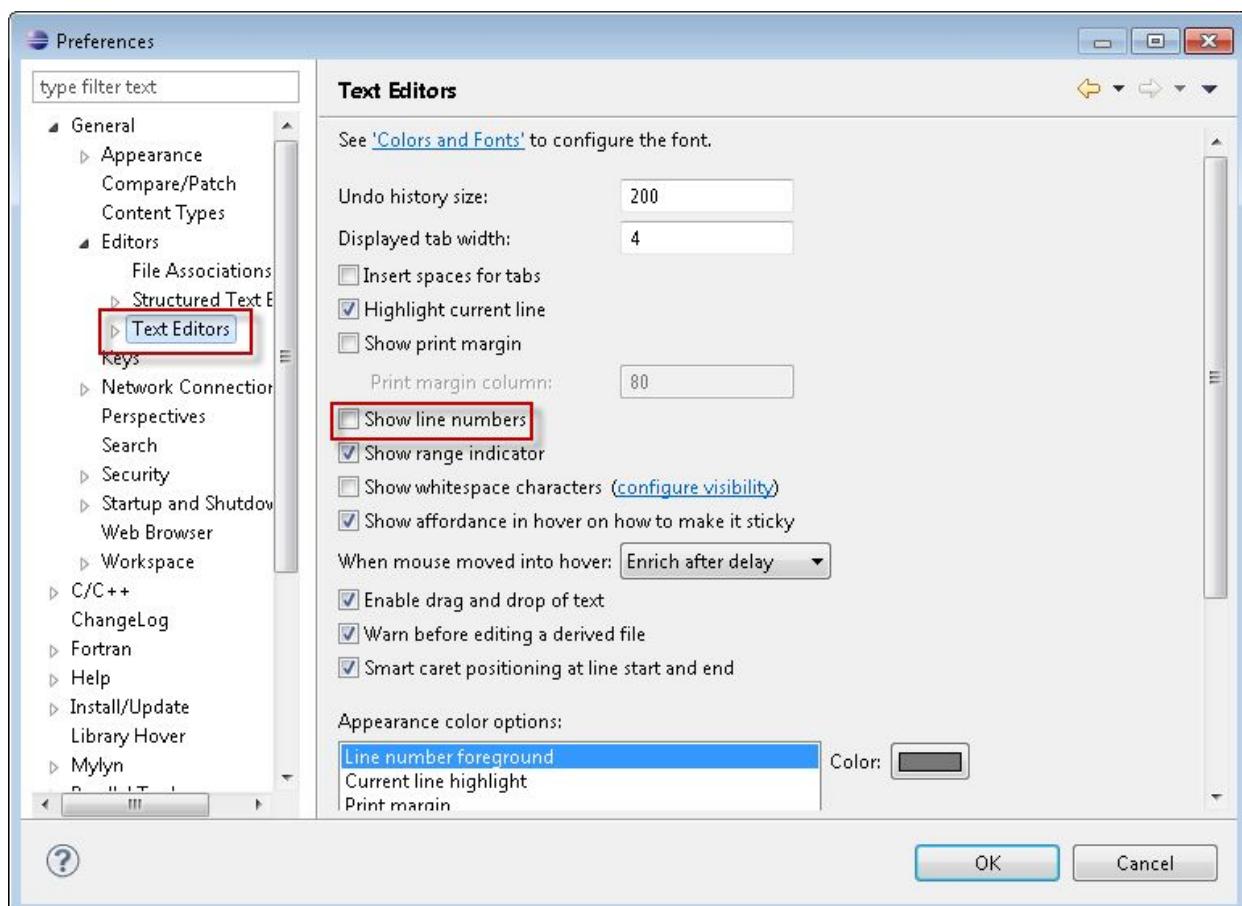
To change the font, select **General -> Appearance -> Colors and Fonts** in the left panel, and then select **Basic -> Text Font** in the right panel to edit the font.



Show Line Number

The line number is not shown in the editor by default. Showing line number would be useful for some cases.

To show line number, select **General -> Editors -> Text Editors** in the left panel and then check **show line number** in the right panel.

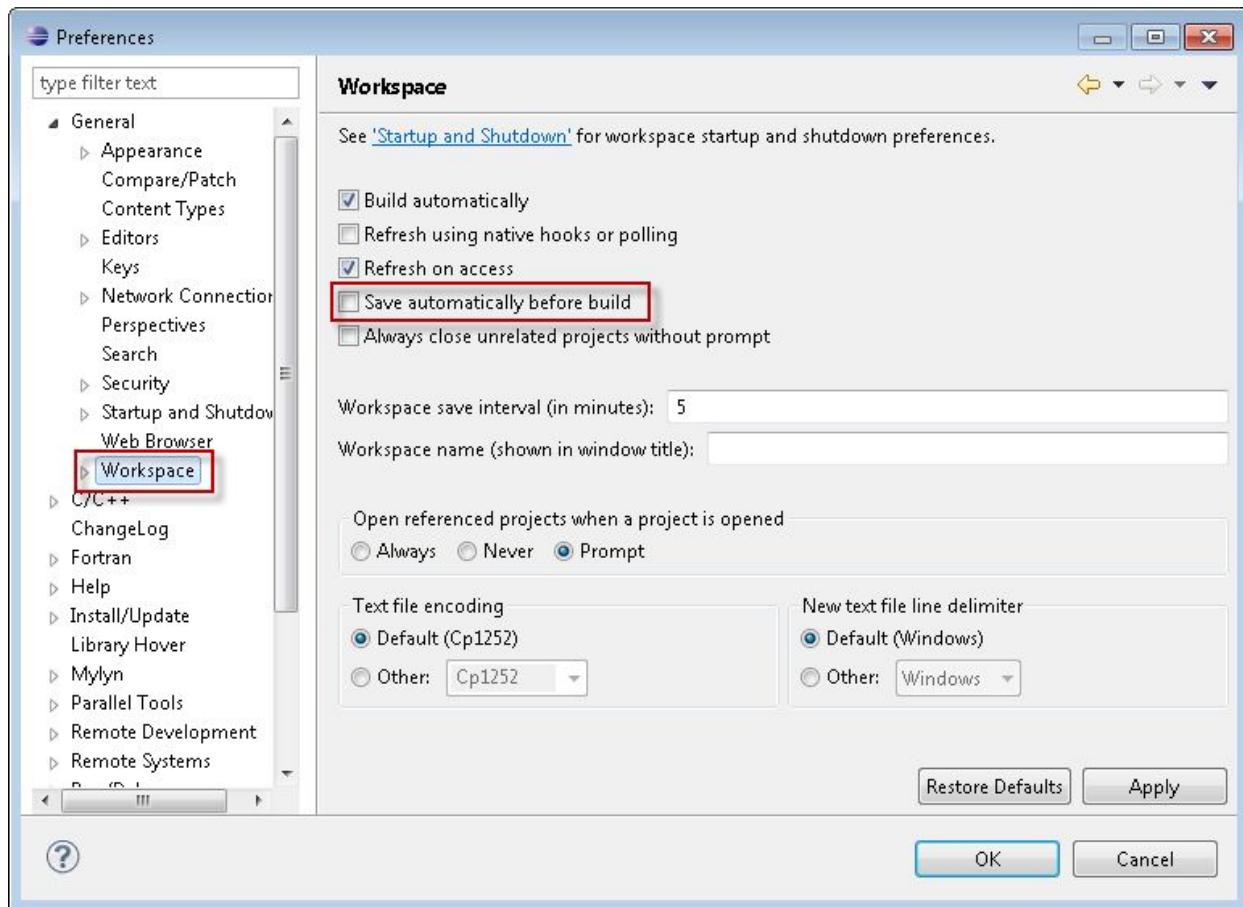


```
1 include 'modparm.f'
2 program main
3 !! this is the main program that reads input, calls the main simulation
4 !! model, and writes output.
5 !! ~ ~ ~ INCOMING VARIABLES ~ ~ ~
6 !! name      |units      |definition
7 !!          ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
8 !! date      |NA        |date simulation is performed where leftmost
9 !!          |           |eight characters are set to a value of
10 !!         |           |yyyyymmdd, where yyyy is the year, mm is the
11 !!         |           |month and dd is the day
12 !! isproj     |none      |special project code:
13 !!          |           |1 test rewind (run simulation twice)
14 !! time      |NA        |time simulation is performed where leftmost
15 !!          |           |ten characters are set to a value of
16 !!         |           |hhmmss.sss, where hh is the hour, mm is the
17 !!         |           |minutes and ss.sss is the seconds and
18 !!         |           |milliseconds
```

Save automatically before build

By default, the edit is not save automatically when the project is built. You need to click the save or save all button to make the edit works, which is not very convenient.

To enable automatically saving, click **General -> Workspace** in the left panel and check **Save automatically before build** in the right panel.



Change Keys

The default keys in Eclipse is different from Visual Studio. For a long time Visual Studio user, it would be great to use the same keys. It's located in **General -> Keys**. Select **MicroSoft Visual Studio** in the Schema list to change the keys.

