

Building ION on Windows for use with Visual Studio

Jeff Lippincott

9 September 2014

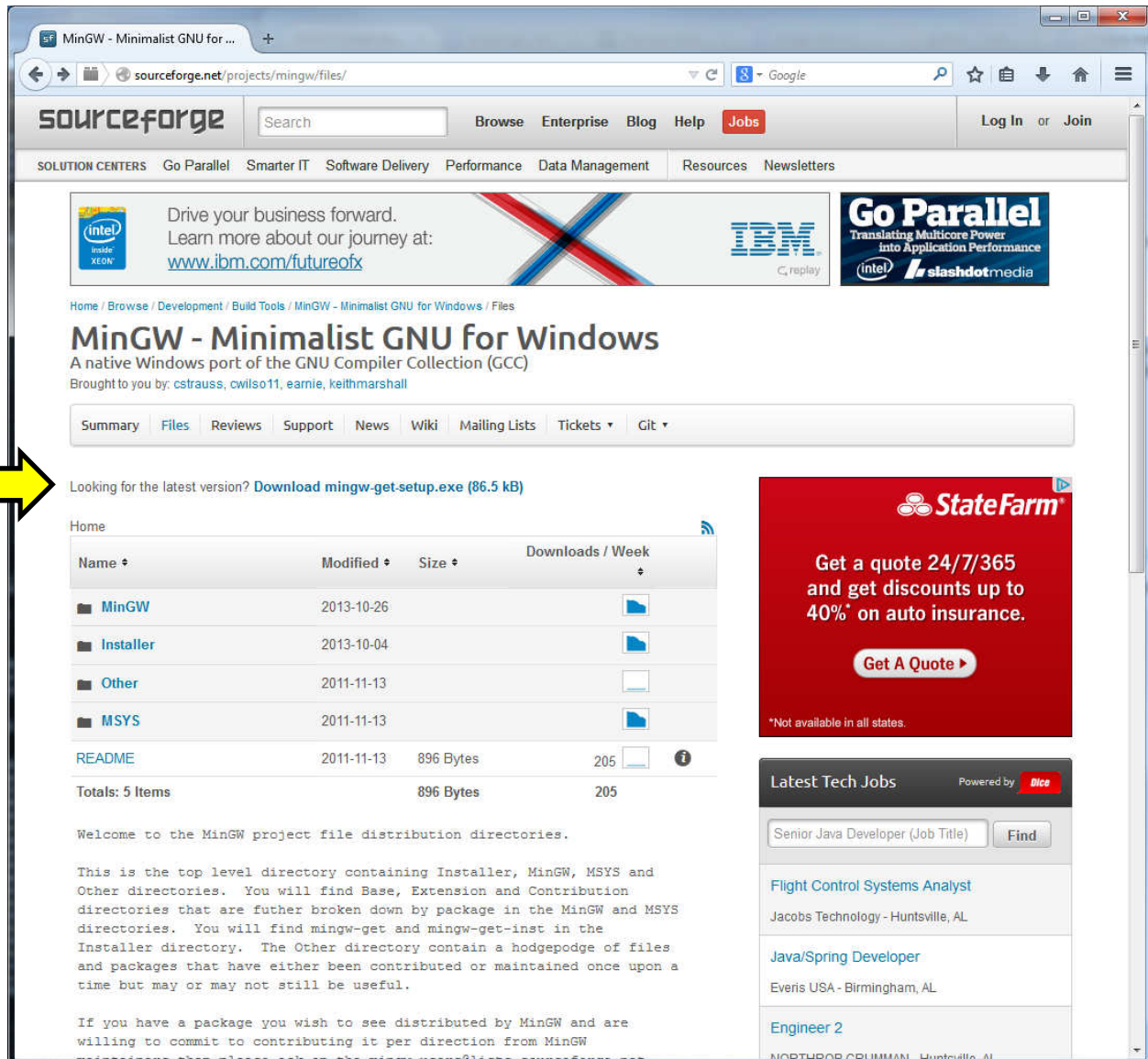
These instructions are for ION version 3.2.2. Later versions should work the same with substitution of the correct version number in the instructions.

MinGW

MinGW must be loaded and properly configured. There is a download link on the main MinGW page <http://www.mingw.org> which should take you to here:

<http://sourceforge.net/projects/mingw/files/>

You will want to download the graphical installer. There should be a link on the main page as shown below (mingw-get-setup.exe):

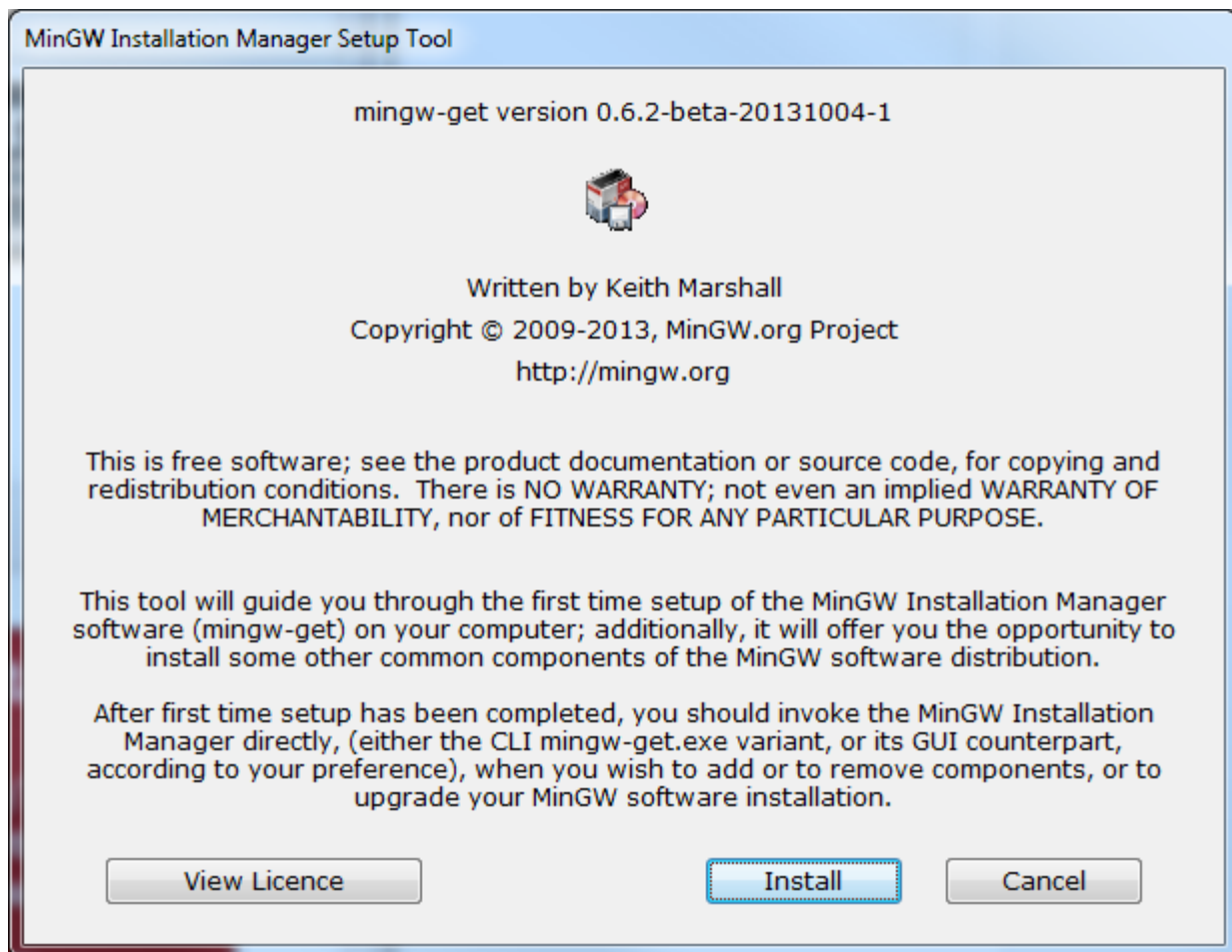


The screenshot shows the SourceForge project page for MinGW. The page title is "MinGW - Minimalist GNU for Windows". Below the title, it says "A native Windows port of the GNU Compiler Collection (GCC)". The page has a navigation bar with links: Summary, Files, Reviews, Support, News, Wiki, Mailing Lists, Tickets, and Git. A yellow arrow points to the link "Download mingw-get-setup.exe (86.5 kB)". Below this link is a table listing files:

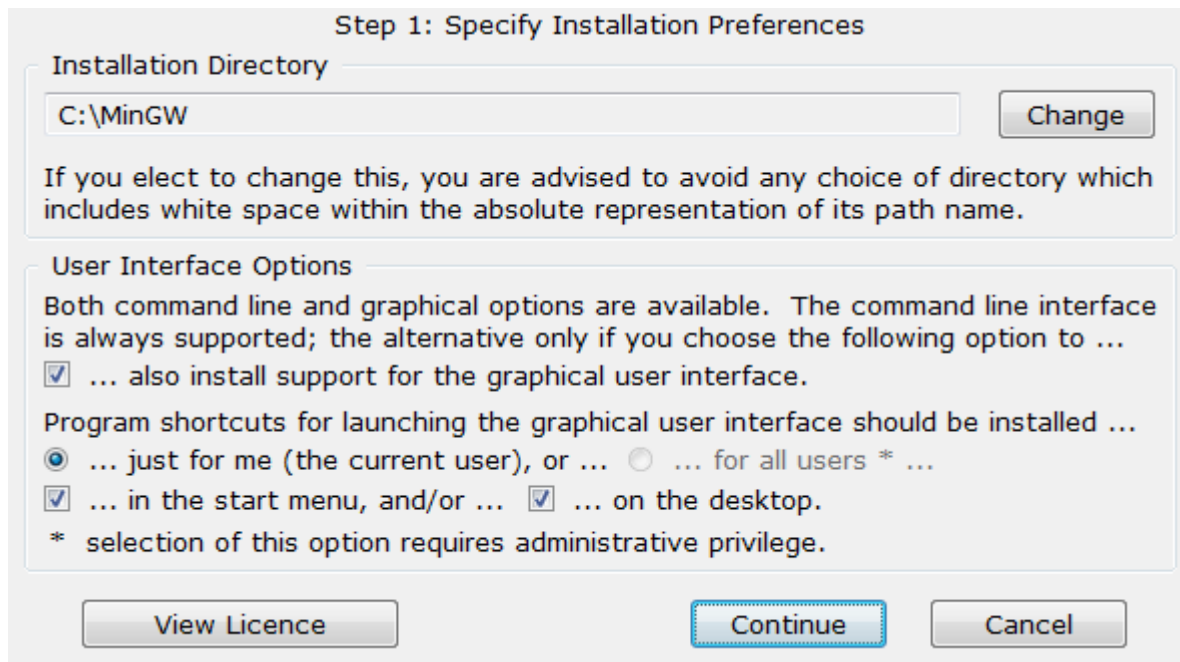
Name	Modified	Size	Downloads / Week
MinGW	2013-10-26		
Installer	2013-10-04		
Other	2011-11-13		
MSYS	2011-11-13		
README	2011-11-13	896 Bytes	205
Totals: 5 Items		896 Bytes	205

Below the table, there is a welcome message and a description of the project. The text says: "Welcome to the MinGW project file distribution directories. This is the top level directory containing Installer, MinGW, MSYS and Other directories. You will find Base, Extension and Contribution directories that are further broken down by package in the MinGW and MSYS directories. You will find mingw-get and mingw-get-inst in the Installer directory. The Other directory contain a hodgepodge of files and packages that have either been contributed or maintained once upon a time but may or may not still be useful. If you have a package you wish to see distributed by MinGW and are willing to commit to contributing it per direction from MinGW..."

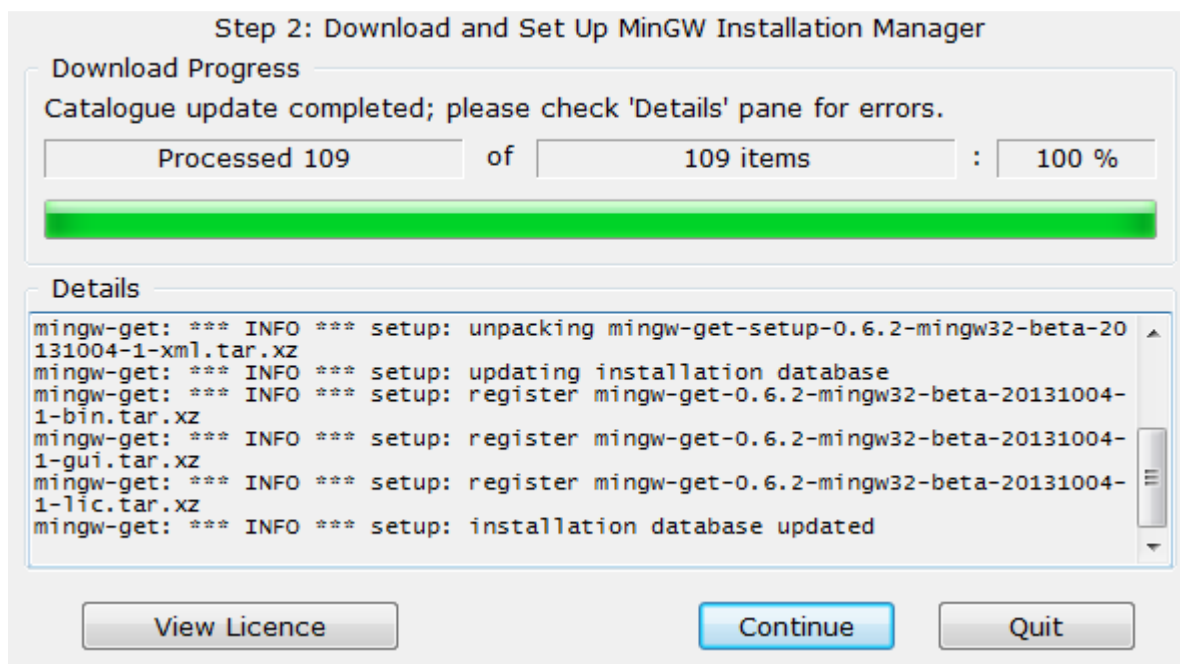
Once the package downloads, open the file and follow the installation instructions. The first screen you see should be this:



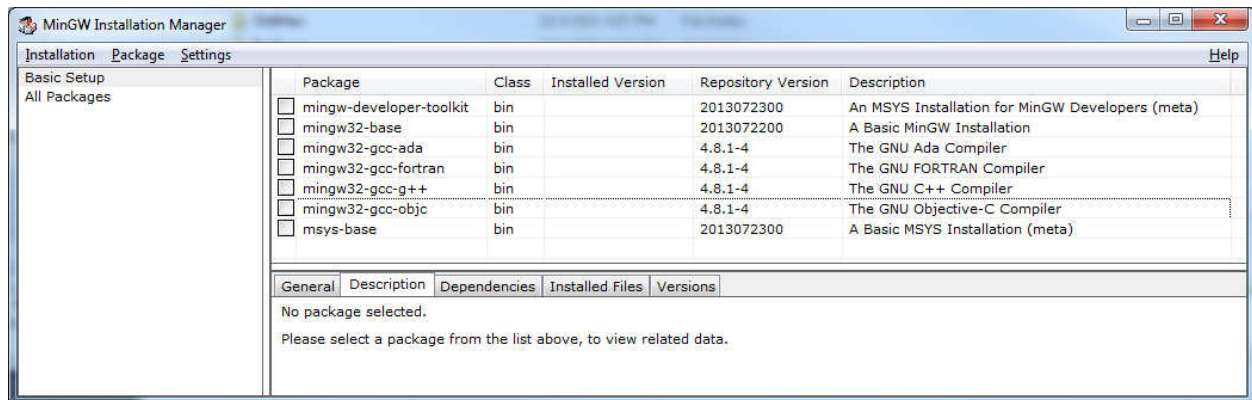
Click on Install and choose the defaults on the next screen. You can install “for all users” if you want, but it isn’t necessary.



Click Continue to install the program. You will see a series of downloads coming in on a screen similar to the one below. It can take a minute or two to finally get everything. Once it is complete the Continue button will be available.



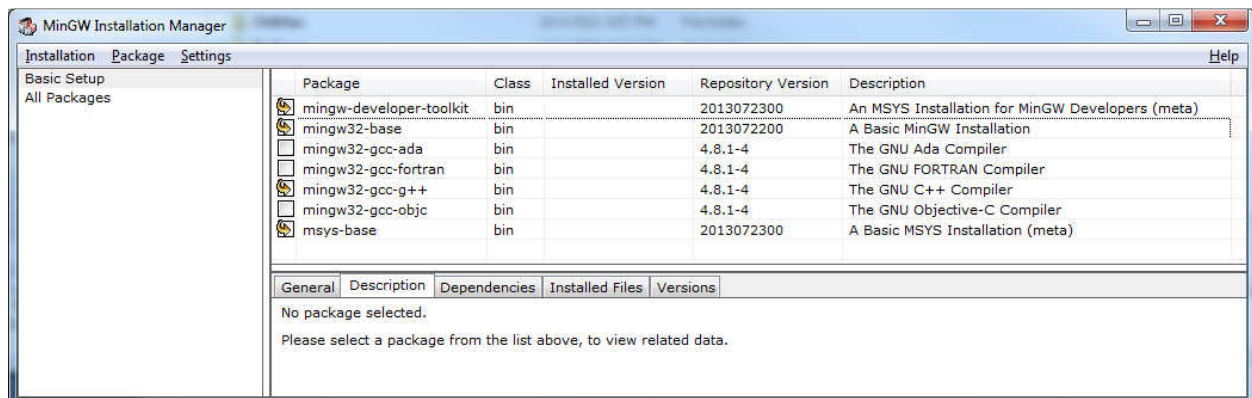
Push the Continue button and the MinGW Installation Manager will be started. If you close it for any reason, there's a link on the Windows Start menu under "All Programs" named MinGW Installation Manager. The window below should appear:



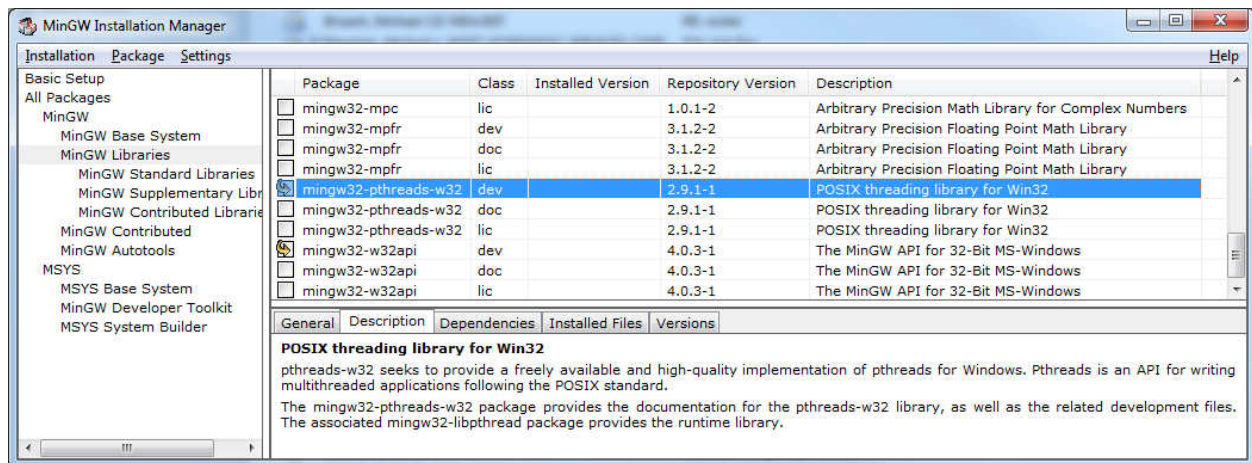
You will need to install the several items. Most of them can be selected from the main page shown above. To select an item, use the right mouse button and choose “Mark for Installation” from the popup menu. The needed items are:

- mingw-developer-toolkit
- mingw32-base
- mingw32-gcc-g++
- msys-base
- mingw32-threads-w32 (dev)

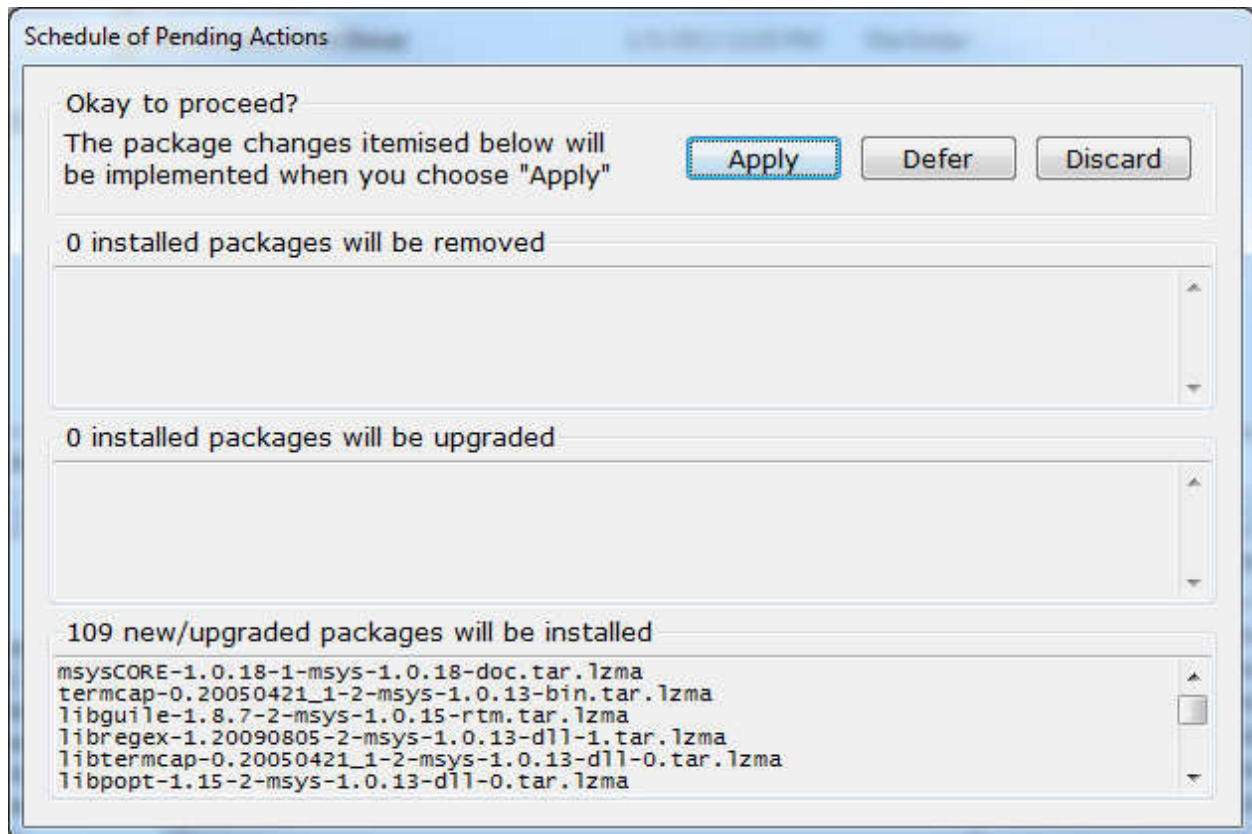
The first four items are available in the “Basic Setup”. Your main window should look like this now:



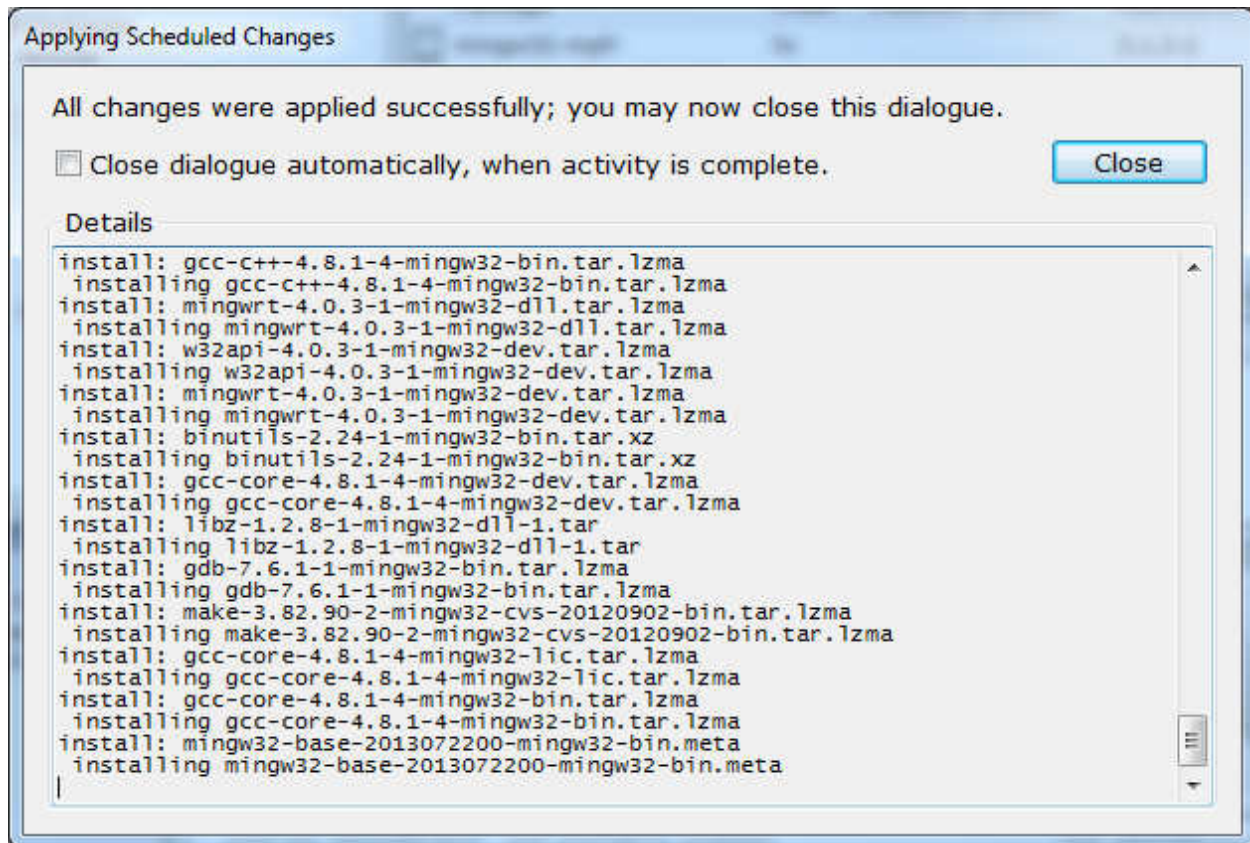
To get the pthreads developer library, select the “All Packages” item in the left hand column and then select MinGW and MinGW Libraries. You’ll have to scroll down to find pthreads. It’s almost at the bottom of the list. Make sure you get the version with Class of “dev”. Otherwise, you’ll have to choose it again. The window below shows where it was selected for installation:



You can now go to the Installation menu and select “Apply Changes”. The following dialog will appear:



Select Apply. You will see a dialog pop up and show the different libraries being downloaded. There are over 100, so it takes a few minutes. Once the files are downloaded...



The required MinGW items are now installed. You can close the MinGW installation manager.

If you installed MinGW in the default location, you can bring up a command window by going to the `c:\MinGW\msys\1.0` directory and double clicking on `msys.bat`. You will need to add a file to the PATH variable. You can do this either through the command line (using `vi`...actually `vim`) or via the Windows operating system with Notepad or something else. If you want to use the command line, the directory should already be set for you (`/home/<username>`). If you want to use Notepad or something else, the directory will be:

`C:/MinGW/msys/1.0/home/<username>`.

Create a file named `.profile` (don't forget the period at the beginning of the name). The only content required in the file is the first line below. The others are helpful for command line users:

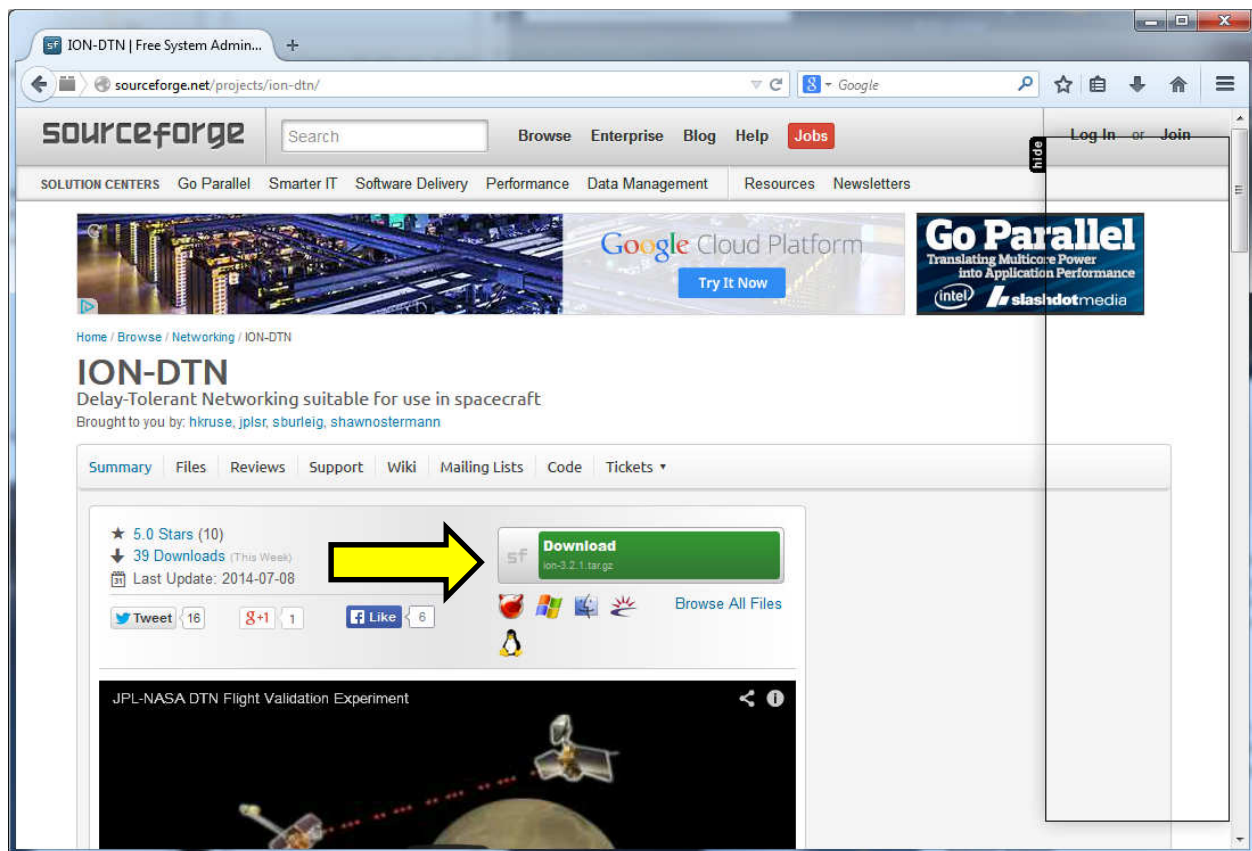
```
export PATH=/c/MinGW/bin:$PATH
alias vi=vim
alias ll='ls -ls'
```

After you edit the file, type `'source .profile'` on the command line. You should be able to type `gcc` at the command prompt and will get an error that there are no input files. If you get an error that says the `gcc`

command was not found, then check your changes to make sure the path listed above is correct. MinGW is now ready to compile ION.

ION

Download ION from SourceForge at <http://sourceforge.net/projects/ion-dtn/>. The latest version is always available on this page as shown below.



You can extract the file using WinZip or 7-Zip or use the following commands in the shell window (the change directory command will vary based on where you download the software):

```
> cd /d/dtn
> tar xf ion-3.2.2.tar.gz
> cd ion-open-source
```

Compiling ION

To compile ION you will need to bring up a command window using the provided batch file (c:\MinGW\msys\1.0\msys.bat). If you followed the steps in the MinGW section everything should be ready to go. You can make sure of that by typing 'gcc' at the command line and see if you get a message that says there are no input files. If you do, continue on. If not, go back and read the MinGW section again to make sure you did everything.

ION compilation with MinGW is a little bit different than compiling on Linux. You will need to run the following command. There's a winion.pdf file available in the ion-open-source directory that contains this information.

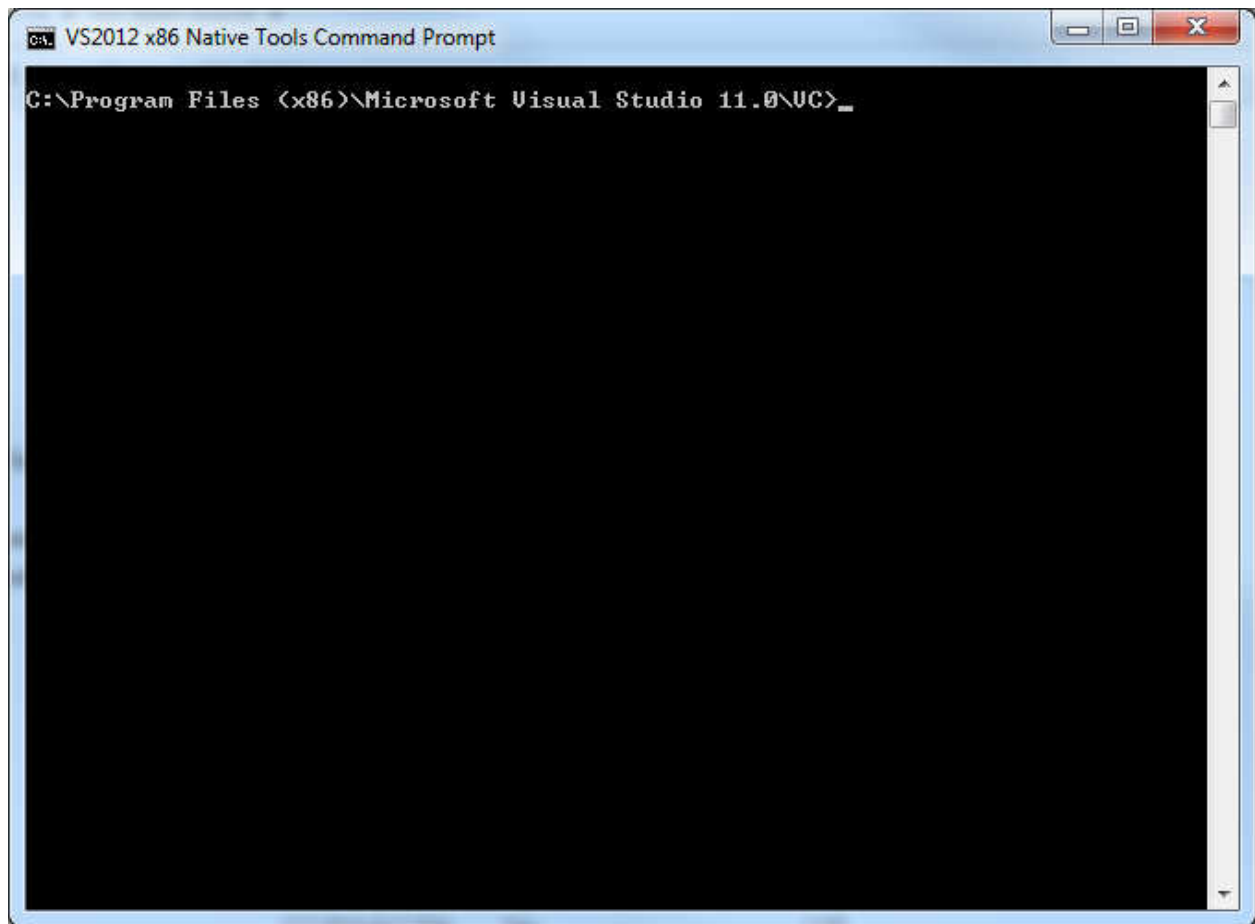
```
> cd /d/dtn/ion-open-source
> mkdir /opt
> ./mingw-setup
> make
```

That should compile and install ION in the /opt directory. You can check the contents of the install directory by running the 'ls -R' command. There should be files in the following directories:

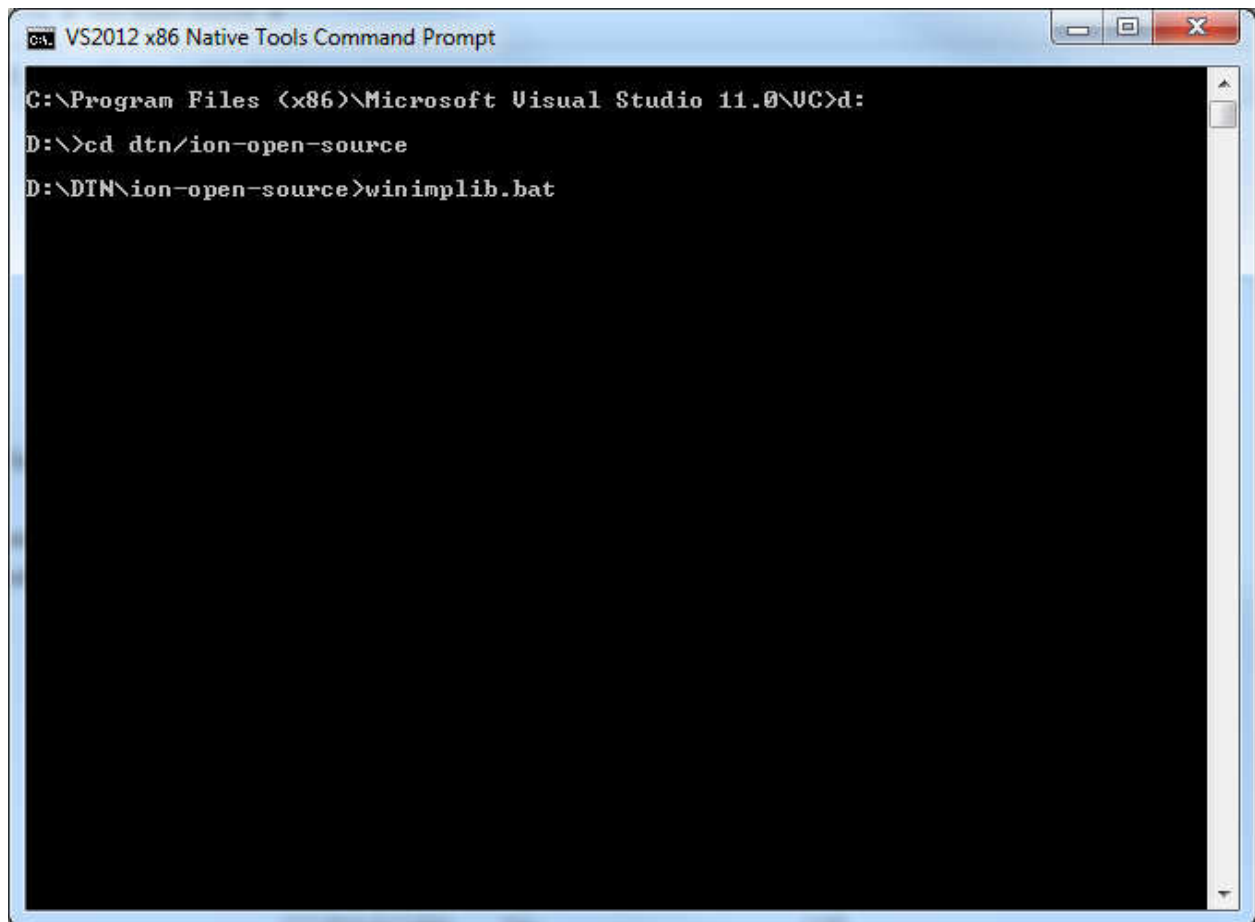
```
/opt/bin
/opt/include
/opt/lib
/opt/man/man1
/opt/man/man3
/opt/man/man5
```

Generating Import Libraries for ION DLLs

To generate the import libraries needed for linking within Visual Studio, you will need to bring up a Visual Studio command prompt similar to the one shown below.



In the command window, change directory to where the ION files are located. Once there run the 'winimp.bat' command as shown in the next screen dump.



```
VS2012 x86 Native Tools Command Prompt
C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC>d:
D:\>cd dtn/ion-open-source
D:\DTN\ion-open-source>winimplib.bat
```

When it completes you will have a new directory named winimplib that contains all of the .lib files needed for linking Visual Studio applications with the ION DLLs.

```
VS2012 x86 Native Tools Command Prompt
1 file(s) copied.
D:\DTN\ion-open-source\dgr\i86-mingw>cd ../../
D:\DTN\ion-open-source>cd ici/i86-mingw
D:\DTN\ion-open-source\ici\i86-mingw>lib /machine:i386 /def:libici.def
Microsoft (R) Library Manager Version 11.00.61030.0
Copyright (C) Microsoft Corporation. All rights reserved.

Creating library libici.lib and object libici.exp
D:\DTN\ion-open-source\ici\i86-mingw>copy *.lib ../../winimplib
libici.lib
1 file(s) copied.
D:\DTN\ion-open-source\ici\i86-mingw>cd ../../
D:\DTN\ion-open-source>cd ltp/i86-mingw
D:\DTN\ion-open-source\ltp\i86-mingw>lib /machine:i386 /def:libltp.def
Microsoft (R) Library Manager Version 11.00.61030.0
Copyright (C) Microsoft Corporation. All rights reserved.

Creating library libltp.lib and object libltp.exp
D:\DTN\ion-open-source\ltp\i86-mingw>copy *.lib ../../winimplib
libltp.lib
1 file(s) copied.
D:\DTN\ion-open-source\ltp\i86-mingw>cd ../../
D:\DTN\ion-open-source>ls winimplib
libams.lib libcfdp.lib libdtn2fw.lib libipnfw.lib libudpcla.lib
libbp.lib libcgr.lib libici.lib libltp.lib
libbss.lib libdgr.lib libimcfw.lib libtcpcla.lib
D:\DTN\ion-open-source>
```

All of the files needed to compile and link ION with Visual Studio are now ready. You will need to set a preprocessor directive named ION4WIN. This will prevent the compiler from trying to load mingw specific include files. You may need to include the libgcc_s_dw2-1.dll and pthreadGC2.dll in addition to the ION DLLs in your run directory when executing your program.