The End All Guide to Building LaGriT in Windows

32 Bit Windows

Setting Up the Windows Build Environment

The user may choose whether to utilize either the Cygwin or MinGW build environments

Cygwin

1. Download Cygwin from (https://cygwin.com/install.html)
2. Execute the setup-x86.exe file.
3. Follow the onscreen directions all the way to the package selection screen.
4. Check the boxes to install gcc-core, gcc-g++, gcc-gfortran, and gdb.
5. Continue the install process. Cygwin will automatically install and configure the packages and create shortcuts in the start menu and desktop to access the terminal.
6. Access the Cygwin environment by running the Cygwin Terminal shortcut from the Start Menu.

MinGW

1. Download MinGW from (http://sourceforge.net/projects/mingw/)
2. Execute the mingw-get-setup.exe file.
3. Follow the onscreen directions all the way to the package selection screen.
4. Check the boxes to intall mingw32-base, mingw32-gcc-g++, and mingw32-gcc-fortran.
5. Continue the install process. Everything will be automatically completed. Click Finish.
6. Add MinGW to the PATH variable by opening a command prompt and executing:

   SET PATH = %PATH%;C:\MinGW\bin;

Install all of the following.

Microsoft HPC Pack 2008 R2

1. Download Microsoft HPC Pack 2008 R2 from (http://www.microsoft.com/en-us/download/details.aspx?id=14737) You want to download mpi_x86.msi when you are prompted.
2. Execute the Microsoft Installer file, following the onscreen directions to install the MPI libraries.

Compiling Libraries and the Code

Installing NetCDF Binaries

1. Download the latest NetCDF (http://www.unidata.ucar.edu/software/netcdf/docs/winbin.html)
   Select the 32-bit version with no DAP.
2. Run the installer and follow the on screen directions.

Compiling Exodus II Libraries

1. Download the latest Exodus II libraries from (http://sourceforge.net/projects/exodusii/files)
2. Unzip the files in any directory of your choosing.
3. Change directories to exodus-x.xx\cbind\src
4. Create a batch script (called install.bat) with the following two lines in it:

   for %%i in (*.c) do gcc -c -m32 -mwindows -I"..\include" -I"C:\Program Files\netCDF 4.3.2\include" -L"C:\Program Files\netCDF 4.3.2\lib" %%i

   Replace the path for netCDF with the one for your computer.

   ar rvs libexodus.a ^
5. Execute the following command to populate the batch script with a list of the object files.

   dir –b *.o >> install.bat
6. Go back to editing the batch script and append a '^' character to the end of each line after the ar rvs libexodus.a ^ line.
7. Execute the script by typing 'install' in the command window.

Compiling LaGriT

1. Clone the latest version of LaGriT from (http://git.lanl.gov/sft/lagrit) onto your computer.

   **** IF USING CYGWIN, MAKE SURE TO CLONE IT INTO THE C:\CYGWIN\home*username*\ FOLDER SO IT CAN BE ACCESSED****
2. Change directories to the lagrit\src folder.
3. Create a directory called objects_win32_o_gfort4.5 (Or replace gfort4.5 with your own compiler version and type. For example gfort4.8 is the current version, 4.9 is in beta. My PC has 4.5)
4. Compile dumpexodusII.f with the following command:

   gfortran –o objects_win32_o_gfort4.5\dumpexodusII.o –fcray-pointer –fno-exceptions –fno-underscoring –m32 –w –c –mwindows dumpexodusII.f
5. Execute the following command to compile the .f90 files into their respective (and required) .mod files.

   gfortran –c *.f90
6. Copy and overwrite the following files:
   a. machine_m32.h overwrites machine.h
   b. mm2000_m32.h overwrites mm2000.h
   c. lagrit_win32.h overwrites lagrit.h
7. Execute 'make'

Linking LaGriT and the Libraries

1. Copy libexodus.a to lagrit\lib folder from the exodus-x.xx\cbind\src folder.
2. Copy netcdf.lib from the netCDF installation lib folder (probably C:\Program Files\netCDF 4.3.2\lib) into the lagrit\lib folder.
3. Rename netcdf.lib to libnetcdf.a. (You may need to enable extension displaying in Windows. To do this, navigate to Control Panel, select Folder Options, click **View**, then uncheck the box labeled 'Hide extensions for known file types'.)

4. Copy hdf5.dll, hdf5_hl.dll, msvcp100.dll, msvcr100.dll, and zlib.dll from the netCDF installation deps\w32\bin folder (probably C:\Program Files\netCDF 4.3.2\deps\w32\bin) into the lagrit\bin folder.
5. Copy netcdf.dll from the netCDF installation bin folder (probably C:\Program Files\netCDF 4.3.2\bin) into the lagrit\bin folder.
6. Link all the libraries together using the command:

   g++ -o "..\bin\lagrit32.exe" –L"..\lib" lagrit_win32_o_gfort4.5.a –m32 –lgfortran –lexodus –lnetcdf

Combining LaGriT and Dependencies Into One Executable

1. Download DLL Packager from (http://code.google.com/p/dllpackager/downloads/list)
2. Run DLL Packager and in the top box, select lagrit32.exe.
3. Then select the shared library files hdf5.dll, hdf5_hl.dll, msvcr100.dll, mscvp100.dll, netcdf.dll, and zlib.dll.
4. Press the combine button and a bundled executable will be generated in the same directory as the original executable. This new executable will not require the dll files at distribution time.

Executing LaGriT

To execute LaGriT, change to the lagrit\bin directory and run the following command, changing the value of n to suit the number of processor cores that your computer has.

"C:\Program Files\Microsoft HPC Pack 2008 R2\bin\mpiexec.exe" –n 1 lagrit32.exe

64 Bit Windows

Setting Up the Windows Build Environment

The user may choose whether to utilize either the Cygwin or MinGW build environments

Cygwin

1. Download Cygwin from (https://cygwin.com/install.html)
2. Execute the setup-x86_64.exe file.
3. Follow the onscreen directions all the way to the package selection screen.
4. Check the boxes to install gcc-core, gcc-g++, gcc-gfortran, and gdb.
5. Continue the install process. Cygwin will automatically install and configure the packages and create shortcuts in the start menu and desktop to access the terminal.
6. Access the Cygwin environment by running the Cygwin Terminal shortcut from the Start Menu.

MinGW

1. Download MinGW from (http://mingw-w64.sourceforge.net/download.php#win-builds)
2. Execute the mingw-get-setup.exe file.
3. Follow the onscreen directions all the way to the package selection screen.
4. Check the boxes to intall mingw32-base, mingw32-gcc-g++, and mingw32-gcc-fortran.
5. Continue the install process. Everything will be automatically completed. Click Finish.

6. Add MinGW to the PATH variable by opening a command prompt and executing:

   SET PATH = %PATH%;C:\MinGW-x64\bin;

Install all of the following.

Microsoft HPC Pack 2008 R2

1. Download Microsoft HPC Pack 2008 R2 from (http://www.microsoft.com/en-us/download/details.aspx?id=14737) You want to download mpi_x64.msi when you are prompted.
2. Execute the Microsoft Installer file, following the onscreen directions to install the MPI libraries.

Compiling Libraries and the Code

Installing NetCDF Binaries

1. Download the latest NetCDF (http://www.unidata.ucar.edu/software/netcdf/docs/winbin.html)
   Select the 64-bit version with no DAP.
2. Run the installer and follow the on screen directions.

Compiling Exodus II Libraries

1. Download the latest Exodus II libraries from (http://sourceforge.net/projects/exodusii/files)
2. Unzip the files in any directory of your choosing.
3. Change directories to exodus-x.xx\cbind\src
4. Create a batch script (called install.bat) with the following two lines in it:
   for %%i in (*.c) do gcc -c –m64 -mwindows -I"..\include" -I"C:\Program Files (x86)\netCDF 4.3.2\include" -L"C:\Program Files (x86)\netCDF 4.3.2\lib" %%i
   Replace the path for netCDF with the one for your computer.
   ar rvs libexodus.a ^
5. Execute the following command to populate the batch script with a list of the object files.
   dir –b *.o >> install.bat
6. Go back to editing the batch script and append a '^' character to the end of each line after the ar rvs libexodus.a ^ line.
7. Execute the script by typing 'install' in the command window.

Compiling LaGriT

1. Clone the latest version of LaGriT from (http://git.lanl.gov/sft/lagrit) onto your computer.
   **** IF USING CYGWIN, MAKE SURE TO CLONE IT INTO THE C:\CYGWIN\home*username*\ FOLDER SO IT CAN BE ACCESSED****
2. Change directories to the lagrit\src folder.
3. Create a directory called objects_win64_o_gfort4.5 (Or replace gfort4.5 with your own compiler version and type. For example gfort4.8 is the current version, 4.9 is in beta. My PC has 4.5)
4. Compile dumpexodusII.f with the following command:
   gfortran –o objects_win64_o_gfort4.5\dumpexodusII.o –fcray-pointer –fno-exceptions –fno-underscoring –fdefault-integer-8 –m64 –w –c dumpexodusII.f

5. Execute the following command to compile the .f90 files into their respective (and required) .mod files.

   gfortran –c *.f90 –m64 –fdefault-integer-8
6. Copy and overwrite the following files:
   a. machine_m64.h overwrites machine.h
   b. mm2000_m64.h overwrites mm2000.h
   c. lagrit_win64.h overwrites lagrit.h
7. Execute '**make MOPT=64**'

Linking LaGriT and the Libraries

1. Copy libexodus.a to lagrit\lib folder from the exodus-x.xx\cbind\src folder.
2. Copy netcdf.lib from the netCDF installation lib folder (probably C:\Program Files\netCDF 4.3.2\lib) into the lagrit\lib folder.
3. Rename netcdf.lib to libnetcdf.a. (You may need to enable extension displaying in Windows. To do this, navigate to Control Panel, select Folder Options, click **View**, then uncheck the box labeled 'Hide extensions for known file types'.)
4. Copy hdf5.dll, hdf5_hl.dll, msvcp100.dll, msvcr100.dll, and zlib.dll from the netCDF installation deps\w32\bin folder (probably C:\Program Files\netCDF 4.3.2\deps\w32\bin) into the lagrit\bin folder.
5. Copy netcdf.dll from the netCDF installation bin folder (probably C:\Program Files\netCDF 4.3.2\bin) into the lagrit\bin folder.
6. Link all the libraries together using the command:

   g++ -o "..\bin\lagrit64.exe" –L"..\lib" lagrit_win64_o_gfort4.5.a –m64 –lgfortran –lexodus –lnetcdf

Executing LaGriT

To execute LaGriT, change to the lagrit\bin directory and run the following command, changing the value of n to suit the number of processor cores that your computer has.

"C:\Program Files\Microsoft HPC Pack 2008 R2\bin\mpiexec.exe" –n 1 lagrit64.exe

The End All Guide to LaGriT Code Changes for Windows Support

1. Fix **srand48** in metis_lg.c, matrix_values_compress.c, metic_macros.h, and metis_proto.h by substituting it with the cross platform function **rand_s**.
2. Change mm2000.F in some way I don't remember to fix multiple definitions error. (May have to do with line 1386)
3. Edit dumpexodusII.f so that the line

   subroutine dumpexodusII(

   reads

   subroutine dumpexodusII_(

   instead.
4. Edit dumpexodusII.f so that the following functions now all end in an underscore.
   a. writloga -> writloga_

    b. mmgetblk -> mmgetblk_
    c. x3d_error -> x3d_error_
    d. icharlnf -> icharlnf_
    e. fexist -> fexist_
    f. cmo_exist -> cmo_exist_
    g. writset -> writset_
    h. dotask -> dotask_
    i. cmo_get_info -> cmo_get_info_
    j. mmfindbk -> mmfindbk_
    k. hassign -> hassign_
    l. excre_wrapper -> excre_wrapper_
    m. exo_lg_ini -> exo_lg_ini_
    n. exo_put_sets -> exo_put_sets_
    o. mmprint -> mmprint_
    p. mmrelprt -> mmrelprt_
    q. pntlimc -> pntlimc_
    r. eltlimc -> eltlimc_
    s. mmrelblk -> mmrelblk_
5. Replace the following Exodus function names to reflect the ExodusII update. Make sure to use continuation characters when needed.
    a. exerr -> ex_err_wrapper
    b. expqa -> ex_put_qa_wrapper
    c. expcor -> ex_put_coord_wrapper
    d. expelb -> ex_put_elem_block_wrapper
    e. expelc -> ex_put_elem_conn_wrapper
    f. expnp -> ex_put_set_param_wrapper
    g. expns -> ex_put_set_wrapper
    h. expsp -> ex_put_side_set_param_wrapper
    i. expss -> ex_put_side_set_wrapper
    j. exclos -> ex_close_wrapper
6. Change excre_wrapper.c so that calls to excre now call ex_create_int instead.
7. Change the following files and insert a section for win64 in all of them that fixes the inappropriate type sizes.
    a. opsys.h
    b. lagrit_win64.h
    c. skiplist.ch
    d. type_sizes.h
8. Edit mmsc.c such that the functions util_malloc and util_realloc have a type cast to fix the memory clobbering issues caused by the invalid type conversion in Windows. (Read more about it in the comments)
9. Edit cmo_memory.f to add in 2 lines and a new variable to fix a memory allocation clobbering problem that is present in the 64-bit version. The variable is integer k, the two lines are k=j, and print *, k. Without these lines, the 64 bit version is incapable of allocating memory to store a mesh object inside. It ends up in an infinite loop whereby j is set to 0 and never increments.
10. Write all of the wrapper functions that are used in the above mentioned exodusII library compatability step. These function are:
    a. ex_err_wrapper
    b. ex_put_qa_wrapper

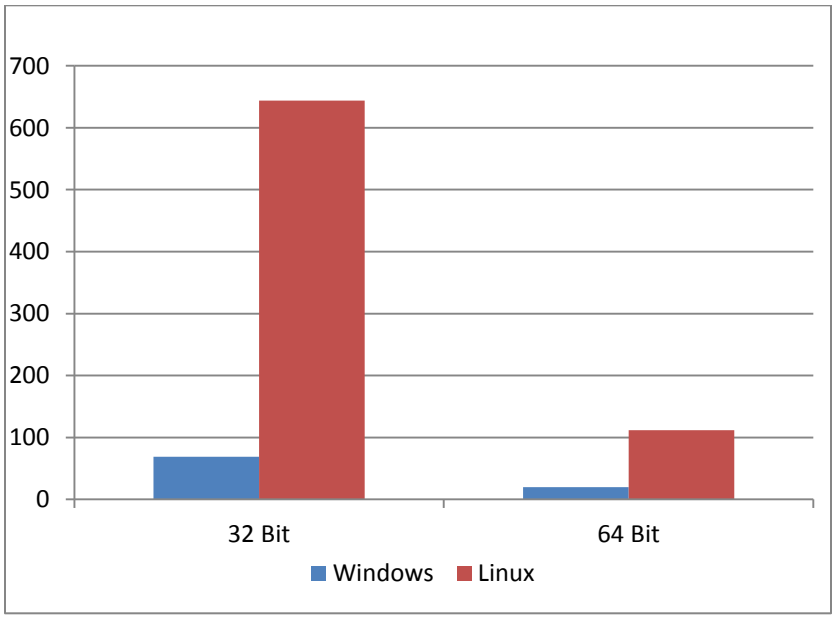
c. ex_put_coord_wrapper
d. ex_put_elem_block_wrapper
e. ex_put_elem_conn_wrapper
f. ex_put_set_param_wrapper
g. ex_put_set_wrapper
h. ex_put_side_set_param_wrapper
i. ex_put_side_set_wrapper
j. ex_close_wrapper

See the code for more details on the operation. They basically just translate the invalid function calls that LaGriT makes into the correct syntax ones for the latest Exodus II release.

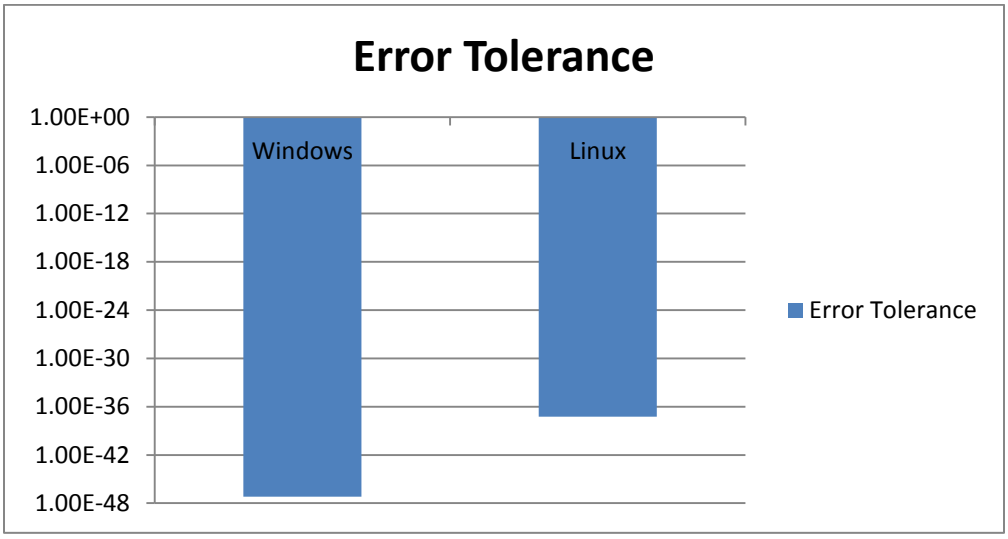
11. Edit writinit.f to add in sections for both Windows 32 and 64 bit versions. These changes just make sure that the correct header information is displayed.
12. Add in date functions for Windows to fix the error in date formatting. These functions are contained in the files date_time.f and datestring.f.
13. Edit writinit.f to include the changes that these date function make. See the comments for more detail. There are several new lines added to that file for the date function.
14. Fix error in ex_put_conn by casting the blk_id as an int. This error is unexplainable but completely necessary for anything to work correctly.
15. Fix date error in fdate_2_mmddy in file dumpgmv_hybrid.f using conditionally executed code.

Advantages of Using Windows

1. **Drop in LaGriT Execution Time:**



2. **Increase in variable precision (64 bit only):**



3. **Dates actually format themselves correctly:**

Windows: 2014/08/07 Linux: 2014/Aug 7

Problems of Using Windows

1. **Mesh objects are limited in amount of memory (64 bit mainly):**