Overture

Table of Contents

Overture

This page documents the installation of a **custom version of Overture** that does not have any Motif, OpenGL or X11 dependencies available as a tarball & in our SVN repository. Note: These instructions are **not** valid for the latest version of Overture available at the Overture Install page.

Resources:

- Official Overture page
- An Overture user has setup Doxygen documentation of Overture.

Overture Dependencies & Preqrequisites

Ensure that you have the HDF, A++ & P++ libraries and header files on your system. If you do not know the location of these libraries, or if they are not installed ask your system administrator. If you are the system administrator, follow the instructions below for obtaining and installing these dependencies:

HDF

- Parallel LFM requires HDF4 (or newer), which is available on the HDF4 website. If HDF is not already on your system, download and install HDF4 according to the instructions in the INSTALL document. If you have problems building HDF, see the download and install HDF documentation on the Overture website.
- Note: You need to have *both* the libraries *and* the header files installed to properly compile overture.

A++ & P++

- You should have already built and installed A++ and P++ according to the instructions at A+/P+.

Overture Build Instructions

1. Download tarball of source code: overture-02-19-2008.tar.gz and unpack to `$INSTALL_DIR/overture`:

   ```
   cd $INSTALL_DIR
   wget http://download.hao.ucar.edu/pub/schmitt/cism/LTR-para/overture-02-19-2008.tar.gz
   tar -zxvf overture-02-19-2008.tar.gz
   ```

   Developers who want the latest & greatest (and quite possibly the most broken version of code) may download from the CISM repository:
   Note: This requires an the UCAR Subversion server

   ```
   cd $INSTALL_DIR
    svn --username [yourname] co https://proxy.subversion.ucar.edu/cism_CSE/trunk/overture
   ```

2. Change directories & set the following environment variables in your corresponding shell. On an IBM AIX system using csh, this looks like:

```
cd $INSTALL_DIR/overture
 setenv APlusPlus $INSTALL_DIR/A++
 setenv PPlusPlus $INSTALL_DIR/P++
 setenv HDF $INSTALL_DIR
 setenv Overture $INSTALL_DIR/overture
 setenv XLIBS IGNORE_X
 setenv OpenGL IGNORE_X
 setenv MOTIF IGNORE_X
 setenv PERL IGNORE_PERL
```

See the `env.example` file included with our Overture distribution as an example of the environment variables you need to set.

3. Configure and build:
   On some 64-bit Linux systems (i.e. OS = "linux" from below), you must first edit `config/MakeDefs.linux` and replace `-DINTEL86` with

```
-D__x86_x64__ -DBL_AUTO_INSTANTIATE
```

and then run configure:

```
./configure [OS]  CC=${CXX} cc=${CC} FC=${F77}
```

where `[OS]` is replaced with one of `aix, dec, dec-linux, dec-linux-compaq, hpux, linux, irix, sun5` (run `sh configure --help` for more information). `$CC`, `$CXX` and `$F77` are environment variables pointing to your C, C++ and Fortran compilers. You may wish to compile an optimized parallel version of Overture (note: We have not extensively tested the optimized parallel version of Overture) by issuing the following commands:

```
chmod gu+x configure
    sh configure [OS] opt parallel CC=${CXX} cc=${CC} FC={F77}
```

Warning: It can take a <u>long</u> time to build the optimized version of Overture.
On some 64-bit Linux machines (i.e. OS = "linux" from above), you must edit `boxlib/Makefile`, swapping

```
-D__x86_x64__ -DBL_AUTO_INSTANTIATE
```

with

```
-D__x86_x64__ -DBL_AUTO_INSTANTIATE -D__ia64
```

4. Build the code

```
gmake
```

   a. Warning: On machines with limited memory, Overture (specifically dpmVector.C) may take *several* hours to build thanks to lots of C++ templates which require the C pre-processor to to utilize lots of memory. We've seen this take 25 hours and utilize over 2.5 Gb of ram using the PGI compilers (on Kraken)!
   b. Certain machines have an error linking `bin/ogen`:

```
async.c:-( .text+0x32): undefined reference to `aio_suspend'
async.c:-( .text+0x64): undefined reference to `aio_return'
async.c:-( .text+0x8d): undefined reference to `aio_error'
/export/pgf70/linux86-64/7.0-3/lib/libpgftnrtl.a(async.o): In function  `Fio_asy_read':
async.c:-( .text+0x2b0): undefined reference to `aio_read'
/export/pgf70/linux86-64/7.0-3/lib/libpgftnrtl.a(async.o): In function  `Fio_asy_write':
async.c:-( .text+0x362): undefined reference to `aio_write'
```

To fix this, add `-lrt` to the end of FORTRAN_LIBS in `bin/Makefile`

5. You have now built and installed the prerequisites.

64-bit Linux with Intel Compilers

Edit `config/MakeDefs.linux` and do the following:

Replace `-DINTEL86` with

```
-D__x86_x64__ -DBL_AUTO_INSTANTIATE
```

Add the following to the lines that specify `$GLIBS`:

```
-L/usr/lib64 -lg2c -lstdc++
```

Run `./configure ...` (e.g. by following the standard Overture build instructions). Before running `gmake`, you should edit boxLib/Makefile and replace the line containing "`x86_x64`" with

```
-D__x86_x64__ -D__ia64
```

Otherwise, you may get errors about "nld" and "nrd" being un-defined.
Then run

```
./configure linux cc=icc CC=icc FC=ifort
    gmake
```

Developer website

http://www.llnl.gov/CASC/Overture

Troubleshooting

General Troubleshooting Notes

The notes below say to "edit the appropriate Makefile". You have two options for this:

1. Edit `config/Makedefs.$ARCHITECTURE`, gmake clean, and re-run the `configure` script.
2. Note the last directory you were in when getting an error and edit the makefile in that directory. So, if the last few lines during the error are

```
gmake[1]: *** [rap] Error 1
gmake[1]: Leaving directory }}/misc/p63/cmit2/pschmitt/releaseCandidate/opt/overture/bin'
gmake: *** [all] Error 2
```

Note the last directory: `overture/bin`, so you would edit `bin/Makefile`.

To build on a 64-bit Intel machine (like wjet and STIC), we had to edit config/Makedefs.linux and replace `-DINTEL86` with

```
-D__x86_64__
```

Issues

- While building Overture, the Portland Group C++ compiler may complain about `offsetof`:

```
"plyFileInterface.C", line 153: error: identifier "__offsetof__" is undefined
{"x", PLY_FLOAT, PLY_FLOAT, offsetof(PlyVertex,x), 0, 0, 0, 0},
^

"plyFileInterface.C", line 194: error: identifier "__offsetof__" is undefined
{"vertex_indices", PLY_INT, PLY_INT, offsetof(PlyFace,verts),
```

Solution: Define your own `offsetof`, as suggested on the Portland Group C++ forum:

```
#define offsetof(TYPE, MEMBER) ((size_t) &((TYPE *)0)->MEMBER)
```

- While building Overture, the GNU C++ compiler (v. 4.3+) complains about missing standard header files, such as:

```
iostream.h: No such file or directory
```

There are mainly 2 changes here; (1) GCC 4.3 does not recognize iostream.h anymore and will comply with the latest C++ standards of using iostream and (2) you will need to have `using namespace std;` at the top for it to recognize iostream. The simplest fix is to install the g++ 3.3 header files & libraries. For more information, see this article.
- While building Overture, you may get error like:

```
/home/pschmitt/cmit2/pschmitt/opt-intel/overture/bin/../lib/libOverture.so: undefined reference to
`dtime_'
```

Solution: Add `-lg2c` to the GLIBS environment variable of the appropriate Makefile (i.e. if you get this crash in overture/bin, edit overture/bin/Makefile). Note: You might also need to tell the compiler where the `libg2c.a or libg2c.so` library resides. Sometimes it's in `/usr/lib64`, in which case you need to add `-L/usr/lib64 -lg2c`.
- While compiling Overture on a 64-bit AIX machine with IBM compilers, you may get an error like:

```
"/hao/wiltbemj/opt/AIX-64/src/overture/include/UnstructuredMapping.h", line 428.47: 1540-0216 (S) An
expression of type "void *" cannot be converted to type "int".
```

This is fixed in Overture version 2.2 & in the latest version of Overture from SVN & the Overture tarball overture-02-19-2008.tar.gz
- When building overture, you may get an error while compiling "overture/Ogshow/OvertureParser.C" about perl dependencies. First, manually edit "Overture/Ogshow/OvertureParser.C". Replace line 11 with

```
#ifndef IGNORE_PERL
#include <perl.h>
#endif
```

Second, manually edit "Overture/Ogshow/Makefile" and add `-DIGNORE_PERL` to `cc_FLAGS` and `CC_FLAGS`. Try recompiling.
- When building overture, you may get an error while compiling "overture/Cad/fillVolumeWithSpheres.C" of the form

```
error: more than one instance of
       overloaded function "abs" matches the argument list:
          function "abs(int)"
          function "std::abs(long)"
          function "std::abs(long double)"
          argument types are: (double)
   if (abs(1.0 - tvf) > distTolerance) {
```

Using your favorite text editor, edit "overture/Cad/fillVolumeWithSpheres.C". Replace every instance of `abs` with `fabs` and add `#include <cstdlib>` to the first line of the file.
- At the link step of "overture/bin/plotstuff", you may get an error like:

```
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`BoxList::__ct(BoxList const &)':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:46: undefined reference to `List<Box>::__ct
(List<> const &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`BoxList::operator=(BoxList const &)':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:50: undefined reference to
`__CPR63____as__16List__tm__5_3BoxFRC15List__tm__4_Z1Z_RJ27J'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`BoxList::intersect(Box const &)':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:128: undefined reference to `List<Box>::
remove(ListIterator<> &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`BoxList::complementIn(Box const &, BoxList const &)':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:155: undefined reference to `List<Box>::
catenate(List<> &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:156: undefined reference to `List<Box>::
remove(ListIterator<> &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`BoxList::simplify(void)':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:275: undefined reference to `List<Box>::
remove(ListIterator<> &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`length':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:316: undefined reference to `Array<Box>::
length(long const(void))'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`operator[]':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:344: undefined reference to
`get__17Array__tm__5_3BoxCFl_RCZ1Z'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`clear':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:465: undefined reference to `List<Box>::
clear(void)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`length':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:481: undefined reference to `List<Box>::
length(int const(void))'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`complementIn':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:406: undefined reference to `List<Box>::
catenate(List<> &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:407: undefined reference to `List<Box>::
remove(ListIterator<> &)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`List<Box>::__dt(void)':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:542: undefined reference to `List<Box>::
clear(void)'
/local/d/schmitt/cism/opt-pgi-mpich/overture/bin/../lib/libOverture_static.a(BoxList.o): In function
`append__16List__tm__5_3BoxFRCZ1Z_v':
/local/d/schmitt/cism/opt-pgi-mpich/overture/boxlib/BoxList.C:557: undefined reference to
`__CPR79__addAfter__16List__tm__5_3BoxFP19ListLink__tm__4_Z1ZRCZ1Z_PJ30J'
```

Solution: add the flag `-DBL_AUTO_INSTANTIATE` to `cc_FLAGS`, `CC_FLAGS` and `FF_FLAGS` in config/MakeDefs.ARCHITECTURE. Then run `gma ke clean` and return to step 2 above (re-run `configure` and `gmake` to rebuild Overture).

- At the link step of "overture/bin/plotstuff", you may get an error like:

```
async.c:(.text+0x26b): undefined reference to `aio_read'
/opt/local/pgi7.1/linux86/7.1/lib/libpgftnrtl.a(async.o): In function `Fio_asy_write':
async.c:(.text+0x33b): undefined reference to `aio_write'
```

These symbols are defined in librt. Try adding `-lrt` to "Lib" inside "overture/bin/Makefile".
- At the link step of "overture/bin/plotstuff", you may get an error like:

```
/local/d/schmitt/cism/opt/lib/libdf.a(cszip.o): In function `HCPcszip_write':
cszip.c:(.text+0x38d): undefined reference to `SZ_encoder_enabled'
```

Looks like HDF was compiled with libsz? Try adding -lsz to "Lib" inside "overture/bin/Makefile".
* When building overture, you may get an error of the form

```
'''* No rule to make target 'ABC.h' needed by 'XYZ'.  Stop.
```

This error may go away by running the `gmake` command again.
* When building Overture with the Intel compilers on a 64-bit Intel machine, you may get errors from `hdfi.h` about "GOT_MACHINE" of the form

```
If you get an error on this line more than one machine type has been defined.
Please check your Makefile.
```

If this happens, edit the overture/config/MakeDefs.[OS] and replace the `-DINTEL86` flag with

```
-DINTEL64 -D__386 -DUNIX386
```

* When building Overture with the Intel compilers on a 64-bit Intel machine, you may get errors while building `boxlib/FPC.c` of the form

```
FPC.C(79): error: identifier "nld" is undefined
      return nld;
             ^
FPC.C(109): error: identifier "nrd" is undefined
      return nrd;
             ^
FPC.C(149): catastrophic error: #error directive: We do not yet support FAB I/O on this machine
  #error We do not yet support FAB I/O on this machine
```

Solution: edit "config/MakeDefs.linux" and add `-D__ia64` to the `CFLAGS`. Run `gmake clean` and return to step 2 above: re-run `configure` and `gmake`.
* While building Overture with Intel compilers on a 64-bit Intel machine, you may get an error while building `overture/boxlib/FPC.C`. Edit `boxlib/Makefile` and add

```
-D__i486__
```

to `cc_FLAGS` and `CC_FLAGS`.
* While building Overture with Intel compilers on a 64-bit Intel machine, you may get an error while building `overture/static/mogl.C` of the form

```
mogl.C catastrophic error:  cold not open source file X11/GLw/GLwMDraw.h
```

. First, try adding `-DOV_USE_MESA` to `cc_flags` and `CC_flags` of "overture/static/Makefile". If that doesn't work, make sure your system has "GLwMDraw.h" installed. It may be installed in "/usr/include/GL/GLwMDraw.h" Edit "mogl.C". Remove "X11/" from line 167 of `overture/static/mogl.C`
* When building Overture with the Portland Group Compilers, you may get an error:

```
pgCC-Error-Unknown switch: -Wno-deprecated
```

Then you must manually remove any reference to the "-Wno-deprecated" flag from overture/config/MakeDefs.OS and then run the Overture `configure` script again.
* When building overture, you may get an error like:

```
catastrophic error:
          #error directive: "SEEK_SET is #defined but must not be for the C++ binding of MPI"
   error "SEEK_SET is #defined but must not be for the C++ binding of MPI"
```

Solution: add `-DMPICH_IGNORE_CXX_SEEK` and `-DMPICH_SKIP_MPICXX` to `cc_FLAGS` and `CC_FLAGS` of `overture/config/MakeDefs.O`. `gmake clean` and re{{configure}} and `gmake` overture.

- While building Overture with Intel compilers (ifort), you get the error:

```
ifort  -fPIC -w -inline-level=0 -r8 -i4   -I/dev/shm/wilsone/overture/include -I.  -c
assignBoundaryConditions.f
.
.
.
Fatal compilation error: Out of memory asking for 69640.
```

Solution: Compile assignBoundaryConditions.f with flag `-O0`:

```
cd GridFunction/ && ifort  -fPIC -w -O0 -r8 -i4   -I/dev/shm/wilsone/overture/include -I.  -c
assignBoundaryConditions.f
```

- While building Overture with the gnu compilers (gcc, g++, gfortran), you get the error:

```
gfortran -O -fPIC  -r8 -i4   -I/tor/d/schmitt/cism/opt-gnu/overture/include -I.  -c
assignBoundaryConditions.f
gfortran: unrecognized option '-r8'
 In file assignBoundaryConditions.f:3969
             stop 123456
                        1
Error: Too many digits in STOP code at (1)
 In file assignBoundaryConditions.f:4279
             stop 123456
                        1
Error: Too many digits in STOP code at (1)
 In file assignBoundaryConditions.f:4581
             stop 123456
                        1
Error: Too many digits in STOP code at (1)
```

Solution: Edit "GridFunction/assignBoundaryConditions.f" and change every instance of "123456" with "12345"
- When building Overture on an IBM machine using the IBM compilers, you get the error

```
ld: 0711-781 ERROR: TOC overflow. TOC size: 122376     Maximum size: 65536
```

Solution: Add the flag `-bbigtoc` to the corresponding Makefile & command. It should compile, even though you get the warning:

```
ld: 0711-783 WARNING: TOC overflow. TOC size: 122376     Maximum size: 65536
        Extra instructions are being generated for each reference to a TOC
        symbol if the symbol is in the TOC overflow area.
```

- When building Overture on an IBM machine using the IBM compilers, you get the error

```
ld: 0706-006 Cannot find or open library file: -l Rapsodi
        ld:open(): A file or directory in the path name does not exist.
```

even though you are certain that this library is stored in `$(Overture)/lib` and the compile command looks okay. Solution: Open the Makefile and add the following before including the library `-lRapsodi`:

```
-Wl,-brtl,-blibpath:$(Overture)/lib
```

- When building Overture, you get several "undefined reference to ..." errors of the form

```
/home/pschmitt/cmit2/pschmitt/releaseCandidate/opt/overture/bin/../lib/libOverture.so: undefined
reference to `std::basic_ostream<char, std::char_traits<char> >& std::flush<char, std::char_traits<char>
>(std::basic_ostream<char, std::char_traits<char> >&)'
```

Solution: Edit the appropriate Makefile and add `-lstdc++` to `FLIBS`.

- When building Overture with IBM compilers, you get the following error:

```
xlC -DIGNORE_X -DIGNORE_PERL -g -qspill=50000 -qmaxmem=-1 -w -DFBIND=1   -I/home/bluevista/schmitt/cism
/releaseCandidate/opt/overture/bin/../include -I/home/bluevista/schmitt/cism/releaseCandidate/opt/A++
/include -IIGNORE_X/include -IIGNORE_X/include  -DBL_USE_DOUBLE -DBL_Solaris  -I -o ogen ogenDriver.o
ogen.o -Wl,-brtl,-blibpath:/usr/lib/threads:/usr/lib:/home/bluevista/schmitt/cism/releaseCandidate/opt
/overture/bin/../lib::/home/bluevista/schmitt/cism/releaseCandidate/opt/overture/bin/../lib:/usr/lib
/threads:/usr/lib -L/home/bluevista/schmitt/cism/releaseCandidate/opt/overture/bin/../lib -lOverture -
lOverture_static -L/home/bluevista/schmitt/cism/releaseCandidate/opt/A++/lib -lApp -lApp_static  -lf  -
Wl,-rpath,/home/bluevista/schmitt/cism/releaseCandidate/opt/lib -L/home/bluevista/schmitt/cism
/releaseCandidate/opt/lib -lmfhdf -ldf -ljpeg -lz
ld: 0706-012 The -p flag is not recognized.
ld: 0706-012 The -a flag is not recognized.
ld: 0706-012 The -t flag is not recognized.
ld: 0706-012 The -h flag is not recognized.
gmake[1]: *** [ogen] Error 255
```

Solution: The IBM compiler doesn't understand the `-rpath` flag; Remove it from the corresponding Makefile.

- When building Overture on an IBM AIX system, the compiler complains about a missing library "Rapsodi". Solution: Edit the corresponding makefile, modifying the "LibRapsodi" line like so:

```
LibRapsodi= -Wl,-brtl,-blibpath:$(Overture)/lib
```

- When building Overture with certain MPI compiler combinations, you may find that definitions in include/List.H, include/List.C (and all the sources that use those) clash with the "List" class defined inside the MPI library. i.e. you may see errors like:

```
[]$ gmake
mpiCC -DIGNORE_X -DIGNORE_PERL -fPIC -DBL_AUTO_INSTANTIATE -
fpermissive -DIGNORE_PERL -I/home/ccmc/LFM_MIX_install/overture/
include -I.  -D__x86_64__  -I/home/ccmc/LFM_MIX_install/P++//include
-IIGNORE_X/include -IIGNORE_X/include  -DBL_USE_DOUBLE -DBL_Solaris -I
-DHAS_MODFL_PROTO -g -DNO_REAL  -I/home/ccmc/LFM_MIX_install/include -c grid.C
/home/ccmc/LFM_MIX_install/overture/include/List.H:43: error: 'List'
is not a template type
/home/ccmc/LFM_MIX_install/overture/include/List.H:49: error: 'List'
is not a template
cd boxlib
/home/ccmc/LFM_MIX_install/overture/include/List.H:83: error: 'List'
is not a template
/home/ccmc/LFM_MIX_install/overture/include/List.H:164: error: 'List'
is not a template
/home/ccmc/LFM_MIX_install/overture/include/List.H:169: error: 'List'
is not a template
/home/ccmc/LFM_MIX_install/overture/include/List.H:176: error: 'List'
is not a template
/home/ccmc/LFM_MIX_install/overture/include/List.H:217: error: 'List'
is not a template type
/home/ccmc/LFM_MIX_install/overture/include/List.H:217: error:
redefinition of 'class List'
/export/mpich/include/mpi2c++/mpi2c++_list.h:34: error: previous
definition of 'class List'
/home/ccmc/LFM_MIX_install/overture/include/List.H:400: error: 'List'
is not a template
...
```

Solution: Change `List<T>`, `List ()` and `::List` to `OgshowList<T>`, `OgshowList ()`, `::OgshowList` in all the Overture files.
* When building Overture with the Portland Group Compilers (6.2-3 and earlier), you get the error:

```
/usr/local/pgi/linux86-64/6.2/lib/libpgc.a(trace_lin.o)(.text+0x10): In function `findrout':
: undefined reference to `__pgi_trace'
/usr/local/pgi/linux86-64/6.2/lib/libpgc.a(trace_lin.o)(.text+0x18): In function `findrout':
: undefined reference to `__pgi_tracee'
```

Solution: Add this to the link line:

```
-L/usr/local/pgi/linux86-64/6.2/lib /usr/local/pgi/linux86-64/6.2/lib/pgi.ld -lpgc
```

Note: This has been fixed in the 6.2-4 PGI compilers
* When compiling, you get the error

```
"HDF_DataBase.C", line 1788: error: identifier "MAX_VAR_DIMS" is undefined
int32 tag, ref, rank, nt, dims[MAX_VAR_DIMS|MAX_VAR_DIMS], nattrs;
       ^

"HDF_DataBase.C", line 1790: error: identifier "MAX_NC_NAME" is undefined
char sd_name[MAX_NC_NAME|MAX_NC_NAME];
     ^

"HDF_DataBase.C", line 1882: error: identifier "MAX_VAR_DIMS" is undefined
int32 tag, ref, rank, nt, dims[MAX_VAR_DIMS|MAX_VAR_DIMS], nattrs;
       ^

"HDF_DataBase.C", line 1884: error: identifier "MAX_NC_NAME" is undefined
char sd_name[MAX_NC_NAME|MAX_NC_NAME];
     ^

"HDF_DataBase.C", line 1976: error: identifier "MAX_VAR_DIMS" is undefined
int32 tag, ref, rank, nt, dims[MAX_VAR_DIMS|MAX_VAR_DIMS], nattrs;
       ^

"HDF_DataBase.C", line 1978: error: identifier "MAX_NC_NAME" is undefined
char sd_name[MAX_NC_NAME|MAX_NC_NAME];
```

Solution: Find and replace all instances of MAX_ with H4_MAX_. Here's an example using Perl:

```
perl -pi -e "s/MAX_VAR/H4_MAX_VAR/g" DataBase/*C
```

- When building with g77, you get the error

```
cc1: error: unrecognized command line option "-i4"
```

Solution: Edit the corresponding Makefile and remove "-i4" from the corresponding FFLAGS.
- When building, you get the error

```
rap.C(32): error: a value of type "const char *" cannot be used to initialize an entity of type "char *"
char * dot = strrchr((const char*) line, '.');
             ^

compilation aborted for rap.C (code 2)
make[1]: *** [rap.o] Error 2
make[1]: Leaving directory `/home/col/lfm/lfm_2_1_4/overture/bin'
make: *** [all] Error 2
```

Solution: edit line 32 of bin/rap.C and add "const_cast":

```
char * dot = const_cast<char *>( strrchr((const char*) line, '.') );
```

If you still see errors, you might need to specify the namespace like so: std::const_cast.

- When building, you get the error

```
gmake[1]: Entering directory `/nics/d/home/schmitt/opt-pgi-11.9/overture/bin'
/usr/bin/ld: cannot find -lOverture
gmake[1]: *** [ogen] Error 2
gmake[1]: Leaving directory `/nics/d/home/schmitt/opt-pgi-11.9/overture/bin'
gmake: *** [all] Error 2
```

For whatever reason, your environment is having hard time linking to dynamic libraries. Try adding whatever compiler flag you need to link to dynamic libraries. With the PGI compilers, you need to edit bin/Makefile and add "-Bdynamic" to "LibOverture"