openCMISS

0.1

Generated by Doxygen 1.5.7.1

Wed Oct 22 15:25:15 2008

Contents

Chapter 1

openCMISS Documentation

An open source interactive computer program for Continuum Mechanics, Image analysis, Signal processing and System Identification. Target usage: Bioengineering application of finite element analysis, boundary element and collocation techniques.

1.1 LICENSE

Chapter 2

Obtaining the Code and Setting up the Development Environment

2.1 Obtaining the Code and Libraries

To obtain the openCMISS source you need to check it out from the subversion repository. There are two parts to openCMISS to obtain - openCMISS itself and the various libraries it needs. In your root openCMISS directory, make the opencmiss and opemcmissextras directories.

2.1.1 Obtain the Code

The openCMISS repository is at https://opencmiss.svn.sourceforge.net/svnroot/opencmiss/cm
To check out the main trunk of openCMISS issue the following command in the opencmiss directory:

```
svn co https://opencmiss.svn.sourceforge.net/svnroot/opencmiss/cm/trunk cm
```

If you are not familar with subversion, have a look at http://svnbook.red-bean.com.

2.1.2 Obtain the Libraries

The openCMISS libraries repository is at http://www.physiome.ox.ac.uk/svn/opencmissextras/cm
To check out the main trunk of the various libraries required with openCMISS issue the following command in the opencmissextras directory:

```
svn co http://www.physiome.ox.ac.uk/svn/opencmissextras/cm/trunk/external/architecture cm/external/archit
```

where architecture is the appropriate architecture for the machine. Possible architectures are:

- i386-win32
- i386-win32-debug
- i686-linux
- i686-linux-debug
- x86_64-linux
- x86_64-linux-debug
- rs6000-32-aix
- rs6000-32-aix-debug

Currently, the svn repository for openCMISS libraries is down. An alternative location for the libraries within the ABI is on hpc. Go to \bioengsmb and copy the necessary files. The folder structure is the same as svn repository.

2.1.3 Makefile Structure

The top level makefile will eventually build a library. In the examples directory there are separate compilable "applications" with individual makefiles. However, the library stuff isn't there as we need to code the bindings.

2.2 Programmer documentation

This programmer documentation is written using DocBook with the source located in the opencmiss/cm/doc/programmer_documentation/ folder. The source XML code can be transformed into either chunked or combined documents in PDF or HTML format. The Makefile can be used to perform the transformation using make single or make chunk in the above folder. PDFs can be generated using make fo Doxygen can be included .

2.3 Project Setup

2.3.1 On AIX 5.3 (HPC)

2.3.1.1 Set environment

Set environment variable to point to openCMISS

```
setenv OPENCMISS_ROOT <path to your opencmiss folder>
```

or

```
export OPENCMISS_ROOT=<path to your opencmiss folder>
```

Set environment variable to point to openCMISS-extras

```
setenv OPENCMISSEXTRAS_ROOT <path to your opencmissextras folder>
```

or

```
export OPENCMISSEXTRAS_ROOT=<path to your opencmissextras folder>
```

2.3.1.2 Set MPI

Create a file called hostfile.list in your home directory. Inside the file, add several lines of "hpc.bioeng.auckland.ac.nz" In .rhost file in the home direcotry, add "hpc <username>"

2.3.1.3 Compile

Change directory to opencmiss/cm Change directory to examples/ Use gmake. This should result in a binary that you can run in the bin/rs6000-32-aix folder.

2.3.2 On Ubuntu 8.04

2.3.2.1 Set environment

Set environment variable to point to opencmiss

```
setenv OPENCMISS_ROOT <path to your opencmiss folder>
```

or

```
 export OPENCMISS_ROOT=<path to your opencmiss folder>
```

Set environment variable to point to openCMISS-extras

```
 setenv OPENCMISSEXTRAS_ROOT <path to your opencmissextras folder>
```

or

```
 export OPENCMISSEXTRAS_ROOT=<path to your opencmissextras folder>
```

It is also helpful to add the following

```
 setenv PATH ${OPENCMISSEXTRAS_ROOT}/cm/external/${archname}/bin:${PATH}
setenv PATH ${OPENCMISS_ROOT}/cm/bin/${archname}:${PATH}
```

where ${archname} is the appropriate architecture e.g., i686-linux, x86_64-linux. If you are using totalview you will also need to add

```
 setenv LM_LICENSE_FILES <path-to-the-flex-directory>
```

2.3.2.2 Install Compilers

Download Intel Fotran Compiler from here. Extract the file and follow the install.htm to install

2.3.2.3 Compile

To build an example project:

```
 make
```

To run the example project:

```
 mpd & mpirun -n 2 path/to/the/execution/file
```

To debug the project using TotalView:

```
 mpd & mpirun -tv 2 path/to/the/execution/file
```

2.3.3 On Windows XP (Visual Studio 2005)

2.3.3.1 Install Compilers

Download Intel Fortran Compiler from here. Execute the exe file and follow the installation wizard.

2.3.3.2 Install MPI

Download MPICH2 from here. You can either download the source archive and follow the README.windows file to install or download the installer to install. Set bin folder to the path To start the MPI, run

```
smpd -start
```

in command window. NOTE: as from MPICH2 version 1.0.7 the library names have changed. libmpich2 has now become libmpi!

2.3.3.3 Compile and Debug

Build the Fortran project under the debug mode and generate the opencmisstest-debug.exe file. In the C Project (since the Fortran projects do not support MPI cluster debugger), configure the debugging properties according to this. The MPIShim location is in the path similar to C: Files Visual Studio 8 Debugger\x86\mpishim.exe. Debug the C project.

2.3.3.4 Run

To run the project in the command window:

```
mpiexec -n 2 -localroot <path to the execution file>
```

2.3.4 On Windows Vista (Visual Studio 2008)

Install Compilers Download Intel Fortran Compiler from here. Execute the exe file and follow the installation wizard.

2.3.4.1

Before you install MPICH2 under Vista you must turn off User Account Control

1. Goto Start -> Control Panel
2. Double-click on User Accounts
3. Click "Turn User Account Control on or off"
4. Untick "Use User Account Control (UAC) to help protect your computer" and click OK
5. Restart your computer.

Download from here. Choose the Win32 IA32 (binary) option. Run the downloaded .msi file. Follow all instructions and install "For everybody". Once you have installed MPICH2 you can turn User Account Control back on. Follow the instructions above and in 4. tick the "Use User Account Control ...". NOTE: as from MPICH2 version 1.0.7 the library names have changed. libmpich2 has now become libmpi!

2.3.4.2 Compile

For each example, go into the VisualopenCMISS_08 folder. Double click the VisualopenCMISS project solution file to lauch Visual Studio.

2.4 Libraries Build (Optional)

2.4.1 Compiling PETSc

Note this is assuming you have the Intel Fortran compiler version 10.1.024. Adjust the version string as necessary.

2.4.1.1 Step1: Linux Environment installation and Compiler Environment Set up (For Windows only)

Under windows system:

- Install Cygwin if you need to. Cywin can be found here. Make sure you include the make and python modules when you install.
- Lauch a Command Prompt Window
- Run the ifortvars.bat batch file to setup your Intel Fortran environment. e.g., "C:\Program Files\Intel\Compiler\Fortran\10.1.024\IA32\Bin\ifortvars.bat"
- Run the Cygwin batch file to setup the unix environment e.g., "C:\Cygwin\Cygwin.bat"

N.B: PetSc uses X, so make sure in linux environment, libX11-dev package is installed. Also make sure blas and lapack (-dev) packages are installed.

2.4.1.2 Step2: Compile PETSC

For Windows

- Change to the opencmissextras PETSc directory e.g., if opencmissextras root is E: and we are compiling PETSC version petsc-2.3.3-p8 then "cd /cygwin/e/opencmissextras/cm/external/packages/PETSc/petsc-2.3.3-p8"
- If you have MPICH2 version 1.0.7 or greater edit the python/BuildSystem/config/packages/MPI.py file. Find the self.liblist_mpich line. After the line "['fmpich2.lib','mpich2.lib']," add the line "['fmpich2.lib','mpi.lib'],".
- `PETSC_DIR=/cygdrive/e/opencmissextras/cm/external/packages/PETSc/petsc-2.3.3-p8; export PETSC_DIR`
- For a debug install issue the following commands

  ```
   PETSC_ARCH=cygwin-c-debug; export PETSC_ARCH
  config/configure.py --prefix=/cygdrive/e/opencmissextras/cm/external/i386-win32-debug --with-shared=
  PETSC_ARCH=cygwin-c-debug; export PETSC_ARCH
  ```

  For a non-debug install issue the following commands

  ```
   PETSC_ARCH=cygwin-c-opt; export PETSC_ARCH
  config/configure.py --prefix=/cygdrive/e/opencmissextras/cm/external/i386-win32 --with-shared=no --w
  PETSC_ARCH=cygwin-c-opt; export PETSC_ARCH
  ```

- `make -e all`
- `make -e install`

For AIX5.3

- Change to the opencmissextras PETSc directory. e.g /people/tyu011/workspace/opencmissextras/cm/external/packages/P 2.3.3-p8
- setenv PETSC_DIR /people/tyu011/workspace/opencmissextras/cm/external/packages/PETSc/petsc-2.3.3-p8
- In config directory, copy the file aix5.1.0.0.py and rename it to aix5.3.0.0.py
- For a debug install issue the following commands

  ```
   setenv PETSC_ARCH aix5.3.0.0
  config/aix5.3.0.0.py --prefix=/people/tyu011/workspace/opencmissextras/cm/external/rs6000-32-aix-debu
  setenv PETSC_ARCH aix5.3.0.0
  ```

  For a non-debug install issue the following commands

  ```
   setenv PETSC_ARCH aix5.3.0.0
  config/aix5.3.0.0.py --prefix=/people/tyu011/workspace/opencmissextras/cm/external/rs6000-32-aix --wi
  setenv PETSC_ARCH aix5.3.0.0
  ```

- `make -e all`
- `make -e install`

Chapter 3

Simple Test

Normal text.

User defined paragraph:

Contents of the paragraph.

New paragraph under the same heading.

Note:

This note consists of two paragraphs. This is the first paragraph.

And this is the second paragraph.

See also:

FIELD_ROUTINES::FieldTypes,FIELD_ROUTINES

This module handles all field related routines.