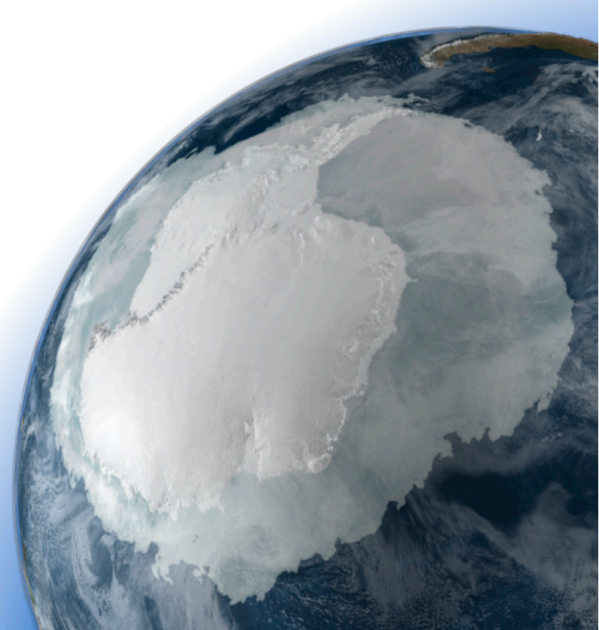
ISSM Workshop 2016

Ice Sheet System Model
Installation Guide

Gilberto PÉREZ[1]

[1] University of California, Irvine



June 2016

Outline

Preparing Your Environment

Before you can begin using ISSM it is necessary that you install the following:

- Terminal with an appropriate shell (BASH, Csh, Zsh)

Preparing Your Environment

Before you can begin using ISSM it is necessary that you install the following:

- Terminal with an appropriate shell (BASH, Csh, Zsh)
- MATLAB or Python

Preparing Your Environment

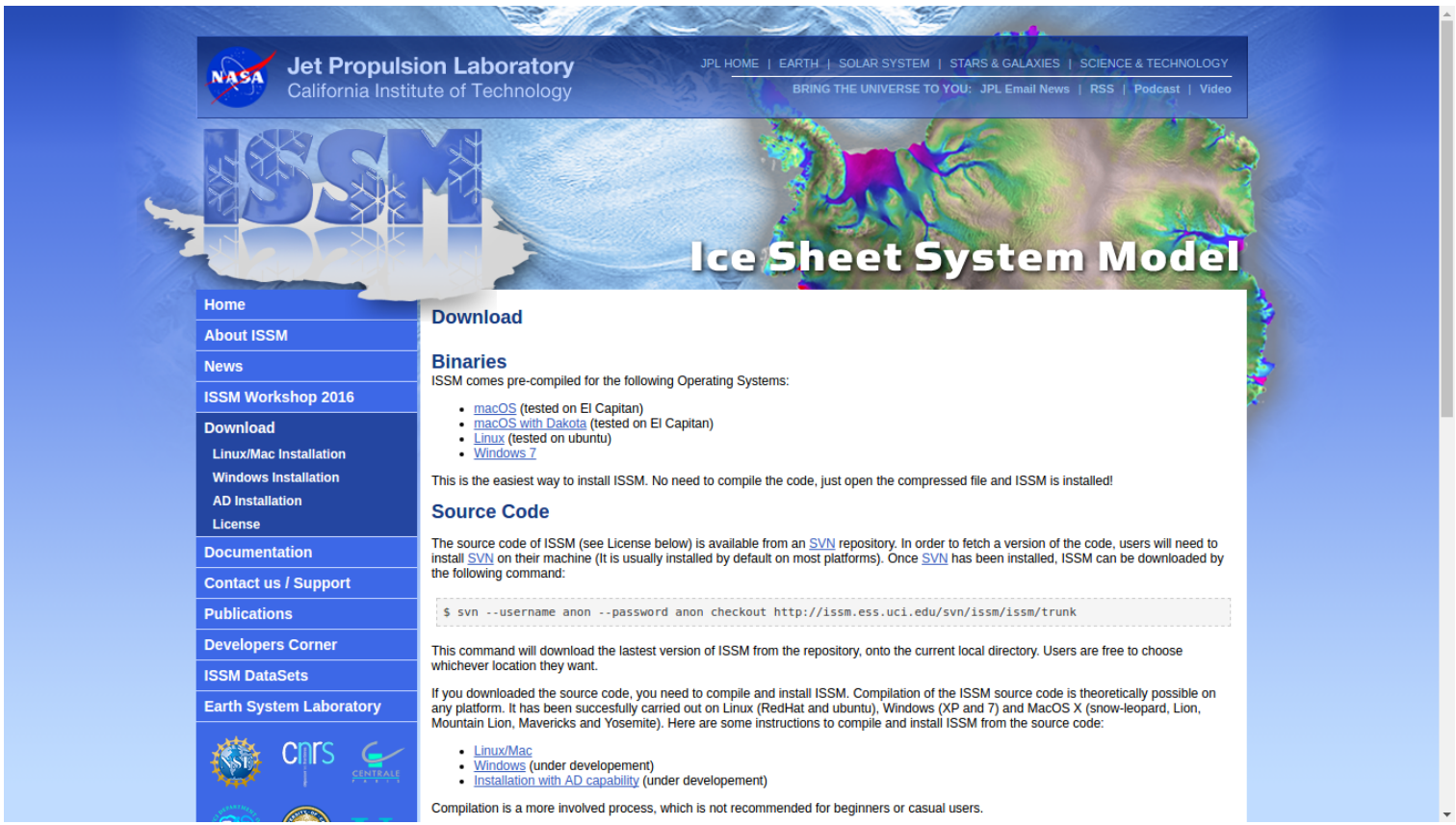
Before you can begin using ISSM it is necessary that you install the following:

- Terminal with an appropriate shell (BASH, Csh, Zsh)
- MATLAB or Python
- Compiler Toolchain(GCC, LLVM, MSVC)

Pre-compiled

The ISSM team maintains pre-compiled binaries that are easy to install and use. They can be found on our website:

`https://issm.jpl.nasa.gov/download/`



Pre-compiled

Advantages

- Quick and Easy!
- Easier for us to help you troubleshoot issues

Pre-compiled

Advantages

- Quick and Easy!
- Easier for us to help you troubleshoot issues

Disadvantages

- Much larger installation size than when compiling yourself
- Not optimized for your system
- Only available for MATLAB

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in
3. Set and export the `ISSM_DIR` to the trunk directory

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in
3. Set and export the `ISSM_DIR` to the trunk directory
4. For Linux, set and export `LD_LIBRARY_PATH` to `$ISSM_DIR/lib`

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in
3. Set and export the `ISSM_DIR` to the trunk directory
4. For Linux, set and export `LD_LIBRARY_PATH` to `$ISSM_DIR/lib`

You are now ready to start using ISSM! Follow these steps:

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in
3. Set and export the `ISSM_DIR` to the trunk directory
4. For Linux, set and export `LD_LIBRARY_PATH` to `$ISSM_DIR/lib`

You are now ready to start using ISSM! Follow these steps:

1. Launch MATLAB

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in
3. Set and export the `ISSM_DIR` to the trunk directory
4. For Linux, set and export `LD_LIBRARY_PATH` to `$ISSM_DIR/lib`

You are now ready to start using ISSM! Follow these steps:

1. Launch MATLAB
2. Addpath the `lib` and `bin` directories in `$ISSM_DIR`

Pre-compiled

To install the pre-compiled binaries follow these easy steps:

1. Download the appropriate tarball from the website
2. Create a directory for your ISSM installation and extract the contents there in
3. Set and export the `ISSM_DIR` to the trunk directory
4. For Linux, set and export `LD_LIBRARY_PATH` to `$ISSM_DIR/lib`

You are now ready to start using ISSM! Follow these steps:

1. Launch MATLAB
2. Addpath the `lib` and `bin` directories in `$ISSM_DIR`
3. Run `issmversion` to make sure everything was loaded correctly

Troubleshooting

Hopefully, you've all successfully installed and used ISSM, but things can go wrong. If so, try the following:

Troubleshooting

Hopefully, you've all successfully installed and used ISSM, but things can go wrong. If so, try the following:

- Your MATLAB version might be incompatible. Try upgrading it!
- Double check that your environment variables are set correctly
- Check that you added the correct directories when using addpath

Troubleshooting

Hopefully, you've all successfully installed and used ISSM, but things can go wrong. If so, try the following:

- Your MATLAB version might be incompatible. Try upgrading it!
- Double check that your environment variables are set correctly
- Check that you added the correct directories when using addpath

If you're still having issues you can always try compiling ISSM yourself!

Build ISSM Yourself

Building ISSM yourself is a considerably more involved process, but the benefits are numerous:

Build ISSM Yourself

Building ISSM yourself is a considerably more involved process, but the benefits are numerous:

- Optimized for your machine
- Configured with more or fewer features to meet your needs
- Choose what external packages and versions you want

Build ISSM Yourself

The ISSM website provides great guides for how to build ISSM for all 3 major operating systems:

- **Linux/MacOSX**: `https://issm.jpl.nasa.gov/download/unix/`
- **Win10**: `https://issm.jpl.nasa.gov/download/windows/`

Build ISSM Yourself

Your first step should be to checkout our code from our Subversion repository. This can be accomplished from the command line:

```
$ svn --username anon --password anon checkout http://issm.ess.uci.edu/svn/issm/issm/trunk
```

Change directory into `trunk`:

```
$ cd trunk
```

Now set and export `ISSM_DIR` and `MATLAB_DIR`:

```
$ export ISSM_DIR=$PWD
$ export MATLAB_DIR=/path/to/MATLAB
```

External Packages

Here is where compiling ISSM really gets interesting, as there are many different choices you can make. However, it is highly recommended that you stick to the basics before experimenting with more specialized builds. For Linux and MacOSX respectively the following external packages should be installed in the given order:

Linux
autotools install.sh
cmake install.sh
mpich: install-3.0-linux64.sh
petsc: install-3.7-linux64.sh
triangle: install-linux64.sh
m1qn3: install.sh

MacOSX
autotools: install.sh
cmake: install.sh
mpich: install-3.0-macosx64.sh
m1qn3: install.sh
petsc: install-3.6-macosx64.sh
triangle: install-macosx64.sh

External Packages Python

There are two ways to get a Python build working:

1. Through our external packages
2. Through a package management tool such as apt-get

If you choose to install through our external packages, then you will have to install the following extra external packages

```
python (recommended version: 2.7)
nose
blas
lapack
numpy
cython
scipy
hdf5
netcdf
netcdf-python
```

If you choose second options, then simply install Numpy, Scipy and Matplotlib.

External Packages

Installing any one external package is straight forward and can be repeated each time. The process is as follow:

```
$ cd $ISSM_DIR/autotools
$ ./install.sh
```

At this point it is recommended that you source our environment script. This is actually not required after every external package, but it is easier to do it every time.

```
$ source $ISSM_DIR/etc/environment.sh
```

Configuring

Once you have finished installing all of the external packages it is time to configure ISSM. Like the external packages there are many different ways to configure ISSM, but the following configuration is considered standard for MacOSX:

MacOSX

```
./configure \
--prefix=$ISSM_DIR \
--with-matlab-dir="$MATLAB_DIR" \
--with-triangle-dir="$ISSM_DIR/externalpackages/triangle/install" \
--with-mpi-include="$ISSM_DIR/externalpackages/mpich/install/include" \
--with-mpi-libflags="-L$ISSM_DIR/externalpackages/mpich/install/lib/ -lmpich" \
--with-petsc-dir="$ISSM_DIR/externalpackages/petsc/install" \
--with-metis-dir="$ISSM_DIR/externalpackages/petsc/install" \
--with-blas-lapack-dir="$ISSM_DIR/externalpackages/petsc/install" \
--with-scalapack-dir="$ISSM_DIR/externalpackages/petsc/install/" \
--with-mumps-dir="$ISSM_DIR/externalpackages/petsc/install/" \
--with-m1qn3-dir="$ISSM_DIR/externalpackages/m1qn3/install" \
--with-numthreads=2
```

Configure

Similarly for Linux:

Linux

```
./configure \
--prefix="$ISSM_DIR" \
--with-triangle-dir="$ISSM_DIR/externalpackages/triangle/install" \
--with-python-dir=/usr\
--with-python-numpy-dir=/usr/lib/python2.7/dist-packages/numpy\
--with-mpi-include="$ISSM_DIR/externalpackages/mpich/install/include" \
--with-mpi-libflags="-L$ISSM_DIR/externalpackages/mpich/install/lib -lpmpich -lm
--with-petsc-dir="$ISSM_DIR/externalpackages/petsc/install" \
--with-scalapack-dir="$ISSM_DIR/externalpackages/petsc/install/" \
--with-mumps-dir="$ISSM_DIR/externalpackages/petsc/install/" \
--with-metis-dir="$ISSM_DIR/externalpackages/petsc/install/" \
--with-m1qn3-dir="$ISSM_DIR/externalpackages/m1qn3/install" \
--with-numthreads=2
```

Installation

Finally, it is time to preconfigure, run our configure script, make and make install:

```
$ autoreconf -ivf
$ make
$ make install
```

If all goes well, then you have successfully installed ISSM.

Troubleshooting

If something went wrong with your build please check the following things:

- You installed all the necessary external packages
- The directories you provided in your configuration script match the directories where you installed everything
- You sourced our environment script after every external package you installed
- You used the same compilers throughout the process

If you are still unable to get your build to work please checkout our forum. There are many useful threads there and it is the easiest way to get help with your build.

`https://issm.ess.uci.edu/forum/`