

Run the code

- [Run the code via Make It So!](#)
 - [Usage](#)
 - [FAQ](#)
 - [How do I restart or continue a long run?](#)
 - [How do I turn on HDF5 file support for MFLFM on stampede and yellowstone?](#)
 - [Extending MakeltSo](#)
 - [Create Config File \(recommended\)](#)
 - [Modify at Run Time](#)

Run the code via Make It So!

The easiest way to configure models coupled to the LFM is via the script generator affectionately named "Make It So". MakeltSo requires Python ≥ 2.4 .

Usage

Once you've set the corresponding [environment variables](#), learn more about MakeltSo via the `--help` flag:

```
makeItSo.py --help
```

There are two ways to run the script generator:

1. Interactive mode: At the command-line, type

```
/path/to/makeItSo.py
```

and follow the prompts. You will be asked to set a series of parameters. Press `<enter>` for the default value or `? <enter>` for more information on a particular variable. Once you've entered all model parameters & environment setup, MakeltSo will create input files & job scripts necessary to run the code.

1. Preconfigured run mode: Each time you run the script generator, it writes a `makeItSo.config` file at:

```
Output directory/RUN_IDENTIFIER_makeItSo.config
```

Once you've run the script generator via interactive mode (see above), open up a `*makeItSo.config` file with a text editor. You can edit run parameters, save the file and configure a new run using this simple text file. Execute:

```
/path/to/makeItSo.py -c /path/to/RUN_IDENTIFIER_makeItSo.config
```

MakeltSo will create all the input files & job scripts necessary to run the code.

Make It So will print instructions and create a set of files:

- Input files that set model parameters
- Shell script to configure the models (`RUN_IDENTIFIER-setup.sh`)
- Run scripts to execute the models (`RUN_IDENTIFIER-001.*`)

Follow the instructions displayed by Make It So to run the code.

FAQ

How do I restart or continue a long run?

Say you ran MakeltSo to generate a 2 hour run of LFM. You should have several files in your run directory, including:

- `LFM-001.sh`: 50 minute spinup interval
- `LFM-002.sh`: Job script for 1st hour of run
- `LFM-003.sh`: Job script for 2nd hour of run
- `SW-SM-DAT`: 6 hours of [solar wind data from OMNI](#)
- `LFM_makeItSo.config`: configuration file for MakeltSo.

You execute the job scripts and your 2 hour run (plus 50 minute spinup) have completed. Let's say you want to restart the run and continue for an additional 2 hours. There are two ways to do this:

1. Extend the run (recommended)
 - a. Edit `LFM_makeItSo.config` and set the stop time 2 hours later
 - b. Re-run `MakeltSo`:

```
makeItSo.py -c LFM_makeItSo.config
```

- c. Submit the next job script in the iteration: `LFM-004.sh`. This will pick up where the previous job script left off and continue executing the model.
2. Configure a restart run (advanced)
 - a. Edit `LFM_makeltSo.config` and
 - i. change start time to 2 hours later
 - ii. change stop time to 2 hours later
 - iii. add the parameter `SPINUP: false` to Shared model settings
 - b. Re-run `MakeltSo`:

```
makeItSo.py -c LFM_makeItSo.config
```

This will create the following run scripts:

- `{LFM-001.sh}`: Job script for 3rd hour of run
 - `{LFM-002.sh}`: Job script for 4th hour of run
- Note: if you ran `MakeltSo` in the original run directory, you may have overwritten the pre-existing job scripts!

How do I turn on HDF5 file support for MFLFM on stampede and yellowstone?

1. **Stampede**
 - a. First, the `hdf5` module needs to be loaded before MFLFM can be built.

```
module load hdf5
```

- b. Build MFLFM as normal.
 - c. To turn on HDF5 support, `makeltSo.py` needs to be run in expert mode. (Warning: This may prompt you to enter dozens of model parameters). Not using expert mode will default back to HDF4.

```
makeItSo.py --expert
```

- d. When the model parameter option "`IO_EXTENSIONS(hdf):`" is reached, enter "`hdf5`". Entering "`hdf`", or just hitting return will default back to HDF4.

```
IO_EXTENSION ( hdf ): hdf5
```

- e. HDF5 file support for MFLFM is now turned on. Run MFLFM as normal, there should be `.hdf5` files in the run directory.

2. **Yellowstone**
 - a. On yellowstone, the `hdf5` module doesn't need to be loaded. Follow the Stampede directions starting at 1b.

3. More details on changes to help add HDF5 support on other machines.

- a. Changes to `env/Makefile.${MACHINE}`
 - i. Stampede changes to `env/Makefile.${MACHINE}`
 1. Add `HDFnmpi` path to HDF5 libraries.

```
HDF5nmpi= /opt/apps/intel13/hdf5/1.8.9
```

2. There was also issues with finding the `szip` library, so `EXTRA_LIBS` needs to be set to find `szip`.

```
EXTRA_LIBS = -L/opt/apps/intel13/hdf5/1.8.9/lib -lsz
```

- ii. Yellowstone changes to `env/Makefile.${MACHINE}`
 1. Add `HDFnmpi` path to HDF5 libraries.

```
HDF5nmpi = /glade/apps/opt/hdf5/1.8.11/intel/12.1.5
```

- b. There may also need to be changes to the machines environment. On yellowstone, there was no changes needed from the default machine environment. On stampede, the hdf5 module needed to be loaded. On other systems, the hdf5 and szip modules might need to be loaded, if they exist.
- c. Switching back and forth between HDF4 and HDF5 file support is possible by running makeltSo.py --expert and specifying a different IO_EXTENSION. The input files will need to be either HDF4 or HDF5, you cannot mix HDF4 input files with HDF5 output files, and vice versa. Running makeltSo.py without --expert will default to HDF4.

Extending MakeltSo



Note

Adding a new environment to MakeltSo is an advanced topic. Do not hesitate to ask for help with initial configuration.

We have tested MakeltSo on a [variety of systems](#). You can customize for your particular machine in two ways:

Create Config File (recommended)

You can permanently add your environment by defining all parameters in a **QUEING-ENVIRONMENT.yml** file. This file must reside in the `MakeItSo/Environments` subdirectory. It describes the computer (MPI installation, flags for the number of processors, etc) and any queue information that is required. We recommend that you start with a pre-existing environment & extend it to your own environment. For example, see `LSF.yml` or `SGE.yml`.

At runtime, MakeltSo will locate this YAML file & provide an interface to the user to set up the environment.

Modify at Run Time

You can modify a pre-configured environment at run-time. Start makeltSo.py in "expert" mode (Warning: This may prompt you to enter *dozens* of model parameters)

```
misc/python/makeItSo.py --expert
```

When prompted for an environment, select the one most similar to yours. If you are working on a Linux desktop computer, select "Standard Linux Bourne shell without a queue". You will be asked to enter a series of parameters related to your compute environment. You must know the settings used to compile the LTR prerequisites.