

CyberShake Install and Auto-submission

1.Prepare Data:

To run the install script, the Models must be under certain Folder and structure

```
Cybershake
  version
    Data
      Sources
        FaultName
          Srf
            FaultName_REL01-S1244.srf
            FaultName_REL02-S1254.srf
            FaultName_REL01-S1244.info
            FaultName_REL02-S1254.info
          Stoch
            FaultName_REL01-S1244.stoch
            FaultName_REL02-S1254.stoch
          VMs
            FaultName
              vs3dfile.s
              vp3dfile.p
              rho3dfile.d
              params_vel.yaml
              model_coords_rt01-h0.400
              model_params_rt01-h0.400
        Runs
```

1a. Creating Rupture Models(SRF/Stoch)

To create SRFs using NHM, please follow [Create SRFs for all realisations of the considered faults](#)

To create SRFs manually, please follow [How to create Rupture Model Manually](#)

IMPORTANT: if SRFs are created manually, and the name are not exactly the same as corresponding VM, the name can only contain '_HYP*number*' otherwise the automated workflow may crash (this will be resolved later on)
for example : AlpineF2K_HYP01-02

1b. Creating Velocity Models

To create VMs using NHM, please follow [Generate velocity models for the considered faults](#)

To create VMs manually, please follow [How to create a Velocity Model Manually](#)

Note: if any models are generated manually, please make sure all files are in the same structure shown in step one.

2. Install

After files are in place, run the install script

THREE arguments needed:

1. the root folder, which contains the Data and Runs folder.
2. the file that contains a list of VMs
3. the version of gmsim template

```
$gmsim/workflow/scripts/cybershake/install_cybershake.py $gmsim/RunFolder/cybershake/v18p5 $gmsim/RunFolder/cybershake/v18p5/nhm_selection_file 16.1 --seed 0
```

The 2rd argument should be the NHM selection file you used to create the VM/SRF models.

something like this:

```
Opotiki02 1r
Opotiki03 1r
OpouaweUruti 10r
Orakeikorako 10r
Orakonui 12r
Oruakukuru 12r
Oruawharo 12r
Otakiri 12r
Otaraiia 12r
OtokoTotoF7 12r
```

the `--seed` is optional. (0 : random)
if not provided, it will use a random number.

3. Create a screen socket

Running scripts on a screen socket and avoid the need of having the terminal open constantly (which means you can disconnect from Kupe but have the script still running on it)

```
screen -S your_prefered_name_for_socket
```

To **detach** a socket, use Ctrl+A+D

To **Terminate** a socket, use Ctrl+D

to show all available socket created before, use `--list`

```
screen -list
There is a screen on:
  289787.cybershake_v18p6      (Detached)
1 Socket in /var/run/usbcreens/S-ykh22.
```

To resume to a specific socket, use `-r`

```
screen -r 289787.cybershake_v18p6
or
screen -r 289787
```

4. Run the simulation in auto

run the auto submission script with an period/interval.

The script takes **Two** arguments, 1st is the path to `sim_root` folder (which is the same as you passed to install script), 2nd is the user name

please note: the arguments should be the path on Kupe, not your local machine.

Important: please run this script on Maui.

1. Run the `queue_monitor.py`. this scripts monitors the the pipe and update `mgmt_db`

```
$gmsim/workflow/scripts/cybershake/queue_monitor.py $nobackup/Runfolder/Cybershake/v19p5
```

Note: this script should always be alive and running in the background, as long as `auto_submit.py` is still running.

2. Run the `auto_submit.py`

```
$gmsim/workflow/scripts/cybershake/auto_submit.py $nobackup/RunFolder/Cybershake/v19p5 `whoami`
```

the whoami is to get current username

Note: this script should only be ran after queue_monitor.py is running.

the old wrapper that used to be able to run two steps into one are currently broken, this would be addressed soon.

5. Monitor Simulation Status

Monitor the status of each simulation by running query script.

```
python $gmsim/workflow/scripts/management/query_mgmt_db.py $nobackup/RunFolder/cybershake/v19p5
```

it should show you something like this:

run_name	process	status	job-id	last_modified
2012p075555	merge_ts	in-queue	2198889	2018-05-29 04:34:39
2012p075555	winbin_aio	created	None	2018-05-29 04:34:39
2012p075555	BB	created	None	2018-05-29 04:34:39
2012p075555	IM_calculation	created	None	2018-05-29 04:34:39
2012p075555	HF	completed	2198881	2018-05-29 21:29:21
2012p075555	EMOD3D	failed	2198858	2018-05-29 04:43:40
2012p713691	merge_ts	created	None	2018-05-29 04:34:40
2012p713691	winbin_aio	created	None	2018-05-29 04:34:40
2012p713691	BB	created	None	2018-05-29 04:34:40
2012p713691	IM_calculation	created	None	2018-05-29 04:34:40
2012p713691	HF	completed	2198882	2018-05-29 21:29:21
2012p713691	EMOD3D	failed	2198860	2018-05-29 04:44:49
2012p764736	merge_ts	created	None	2018-05-29 04:34:40
2012p764736	winbin_aio	created	None	2018-05-29 04:34:40
2012p764736	HF	created	None	2018-05-29 04:34:40
2012p764736	BB	created	None	2018-05-29 04:34:40
2012p764736	IM_calculation	created	None	2018-05-29 04:34:40
2012p764736	EMOD3D	failed	2198862	2018-05-29 04:44:49
2012p781523	merge_ts	created	None	2018-05-29 04:34:40
2012p781523	winbin_aio	created	None	2018-05-29 04:34:40
2012p781523	BB	created	None	2018-05-29 04:34:40

use -e to show only the failed runs(with the errors)

```
python $gmsim/workflow/scripts/management/query_mgmt_db.py /nesi/nobackup/nesi00213/test_auto_submit -e
```

```
Run_name: 2012p075555
Process: EMOD3D
Status: failed
Job-ID: 2198858
Last_Modified: 2018-05-29 04:43:40
Error: Task removed from squeue without completion
```

```
Run_name: 2012p713691
Process: EMOD3D
Status: failed
Job-ID: 2198860
Last_Modified: 2018-05-29 04:44:49
Error: Task removed from squeue without completion
```