Generate a run script

First step

There are several pre defined experiments available in the run-directory, listed as `exp.<experiment-name>`. To generate a run script for one of these experiments do

```
./make_runscripts <experiment-name> -r <script-directory>
```

Be careful not to forget script-directory and experiment-name. make_runscripts will generate **all** possible runscripts otherwise. make_runscripts only works after configuring the model, as it needs the proper information of the compiler used.

The resulting runscript `exp.<experiment-name>.run`, for some experiments also a postprocessing script `post.<experiment-name>.run`, will be stored in the run-directory and might need additional editing.

Use your experimenter ID

In general and to simplify collaboration it is an established way to copy a given script according to our **personalized naming convention**, see List of IDs.

Adaptations for accounting and parallelization

Example: Run the NextGEMS_R2B8 script as developer abc on Levante:

Copy the exp.NextGEMS_R2B8 experiment-template to your personal experiment-id:

```
cp run/exp.NextGEMS_R2B8 run/exp.abc0001
./make_runscripts abc0001 -r run
cd run
```

Edit the file exp.abc0001.run and change the model settings (account e.g. mh0287):

```
#SBATCH --account=mpiaes
to
#SBATCH --account=mh0287
```

```
#SBATCH --nodes=1
to
#SBATCH --nodes=80
```

```
export OMP_NUM_THREADS=$((${SLURM_JOB_CPUS_PER_NODE%%\(*} / 4 / 1))
to
export OMP_NUM_THREADS=4
```

Submit the job:

```
sbatch exp.abc0001.run
```

You can check the status of your jobs with:

```
squeue -u <user>
```

Results will be written to `icon-mpim/experiments/abc0001`

In case you want to run the full 10 days of this experiment you have to adapt the run time in the run script: Set the run time of the job to 3 hours

```
#SBATCH --time=03:00:00
```

— *Monika Esch 2023/05/08 13:44*