GUTTA — Operational manual

Last update: 2021/09/30 — **Author:** Fabio Viola <fabio.viola@cmcc.it>

How to read this document
How to start GUTTA
    Automatic start
    Manual run
        Manual run of a part of the chain
Checking the status of a run
    Structure of the software
    Checking GUTTA
        Is GUTTA running right now?
        Why a job is pending?
        Is there any log I can read?
Telegram notifications

How to read this document

All the instructions reported in this document refer to the user `visir-dev` on Zeus, so remember to login with this username before running each command:

```
$ ssh visir-dev@zeus.cmcc.scc
```

How to start GUTTA

Automatic start

GUTTA is started automatically thanks to crontab. We have two daily run at 2 am and 6 pm UTC (so 4 am and 8 pm local time):

```
# full visir

00 04 * * * /work/opa/visir-dev/operational_scripts/runVisir_full.sh $(date +"\%Y\%m\%
d_\%H") > /work/opa/visir-dev/operational_scripts/logs/runVisir_$(date +"\%Y\%m\%d-\%H
\%M").log &
```

```
00 20 * * * /work/opa/visir-dev/operational_scripts/runVisir_full.sh $(date +"\%Y\%m\%
d_\%H") > /work/opa/visir-dev/operational_scripts/logs/runVisir_$(date +"\%Y\%m\%d-\%H
\%M").log &
```

NOTE: since Zeus is configured with local time, at the moment of switching from Daylight saving time to standard time we will have to manually modify this in order to remain at 2 am UTC and 6 pm UTC). In 2021, the switch will take place on 31st October.

Manual run

Of course, if something goes wrong, or for test purposes, we may need to manually start GUTTA. In that case, we need to perform the following operations:

1. Kill existing jobs with `bkill 0`
2. Copy one of the previous commands (and remove **all** the backslashes in the `date` command)

That's it.

Manual run of a part of the chain

As you can see from the content of `runVisir_full.sh`, the chain is composed by six components (described later on). If you need to start (for example) just the last four pieces, copy the script, comment the first two pieces and remove the job dependency in the first uncommented job... Then you can run it!

Checking the status of a run

Structure of the software

Before understanding how to check if GUTTA is running and if everything is ok, we need to understand how the whole chain is done.

GUTTA is composed by a set of components that we hereby report in the right execution order:

1. **Campi:** it's a python software made by Lorenzo. It is located in `$VISIR/Campi`

2. **Tracce:** it's another python software in the suite built by Lorenzo, and it's located in `$VISIR/Tracce`. In order to speed up the execution of the job, `Tracce` is executed 30 times, one for each route, instead of having a single run that sequentially process all the routes. To run the 30 jobs there is an LSF *job array:* this means that all the 30 jobs will appear with the same job id. The reason for this is to ease the check made by the next script on the status of Tracce. Referring to job dependencies, we also have to say that `Tracce` starts only after the **successful** ending of `Campi`
3. **Visual**: this is the last piece of Lorenzo's code. It's in `$VISIR/Visualizzazioni` and runs only after a **successful** completion of `Tracce` (have a look at `-w` switch). Even this piece of software is executed as a job array.
4. **csv2shape**: this software, written by Giuseppe, is currently outside VISIR-2 package. Also in this case it's a job array of 30 jobs (one per route). It starts after the **successful** completion on `Visual`.
5. **copyN08**: after all the processing made on Zeus, results must be copied to n08. This is the purpose of this script that runs only after the **successful** completion of the job array `csv2shape`.
6. **runn08**: this is the last piece of the chain and it simply connects to n08 to invoke a php script that is the last processing step. It starts only if `copyN08` completes **successfully**.

We can see how to check the status of GUTTA...

Checking GUTTA

Is GUTTA running right now?

Simple! Just ask `bjobs`:

```
$ bjobs
```

If no job is running or in the queue we will read: `No unfinished job found`

If at least a job is running or queued for execution, we will see something like:

```
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
427534  visir-d RUN   s_medium   login1-ib   n198-ib     GUTTA_n08  Sep 29 12:21
```

Then, we can read the component name from the column `JOB_NAME` and see its status from the column `STAT`. It can be `RUN` if it's running right now, or `PENDING` if it's waiting for it's turn.

Why a job is pending?

Standing to Occam's razor, we can say *"because it's not yet its turn"*. But there may be another reason... Take as an example the job `Tracce`. It will start after a **successful** run of `Campi`. But what happens if this job fails? The condition to run `Tracce` will never be satisfied, and it will be pending forever. In order to check if we are in this unlucky situation, just run:

```
$ bjdepinfo JOB_ID
```

where `JOB_ID` is the ID of the JOB that is still pending.

Is there any log I can read?

Yes, of course. Logs are all stored in the directory `/work/opa/visir-dev/operational_scripts/logs`. For simplicity, a bash variable named `$LOGS` is defined, so you just need to:

```
$ cd $LOGS
```

and then you are in the logs folder. For each component you will find a `.log` / `.out` and a `.err` file. In the filename you will find the datetime in the format `YYYYmmdd-HHMM` and the job id. This will help you find easily all the logs of a specific run. For example, to see all the logs produced by the second run of 2021/09/29, just type:

```
$ find $LOGS -iname \*20210929-2200\*
```

Telegram notifications

Notifications can be read in the channel GUTTA_dev. It only write messages if any of the error log files contains at least a line.