## How to manage a job from start to end

If the directory **DRM4G_DIR** (`~/.drm4g` by default) does not exist, `drm4g` will create one with for its local configuration

1. Start up DRM4G :

```
[user@mycomputer~]$ drm4g start
Creating a DRM4G local configuration in '/home/user/.drm4g'
Creating '/home/user/.drm4g/var/acct' directory
Coping from '/home/user/drm4g/etc' to '/home/user/.drm4g/etc'
Starting DRM4G ....
OK
Starting ssh-agent ...
OK
```

2. Show information about all available resources and their host :

```
[user@mycomputer~]$ drm4g resource list
RESOURCE            STATE
localmachine        enabled

[user@mycomputer~]$ drm4g host list
HID ARCH       JOBS(R/T) LRMS       HOST
0   x86_64           0/0 fork       localmachine

[user@mycomputer~]$ drm4g host list 0
HID ARCH       JOBS(R/T) LRMS       HOST
0   x86_64           0/0 fork       localmachine

QUEUENAME      JOBS(R/T) WALLT CPUT  MAXR  MAXQ
default              0/0 0     0     1     1
```

3. Create a job template :

```
[user@mycomputer~]$ echo "EXECUTABLE=/bin/date" > date.job
```

4. Submit the job :

```
[user@mycomputer~]$ drm4g job submit date.job
ID: 0
```

5. Check the evolution of the job :

```
[user@mycomputer~]$ drm4g job list 0
JID DM   EM   START    END      EXEC    XFER    EXIT NAME            HOST
0   pend ---- 19:39:09 --:--:-- 0:00:00 0:00:00 --   date.job        --
```

If you execute successive `drm4g job list 0`, you will see the different states of this job:

```
0   pend ---- 19:39:09 --:--:-- 0:00:00 0:00:00 --   date.job        --
0   prol ---- 19:39:09 --:--:-- 0:00:00 0:00:00 --   date.job        --
0   wrap pend 19:39:09 --:--:-- 0:00:00 0:00:00 --   date.job localhost/fork
0   wrap actv 19:39:09 --:--:-- 0:00:05 0:00:00 --   date.job localhost/fork
0   epil ---- 19:39:09 --:--:-- 0:00:10 0:00:00 --   date.job localhost/fork
0   done ---- 19:39:09 19:39:27 0:00:10 0:00:01 0    date.job localhost/fork
```

- `pend`: The job is waiting for a resource to run on.
- `prol` :The remote system is being prepared for execution.
- `wrap pend` :The job has been successfully submitted to the computing resource and it is waiting.

- `wrap actv`:The job is being executed by the computing resource.
- `epil`:The job is finalizing.
- `done`:The job has finished.

6. Results are standard output (stdout) and standard error (stderr), both files will be in the same directory as the job template:

```
[user@mycomputer~]$ cat stdout.0
Mon Jul 28 12:29:43 CEST 2014

[user@mycomputer~]$ cat stderr.0
```