The OfficeGRID Executor Documentation

A feature normally present in standard computational grids is a batch job system, which makes it possible to remotely distribute and execute job scripts. The OfficeGRID Executor can be used to submit batch jobs to a large number of OfficeGRID machines. A batch job is simply a shell script (i.e. a `.BAT` file under Windows and probably a `bash` or `tcsh` script under Linux) containing a set of commands you want to run on some set of machines. With the OfficeGRID Executor it is possible to target a specific machine for the execution of a given batch job or a set of machines that all execute the job.

This document describes how the OfficeGRID Executor is used.

Setup of The OfficeGRID Executor

In order to execute batch jobs an executor must be started on all the machines on which batch jobs should be executed. The executor is called **`og_executor`** and it can be called from the command line interface with the command:

```
og_batchexe [options]
```

where the options can be any of the following:

`-h`, **`--help`**

Displays the help message.

`-t`, **`--tag <tag string>`**

The machine running the executor can be identified by this string. This can be used if one wish to run a batch job on a subset of the machines that participate in the OfficeGRID.

`-p`, **`--tmp <path string>`**

The directory where the temporary job script files will be saved.

`-q`, **`--queue <queue name>`**

The queue name where jobs are taken from. If this option is not used the default queue will be used.

`-c`, **`--concurrent <concurrent jobs>`**

The number of concurrent jobs from **`<queue name>`**. If this option is not used the default number of concurrent jobs will be 1.

An example of how the executor can be started could be:

```
og_batchexe --tag linux -p /tmp
```

where the tag is "**linux**" (i.e. this executor is probably running on a Linux machine) and the temporary directory is "**/tmp**". It is also possible to start the executors via the **og_start** command (see The OfficeGRID Installation and Management Documentation).

The queue option describes which job queue the batch jobs should be taken from. I.e. it is e.g. possible to have a queue that holds jobs where the running time is short and a queue that holds jobs where the running time is long.

It is also possible to set the number of jobs that are allowed to run simultaneously on the machine. This is done be setting the **-c** option.

Submit a Job

The command **og_job** is used when batch jobs are submitted to the OfficeGRID. **og_job** is used in the following way:

```
og_job [options] <job script> [job script arguments]
```

where the **<job script>** is the name of the batch job script to be submitted. If the batch job is submitted to a group of Windows machines, running the OfficeGRID Executor (**og_executor**), the script is typically a .BAT file and if the job is submitted to Linux machines it is probably a bash or tcsh script. However any other kind of scripts that can be executed on the machines running the og_executor program can be submitted.

The options given to og_job can be any of the following:

-h, **--help**

Displays the help message.

-g, **--group <group flag>**

A flag that can be used to identify the job group to which the job belongs.

-f, **--flag <job flag>**

A flag that can be used to identify the job.

-c, **--copies <num. copies>**

The number of copies of the job.

-t, **--tag <tag string>**

Is used to find a specific machine that matches the given tag. The machine tag is set as an option to the job executor og_executor.

-n, **--host <host name>**

Is used to find a specific machine that has the given host name.

-q, **--queue <queue name>**

The queue **<queue name>** where the job is submitted to. If this option is not used the default queue will be used.

-i, **--id**

A unique job id is written to **stdout** when the job is submitted.

The size of the batch job is currently limited to 16 KB, i.e. it is not possible to submit batch jobs of size greater that 16 KB.

A group flag can be used to group related jobs, but there is also a secrecy feature to group flags. I.e. after jobs with the same group flag have been submitted the results can only be retrieved (see the section below on job retrieval) if the job group is specified. Likewise jobs with an associated group flag can only be listed (see section below) and removed (see section below) if the group flag is known/specified. By default batch jobs are placed in the **default** group if the group flag is not specified.

An example of how a batch job can be submitted could be:

```
og_job -g my_job_group -f job01 -t linux -i hello.sh Bob
```

where the group, the job is placed in, is **my_job_group**, the job flag is **job01** and the tag is "**linux**". I.e. the batch job is executed on a machine that runs **og_executor -t linux**. Also note the "**-i**" option which results in the unique job id beeing printed to **stdout** by the **og_job** command. This id can later be used to find the job or result of the job. The job script **hello.sh** is given the argument "**Bob**".

Submit a Command Line Job

The command **og_shcmd** is used to submit one-line batch jobs to OfficeGRID. **og_shcmd** is used in the following way:

```
og_shcmd [options] [shell] <command line>
```

where the **<command line>** argument is the command line string to be submitted. The options given to **og_shcmd** are the same as the options for **og_job**. The shell must be specified to one of the following:

bat

BAT shell for Windows execution.

bash

bash for Linux execution.

tcsh

tcsh for Linux execution.

An example of how a command line batch job can be submitted could be:

```
og_shcmd -g my_job_group -f job01 -t linux bash echo Hello World
```

where the group, the job is placed in, is **`my_job_group`**, the job flag is **`job01`** and the tag is "**`linux`**". I.e. the batch job is executed on a machine that runs **`og_batchexe -t linux`**. This job would execute "**`echo Hello World`**" on the OfficeGRID.

List Submitted Jobs

After one or more jobs have been submitted to OfficeGRID they can be listed with the command **`og_joblist`**. Submitted jobs are listed with the command:

```
og_joblist [options]
```

`og_joblist` can be given the following options:

`-h`, **`--help`**

Displays the help message.

`-g`, **`--group <group flag>`**

A flag that can be used to identify the job group to be listed.

`-f`, **`--flag <job flag>`**

A flag that can be used to identify a job in a job group. This flag may only be used if the job group is specified with the **`-g`** or **`--group`** option.

`-q`, **`--queue <queue name>`**

The queue **`<queue name>`** to list job from. If this option is not used the default queue will be shown.

`-e`, **`--extended`**

List all information available about the jobs. This will give a display of width 118 characters.

The following is an example of how a job list can be extracted:

```
og_joblist -g my_job_group -q fast
```

where the job group to be listed is **my_job_group** and the jobs are located in the **fast** queue. The following columns are listed is the **-e** option is not used:

User

The user that submitted the job.

ID

The unique job ID.

Script

The name of the script to be executed

Submit Time

A time stamp giving the time and date when the job was submitted.

However when the **-e** is used the following columns will also be printed:

Len

The length/size of the job script.

Tag

The job tag - used to find a specific machine that matches the given tag

Host

The host targeted to execute the job.

Remove Submitted Jobs

If batch jobs have been submitted with the **og_job** command, but have not been executed by a **og_executor** they can be removed with the command **og_jobrm**. The **og_jobrm** command is executed in the following way:

```
og_jobrm [options] (-g | --group) <group flag>
```

The options can be any of the following:

-h, **--help**

Displays the help message.

-g, **--group <group flag>**

A flag that can be used to identify the group of jobs to be removed.

-f, **--flag <job flag>**

A flag that can be used to identify a submitted job.

-i, **--id <job id>**

A job id that can be used to identify a submitted job.

-q, **--queue <queue name>**

The queue **<queue name>** where the job is removed from. If this option is not used the default queue will be used.

For example all the jobs in the job group **my_job_group**, from the default queue, can be removed by issuing the command:

```
og_jobrm -g my_job_group
```

The group flag is mandetory in order to identify the job (or group of jobs, if the job flag is not specified) to be removed. If the job was not given a group flag, when it was submitted, the group flag is set to "**default**" and the job is removed by parsing "**-g default**" to the **og_jobrm** command.

List Job Results

After jobs have been executed on machines running the **og_executor**, the results from the executions can be listed by using the **og_joblist** command. I.e. a result list is generated by the command:

```
og_joblist [options]
```

where the options can be any of the following:

-h, **--help**

Displays the help message.

-g, **--group <group flag>**

A flag that can be used to identify the group of results, from submitted jobs, to be listed.

-f, **--flag <job flag>**

A flag that can be used to identify a result in a result group. This flag may only be used if the result group is specified with the **-g** or **--group** option.

-q, **--queue <queue name>**

The queue **`<queue name>`** to list job from. If this option is not used the default queue will be shown.

`-e`, **`--extended`**

List all information available about the jobs. This will give a display of width 129 characters.

The following is an example of how a result list can be extracted:

```
og_reslist -g my_job_group
```

where the result group to be listed is **`my_job_group`**. When the **`-e`** option is not used the printed job information will be:

User

The user that submitted the job.

ID

The unique job ID.

Script

The name of the script to be executed.

Exit Code

The exit code of the job script.

Submit Time

A time stamp giving the time and date when the job was submitted.

Finish Time

A time stamp giving the time and date when the job finished.

However when the **`-e`** is used the following columns will also be printed:

stdout Len

The length/size of the stdout from the job script.

stderr Len

The length/size of the stdout from the job script.

List Jobs and Results

The command **og_lists** is actually a simple alias for calling both **og_joblist** and **og_reslist**. It can be used to get an overview of waiting and finished jobs in the OfficeGRID system. The parameters for **og_lists** are the same as for **og_joblist** and **og_reslist**, which are described in the sections above.

Retrieve a Job Result

After a job has been executed the result - consisting of the **stdout**, **stderr** output and exit code - can be found by running the **og_res** command:

```
og_res [options] (-o | <stdout file>) (-e | <stderr file>)
```

where the options can be any of the following:

-h, **--help**

Displays the help message.

-q, **--queue <queue name>**

The queue **<queue name>** where the job was submitted to. If this option is not used the default queue will be used.

-g, **--group <group flag>**

The group in which the result of the wanted job is placed.

-f, **--flag <job flag>**

The flag identifies the result in a result group.

-o, **--stdout**

Instead of writing **stdout** of the job result to the file **<stdout>**, redirect it to the local **stdout**.

-e, **--stderr**

Instead of writing **stderr** of the job result to the file **<stderr>**, redirect it to the local **stderr**.

-b, **--block**

og_res will not return before a result has become available.

-i, **--id**

The job id of the wanted job.

Please note that after a result is retrieved with the command **og_res** it is removed from the OfficeGRID system. The following is an example of how the **og_res** command could be called:

```
og_res -q fast -g my_job_group -f job01 -o job01.err
```

In this example the result of the job from the queue **`fast`**, with job group **`my_job_group`** and job flag **`job01`** is written to the local **`stdout`** and the file **`job01.err`**.

Distributing files with OfficeGRID for Windows

The command **`og_filecopy`** copies files from one computer to other computers running the OfficeGRID Executor for Windows using a shared Windows drive. The command requires a configuration file that specifies defaults for the parameters it needs. This configuration file must be a textfile specifying the following parameters:

`FILE`

The file that should be copied (including its path). Multiple files can be specified using wildcards.

`SHARE`

Path to a directory on the shared drive that can be used for temporary storage while copying the file.

`DEST_HOSTS`

A comma separated list of host names of the computers, which the files should be copied to.

`DEST_DIRS`

A comma separated list of paths specifying a target directory for each of the computers, which the files should be copied to.

`DEST_DIR`

Can be used instead of `DEST_DIRS` if the same path should be used as target directory on all the computers, which the files should be copied to.

`PUBLIC`

Can be set to **`TRUE`** if the files to be copied are already on a publically available shared drive, meaning that it is not necessary for **`og_filecopy`** to copy the files to the shared drive, before distributing them to the target computers. If the files are not already on a shared drive this option must be set to **`FALSE`**.

An example of a configuration file is shown below:

```
FILE = C:\Projects\distribute_files\*.*
SHARE = E:\share_dir
DEST_DIRS = C:\Projects\pc1_dest, C:\Projects\pc2_dest, C:\Projects\pc3_dest
DEST_HOSTS = pc1, pc2, pc3
```

```
PUBLIC = FALSE
```

If this configuration file is used all the files in **C:\Projects\distribute_files\**, on the computer where **og_filecopy** is invoked, will be copied to **C:\Projects\pc1_dest\** on the computer named **pc1**, to **C:\Projects\pc2_dest\** on the computer named **pc2**, and to **C:\Projects\pc3_dest\** on the computer named **pc3**.

og_filecopy is used in the following way:

```
og_filecopy [options] <config file>
```

where the options can be any of the following:

/h, **-h**, **--help**

Displays the help message.

/f, **-f**, **--file**

The file that should be copied (including its path). Multiple files can be specified using wildcards.

/s, **-s**, **--share**

Path to location on the shared drive that can be used for temporary storage while copying the file.

/c, **-c**, **--hosts**

A space separated list of host names of the computers, which the files should be copied to.

/t, **-t**, **--destination**

A space separated list of paths specifying a target directory for each of the computers, which the files should be copied to. If only one path is given it will be used for all the target computers.

/p, **-p**, **--public**

Must be set to **TRUE** if the files to be copied are already on a publically available shared drive, meaning that it is not necessary for **og_filecopy** to copy the files to the shared drive, before distributing them to the target computers. If the files are not already on a shared drive this option must be set to **FALSE**.

/A

Is passed directly on to the standard DOS copy commands that are used for doing the actual copying.

/B

Is passed directly on to the standard DOS copy commands that are used for doing the actual copying.

/D

Is passed directly on to the standard DOS copy commands that are used for doing the actual copying.

/N

Is passed directly on to the standard DOS copy commands that are used for doing the actual copying.

/V

Is passed directly on to the standard DOS copy commands that are used for doing the actual copying.

/Z

Is passed directly on to the standard DOS copy commands that are used for doing the actual copying.

The following is an example of how the **og_filecopy** command could be called:

```
og_filecopy /A --file c:\new_files\*.dat og_filecopy.cfg
```