

cexecms is a tool to execute a command, script, or parset file on the cluster nodes for each file (or directory) matching the given file name pattern. The command or file can contain placeholders like <FN> (see help info below), which are substituted by the actual file name.

cexecms must be executed on the head node of a cluster with the proper environment set (via e.g. 'use LofIm'). It copies the settings of important environment variables (like PATH and LD_LIBRARY_PATH) to the cluster nodes, so the proper environment is used when the given command or script is executed on the remote nodes.

For example:

```
use LofIm
cexecms -s dp.parset DPPP '/data/L427806/L427806_SAP000_SB*_uv.MS'
```

executes DPPP with the given parset file on the cluster nodes for each MeasurementSet matching the last argument. The parset file could look like:

```
msin=<FN>
msout=/data/scratch/<BN.>_flag.MS
step=aoflagger
```

For each matching MeasurementSet a temporary parset file is created with the appropriate substitutions. When run with the -d option (dry-run) on the head node of CEP2 (lhn001), the output looks like below showing the actual names.

```
----- locus001 -----
Dryrun: DPPP /home/diepen/nd.parset-diepen-18596
msin=/data/L427806/L427806_SAP000_SB000_uv.MS/
msout=/data/scratch/L427806_SAP000_SB000_uv_flag.MS
steps=[aoflagger]
```

Hereafter some help info is given, that can also be obtained by giving the cexecms command without arguments or with -h.

cexecms runs a command or script on cluster nodes for files matching the given file name glob pattern. Placeholders in the command or script are replaced by the actual file name parts.

The current environment (paths, etc.) is written to an env file that is sourced on the cluster nodes. So you should have done "use LofIm" if you need LofIm in the (remote) command.

usage:

```
cexecms [-c cluster] [-d] [-s script] [-w workdir] [-e envdir]
        command nameglob [arg1 arg2 ...]
```

```
-c cluster    Cluster name as defined for cexec.
               default is lce:    if run on an lfe node
                               test:  if run on lce072
                               locus: otherwise
-d           Do a dryrun.
               (do not execute, but only print the command/script)
```

`-i ids` List of ids to replace <ID> in the nameglob argument. A comma and/or blank can be used as separator. Ids can be given with parset range style (e.g. 33678..33683)

In this way the command can be executed for multiple observations.

`-s script` Script or parset file to be used by the command. It must be accessible on all nodes in the cluster. Placeholders (like <FN>) in the file are replaced. It will be used as the first argument in the command.

`-w workdir` Working directory in remote process. Default is login directory.

`-e envdir` Directory for the env file. Default is \$HOME. It must be visible for all remote nodes.

`command` Command to be executed remotely. Quotes are needed if it contains spaces, etc. Placeholders (like <FN>) in the command are replaced.

`nameglob` File name glob pattern to find matching files # can be used as a shorthand for [0-9]. E.g., one can use SB### meaning any subband. The pattern can contain the placeholder <ID> as explained above in the -i option.

`arg1 arg2 ..` Optional extra arguments to be given to the command.

Using cexec, the script cexecms-part is executed on the given cluster nodes. It looks for files matching the given file name glob pattern. The glob pattern can be bash-style, thus *, ?, [], and {} are possible. The given command will be executed for each matching file name. The actual file name can be made part of the command or script using one or more of the following placeholders:

<FILENAME> or <FN> for the full file name
 <BASENAME> or <BN> for the basename part
 <DIRNAME> or <DN> for the directory part

The first two can be followed by a . (e.g. <FN.>) meaning that the basename is used till the first dot (thus the extension is removed.) Similarly, <.BN> gives the extension (thus after the first dot).

For standard LOFAR file names the following placeholders can also be used:

<OBSID> for the obsid part of <BN.> (till first _)
 <SAP> for the subarray pointing part of <BN.> (till next _)
 <SB> for the subband part of <BN.> (till next _)
 <TYPE> for the dataset type part of <BN.> (after last _)

If -s is given, the command is executed like:

command script arg1 arg2 ..

Otherwise, if no substitutions have been done the command is executed like:

command <FN> arg1 arg2 ...

Otherwise like

command arg1 arg2...

For example:

```
cexecms "ls -d" "/data/scratch/pipeline/L2011_22663/*"
```

is a trivial example and could also be done with cexec. Note that

```
cexecms "ls -d <FN>" "/data/scratch/pipeline/L2011_22663/*"
```

is doing the same.

The following example is more elaborate and creates a _sel.MS table in another directory for each MS of subband 000 till 099.

```
cexecms "taql 'select from <FN> where ANTENNA1 in [0,1,2]
        giving /data/scratch/diepen/<BN.>_sel.MS'"
        "/data/scratch/pipeline/L2011_22663/*SB0##*"
```

Note that quotes have to be used abundantly.

Also note that (t)csH requires a ! to be escaped with a backslash.

Sometimes a command can be dangerous or take a long time to run.

In such a case it makes sense to first do a dry-run execution with the -d option.



See the [note on X11 forwarding](#).

From:
<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:user_software:documentation:cexecms&rev=1481788742

Last update: **2016-12-15 07:59**

