MARS command and request syntax

MARS command

The mars command can accept a request file as a parameter or read the MARS request from standard input.

The mars command issued on the command line will output a message welcoming you to the system. At this point, one can type a mars request, then send an end-of-file to MARS by typing <CTRL>-d. As this method is error prone it is not recommend except for test purposes.

The recommended way to use MARS is by passing a request file, which can be created as a 'here document', to the mars command, e.g.

```
cat > my_request <<EOF
retrieve,
   type  =an,
   date  = -1,
   target="my_an"
EOF

mars my_request
```

Directives can also be passed into mars from the input stream:

```
mars <<EOF
retrieve,
   type  =an,
   date  = -1,
   target="my_an"
EOF
```

All data retrievals should be submitted as (ecgate) batch jobs .

MARS can be called with the command line options listed in the table below. Most of these options are particularly useful when problems arise.

MARS command line options

Option	Meaning
-d	switches debug information on (same as MARS_DEBUG=1)
-h	prints help information
-p	to use the previous MARS binary. If you suspect a new version of MARS may yield different results (including errors, crashing, etc...) this is the first thing to try. If the previous version of MARS works, then contact the CallDesk to report the problem you've found.

In addition, the behaviour of MARS can be changed via various environment variables:

MARS environment variables

Environment variable	Usage
MARS_COMPUTE_FLAG	if set to 0, the model identification number is preserved. When using compute, GRIB headers of resulting fields are changed to reflect fields have been created by MARS. In particular, model identification number is set to 255. With this env. variable, the model identification of the first field is copied onto the result.
MARS_DEBUG	if set to a value different from 0, additional output will be generated

MARS_MULTITARGET_STRICT_FORMAT	if set to 1, 'consistent' file naming will be applied for multi targets, see Data storage keywords
MIR_DEBUG	If MIR_DEBUG **and** MARS_DEBUG are set to 1, debug information for interpolation with MIR will be generated.

Request syntax

A MARS request is the way to specify an action on a set of fields or observations. The directives specified in a MARS request have the following syntax:

```
verb,
   keyword1 = value 1,
   ...      = ...,
   keywordN = value N
```

Where:

- verb identifies the action to be taken, e.g. retrieve or list.
- keyword is a known MARS variable, e.g. type or date
- value is the value assigned to the keyword, e.g. Analysis or temperature

Notes on MARS request syntax:

- Leading spaces and tab characters are ignored.
- verbs and directives keyword=value have to be terminated with a comma, except the last one, which MARS considers as the end of a request.
- Directives may be in upper or lower case or a mixture of both; MARS converts everything except file names to upper case.
- Characters * ! and # can be used to comment until end-of-line.
- Values may be predefined names, numeric values or user-supplied character strings such as file names.
- Abbreviations may be used for verbs, keywords and values, provided enough letters are given to uniquely identify what is meant.
- MARS has acronyms for the names of different parameters which can be specified by users.
- Keywords may be assigned a single value, a list of values or a range of values. A list is indicated by the separator / ; a range is indicated by using the keywords to as well as / and by. Examples of different formats for values are given in the table below.

Examples of MARS keyword assignments

Format	Example
single value	param = temperature/SSRD
list of values	step = 12/24/48
range of values	date = 19990104/to/19990110/by/2

The number of fields described in a MARS request is computed by multiplying the number of values of its identification keywords after the expansion of all ranges. This number, or the value specified by expect, is used by MARS to decide whether the request has been fully satisfied or to fail otherwise.

Certain keywords accept all as a valid value, e.g. param = all. In this case, MARS will deliver data available matching the rest of the request. Please note that if all is assigned to a keyword, MARS will no longer check on missing values. Furthermore, if all is assigned, MARS can retrieve data from the archive only, not from the Fields Data Bases.

Most keywords accept off as a valid value. In this case, the keyword is not considered by MARS and will be removed from the request before processing it.

The following example shows a retrieve request with directives grouped according to their meaning:

```
retrieve                          action
  class=od
  stream   = oper,                identification of archive
  expver   = 1,
  date     = -1,                  date and time related
  type     = analysis,
  levtype  = model levels,
  levelist = 1/to/91,
  param    = temperature,
  grid     = 2.5/2.5,             post-processing
  area     = Europe,
  use      = infrequent,          execution control
  target   = analysis.12          storage
```

Retrieve requests have to specify, at least, directives to identify the data and directives for storage of the retrieved data. The ones to identify fields can be found in the MARS Catalogue. Data manipulation (post-processing) directives and execution control directives are optional, depending on your needs.

With the main focus on data retrievals, this User Guide covers the following MARS actions, or verbs:

retrieve to extract data from MARS or from a source file. Most of this documentation will cover the usage of the retrieve verb.

compute will perform mathematical operations on retrieved grid point fields in GRIB format, see MARS compute.

list queries the MARS archive about the data availability for a given request, without actually retrieving the data. It is suitable for interactive or batch use to check availability of certain datasets and its collocation in the archive, see MARS list.

read can be used to filter or manipulate data already retrieved into a file.

write is mainly used to save data from fieldsets, a temporary storage, into files, see keyword fieldsets.

Multiple requests in one call to MARS

> When accessing MARS via the Web API client, only one target file per MARS call is permitted.

MARS can accept more than one request in the same call. This allows you e.g. to do further computations than the ones provided by the standard post-processing directives before writing the data into a target file and it is how the compute verb can be used. The following concepts have to be taken into account:

Keyword inheritance

When two or more retrieve requests are used in a single call to MARS, keywords not explicitly set in the second request are inherited from the first (previous). The following example retrieves 12 UTC model level and 1000 hPa pressure level temperature for yesterday's analysis:

```
retrieve,
   class  = od,
   type   = analysis,
   stream = oper,
   date   = -1,
   time   = 12,
   levtype= model levels,
   level  = 1/to/91,
   param  = temperature,
   target = analysis.ml

retrieve,
   levtype = pl,
   levelist= 1000,
   target  = analysis.pl
```

There can be cases in which unwanted keywords are inherited. They can be removed from subsequent requests by assigning the value off.

Append to target file

It is possible to store data from different requests into the same target file. Note that MARS will append the result of each retrieve request at the end of the target file. The following example retrieves first some Analysis data from yesterday. Then, a second retrieval will extract the same data from 40 days ago and append it at the end of the (same) target file.

```
retrieve,
   class  = od,
   type   = analysis,
   stream = oper,
   expver = 1,
   date   = -1,
   time   = 12,
   levtype= pressure levels,
   levelis= 1000/850/500,
   param  = temperature,
   target = "analysis.grb"
retrieve,
   date   = -40
```

Manipulate already retrieved data

> This feature is not available when accessing MARS via the Web API.

Read requests can be used to filter or manipulate data from already retrieved files. It reads the data from a Unix file specified by the keyword source. This data can either be stored in a fieldset for further manipulation or written to a Unix file specified with the keyword target. Consider for example that file myfile contains certain meteorological parameters at all pressure levels. One can extract level 1000 hPa only and interpolate those fields to a 2.5x2.5 lat/lon grid, using the following MARS request:

```
read,
   source   = myfile,
   levelist = 1000,
   grid     = 2.5/2.5,
   target   = only_1000
```

The verb retrieve can also be used to extract data from a Unix source file, perform post-processing on the result and store it into a target Unix file. In this case, retrieve needs all necessary directives as if it were a retrieval from the archives. Here is an example, assuming we want to extract geopotential and temperature Analyses for the 1st of January at 12 UTC:

```
retrieve,
   source   = myfile,
   class    = od,
   type     = an,
   date     = 20010101,
   time     = 12,
   param    = z/t,
   levtype  = pl,
   levelist = 1000,
   grid     = 2.5/2.5,
   target   = only_1000
```

The advantage of retrieve is that it will fail if the requested data is not present in the source file. A disadvantage is that the request has to specify all the directives, and you might not know what is the exact contents of the source file.