

MARS User Guide

`mars@ecmwf.int`

January 2006, ECMWF

Table of Contents

1. Introduction.....	3
2. Meteorological Archive	4
2.1. Products	4
2.2. Data formats.....	9
3. Language.....	10
3.1. Request format.....	10
3.2. Retrieve request	10
3.3. Identify the Archive	11
3.4. Date and time directives	13
3.5. Directives for fields	14
3.6. Directives for observations and satellite images.....	16
3.7. Data Storage directives	17
3.8. Post-processing directives.....	17
3.9. Execution control directives	19
3.10. Considerations on Parameter values	20
3.11. Language hints for ECMWF fields.....	20
3.12. Language hints for ECMWF observations	24
4. Architecture	29
4.1. Clients	29
4.2. Servers	29
4.3. Interaction client/server: request execution	30
5. Retrieving data.....	31
5.1. Request scheduling	31
5.2. Post-processing	32
5.3. Retrieval efficiency	33
5.4. Trouble-shooting.....	34
5.5. Data deficiencies in the archive	36
5.6. Frequently Asked Questions	37
6. Advanced use.....	40
6.1. Listing archive contents	40
6.2. Manipulate already retrieved data.....	41
6.3. Retrieve data into multiple targets	41
6.4. Multiple requests in one call to MARS	42
6.5. Fieldset.....	43
6.6. Compute.....	44
7.	

1 Introduction

MARS is ECMWF's Meteorological Archival and Retrieval System. This documentation describes facilities for the retrieval of data from ECMWF's operational and other archives, including the on-line Fields Data Base (FDB).

There are a number of factors differentiating MARS from other Archive systems around the world:

- Facilities to Archive and Retrieve meteorological data, enabling very easy access to the Archives via a **pseudo-meteorological language**
- Different requirements imposed by ECMWF's Operational and Research environments. They force MARS to be a **24 hour service** with as very little down-time as possible
- Batch and interactive modes have to be supported
- **Large amount of data**, both in size and number of items stored
- **Large number of users** with different requirements. Some users retrieve large datasets rarely, whereas others retrieve few fields very often.
- Heterogeneous environment, involving Supercomputers, user workstations and remote systems all interconnected with a variety of network technology
- MARS holds Terabytes of data, mainly using **GRIB** format for Meteorological fields and **BUFR** format for Meteorological Observations. Most of the data produced at ECMWF daily is archived in MARS, and therefore available to users.

This documentation starts with an overview of **ECMWF's activity**, as it is necessary to know the kind of data available from a meteorological archive. The most common products are listed, followed by details on the data formats used.

It follows with the pseudo-meteorological vocabulary, i.e., the **MARS language**. It introduces the MARS request format and syntax. Then, it focuses in the structure of **retrieve requests**, together with the list of all directives and possible values. The most common products are reviewed, providing the MARS terms needed to name each data set.

Later on, the **MARS architecture** is presented: what are the clients and servers. This enables users to understand the request execution of a MARS request: what is happening behind the scenes from the user's point of view.

Within all the above concepts, the documentation focuses on aspects to bear in mind while **retrieving data from MARS**, e.g. request scheduling in the main archive, post-processing capabilities available, hints to maximise retrieval efficiency and trouble shooting guidelines.

Finally, a section is devoted to **advanced usage**, e.g., considerations on multiple requests in a single call to MARS, how to organise the retrieved data in target files, and other operations available from MARS like list or compute.

2 Meteorological Archive

Every day, ECMWF produces a **global Analysis and Forecast** to ten days ahead and archives them in MARS.

The first ECMWF numerical model was a grid-point model with 15 levels in the vertical and a horizontal resolution of 1.875 degrees in latitude and longitude, corresponding to a grid length of 200 kilometers.

A number of major changes have occurred since ECMWF's activity started:

- In **April 1983**, the grid-point model was replaced by a T63 spectral model (i.e. a spectral representation in the horizontal with a triangular truncation at wave-number 63). The number of levels in the vertical was increased to 16.
- In **May 1985**, the spectral truncation was extended to wave-number 106.
- The number of levels was increased to 19 in **1986**.
- In **September 1991**, a much higher resolution spectral model was put into operation. The spectral truncation was extended to wave-number 213 and the number of levels was increased to 31.
- In **1992**, a wave model used for ocean wave forecasting (the WAM model) became operational, followed after a few months by a Mediterranean implementation.
- In **November 1997**, 4DVar became operational.
- In **April 1998**, the spectral truncation was extended to wave-number 319.
- In **June 1998**, the atmospheric model and the wave model were coupled in order to take advantage of the boundary conditions they represent to each other.
- In **March 1999**, the number of levels was increased to 50.
- In **October 1999**, the number of levels was increased to 60.
- In **September 2000**, 4DVar cycling was increased to 12 hours. No more First Guess fields are archived.
- In **November 2000**, the spectral truncation was extended to wave number 511.
- In **June 2004**, the Early Delivery Forecasting System was implemented, comprising two main 6-hour 4D-Var analysis and forecast cycles for 00 and 12 UTC and two additional 12-hour 4D-Var analysis and first guess forecast cycles

A more [comprehensive history](#) of the evolution of ECMWF's operational forecasting system can be found on the **Data Services information site**.

In addition, ECMWF runs an **Ensemble Prediction System** which explores the concept of probabilistic forecasting by creating perturbations on the initial conditions, then running a forecast for each of the perturbations.

Other projects run on a regular basis, such as **Seasonal Forecast**, **Multi-Analysis Ensemble**, **Monthly Forecast** and **Sensitivity forecast**, or have run in the past, for example the ECMWF **Re-Analysis**. All their outputs are available in MARS.

The majority of **Observations** used as model input are also available in MARS.

2.1 Products

In order to know the data available from MARS, users need to be familiar with ECMWF's activities. Products listed here are meant as an introduction to these activities. Users wanting

to learn more are encouraged to visit the [User Guide to ECMWF Products](#), Meteorological Bulletin M 3.2

Operational data produced daily at ECMWF

Atmospheric deterministic models

- **Analysis:** global analyses for the four main synoptic hours 00, 06, 12 and 18 UTC. They are the best gridded estimate of the state of the atmosphere (best fit to observations). For each of the synoptic hours, data is produced at the following levels:
 - [Surface](#), these fields represent the meteorology at the surface.
 - [Model levels](#), used in ECMWF's forecast model to resolve the atmosphere in the vertical.
 - [Pressure levels](#), computed by the model from its Model Levels.
 - [Isentropic levels](#), either potential vorticity or potential temperature.
- **Forecast:** global 10-day forecast based on the 12 UTC Analysis. Again, forecast products are also classified in the same level types: [Surface](#), [Model levels](#), [Pressure levels](#), and [Isentropic levels](#). Meteorological parameters are written output for every forecast time step, 3-hourly intervals from 00 to 72 hours, and 6-hourly from 72 to 240 hours.
- **First guess:** a forecast with base time from the previous synoptic hour and a forecast time step of (usually) 6 hours. Note that since the change to 12 hour-cycling 4-dimensional variational Analysis in the year 2000, this type of data has been discontinued.
- **Initialised Analysis:** the best 'balanced' gridded estimate of the state of the atmosphere (initial state of the forecast). With the implementation of the three dimensional variational Analysis (3D-var) on January 1996, the Initialised Analysis is no longer produced, and the Analysis is the best and the best 'balanced' gridded estimate of the atmosphere.
- **Four-dimensional variational Analysis:** analyses using observations over a time window (e.g., a 12 hour 4DVar at cycle *hh* will contain observations from *hh-08:59* to *hh+03:00*).
- **Errors in Analysis:** the assumed uncertainty of an observation is combined with the assumed uncertainty of the First Guess, resulting in an estimate of the total uncertainty in the Analysis
- **Errors in First Guess:** result of the uncertainty of the First Guess compared with observations
- **Four-dimensional variational Analysis increments:** the low resolution increment which is added to the first-guess after each inner loop minimisation.

Wave deterministic models

Since 1998, ECMWF's deterministic atmospheric model is coupled with a wave model.

- [Analysis](#), global analyses of the surface of the oceans for the four synoptic hours.
- [Forecast](#), global 10 day forecast based on 12 UTC Analysis.

ECMWF runs the WAM model at a finer resolution for a limited area. It used to be called the Mediterranean model but is now known as the **European Wave Model**. This model is run to 5 days and produces [Analysis](#) and [Forecast](#).

Ensemble Prediction System

ECMWF's Ensemble Prediction System has a coupled atmospheric and wave model

- **Control forecast**, an unperturbed forecast at a lower resolution than the main 10-day forecast. Forecast runs to 21 days. Data is available at [Surface](#), [Model](#) and [Pressure](#) levels.

- **Perturbed forecasts**, different forecasts to 10 days with perturbed initial conditions. They are numbered from 1 to N depending on the EPS setup. Data is available at [Surface](#) and [Pressure](#) levels.
- **Forecast probabilities**, a statistical distribution of the weather parameters from all ensemble members is used to produce probabilistic weather forecasts.
- **Forecast mean probabilities**, are means of the above over certain periods of time.
- **Ensemble means**, are means of the ensemble forecast members.
- **Clusters**, similar ensemble members are grouped together into clusters. The mean and standard deviation of these clusters are computed (as well as the mean and standard deviation of the overall ensemble). Five sets of clusters are computed, one for the entire European area, and four for smaller areas.
- **Tubes**, use the tube method for clustering ensemble forecast members.

System for severe weather prediction

On March 2001, an experimental second run of the entire forecasting system (deterministic and Ensemble Prediction System) was setup, based on 00 UTC Analysis. This is now part of the operational system.

Boundary conditions

Also known as **short cut-off**. Four additional Analyses are run for 00, 06, 12 and 18 UTC with a cut-off time of 4 hours, followed by global 4-day forecasts to provide some Member States with boundary conditions for their limited area models. All Analyses data but only the Forecast from 00 UTC are archived.

Multi-Analysis Ensemble

Every day ECMWF receives Analyses from four centres, **National Center for Environment and Prediction**, **The Met Office**, **Meteo-France** and **Deutscher Wetterdienst**, and runs 5 forecasts, 1 based on each different analyses plus one compound of all the analyses (**consensus**) including ECMWF's Analysis.

Seasonal Forecast

ECMWF started an experimental programme for seasonal prediction in 1995, which attempts to predict seasonal changes by coupling three models: **atmospheric**, **wave** and **ocean** models. Learn more on [ECMWF's Seasonal forecast](#) system.

Monthly Forecast

With the same setup as the seasonal forecast, this project runs every week producing forecasts to 32 days, again, coupling **atmospheric**, **wave** and **ocean** models. Learn more on [ECMWF's Monthly forecast](#) system.

Monthly and climatology datasets

ECMWF maintains an archive of monthly means data from the [atmospheric](#) and [wave](#) model archive. The resolution and internal representation of the archive may change according to changes in ECMWF's operational practice.

Atmospheric Analysis monthly means are averaged over the calendar month for each of the synoptic times 00, 06, 12 and 18 UTC.

The **atmospheric Forecast monthly means** are the average of all the forecasts for a particular step that verify in the selected calendar month. All monthly means are archived at model resolution.

Wave Analysis monthly means are averaged over the calendar month for each of the synoptic times 00, 06, 12 and 18 UTC.

Wave forecast monthly means are the average of all the forecasts for a particular step that verify in the selected calendar month. Wave Monthly means are archived at 1.5 degree resolution.

There is a limited climatology data set, which contains the Geopotential and Temperature on Pressure Levels for each month of the year. The data originally came from NCEP, Washington, and it was processed to store it in MARS. Although it is available to users, it is recommended that modern applications use the Monthly means archive or the Re-Analysis Monthly means archive for climatology purposes.

Special Projects

ECMWF Re-Analysis

The [ECMWF Re-Analysis](#) (ERA) project has produced two datasets:

- ERA-15, a validated 15 year data set of assimilated data for the period **1979 to 1994**.
- ERA-40, a validated 45 year data set of assimilated data for the period 1957 to 2002.

These available datasets include: analysis, forecast and forecast accumulations as output from atmospheric model, as well as analysis and forecast from a wave model re-analysis. There is also a Monthly Means data sets contain data at the resolution of the data assimilation and forecast system used by each Re-Analysis.

DEMETER

It is the acronym of the EU-funded project named [Development of a European Multi model Ensemble system for seasonal to inter annual prediction](#). The objective of the project is to develop a well-validated European coupled **multi-model ensemble forecast** system for reliable seasonal to inter annual prediction. Six comprehensive European global coupled atmosphere-ocean models are being installed at ECMWF, those of: **ECMWF, Météo-France, LODYC, Met Office, MPI, INGV, INM-HIRLAM** and **CERFACS**.

PROVOST

Provost stands for [Prediction Of climate Variations On Seasonal to inter annual Time scales](#).

They are a set of experiments from four centres: **ECMWF, MétéoFrance, EDF** and **Met Office**. The experiments are 120 day runs from 9 consecutive starting days, with write-ups every 24 hours of Pressure level and Surface data.

ECSN-HIRETYCS

ECSN is the European Climate Support Network. HIRETYCS is the [High Resolution Ten Year Climate Simulation](#).

This data set consist of 10-year climate simulation produced at three centres: Centre National de Recherches Météorologiques (**CNRM**), Max Planck Institute (**MPI**) and United Kingdom **Met Office**.

TOST

TOST is the THORPEX Observing System Test, an experiment carried out at the end of 2003 to evaluate targeted observations in an Ensemble Prediction System. There is output from 3 different models, **ECMWF**, **Meteo-France** and United Kingdom **Met Office**.

Research or Member State's experiments

A vast amount of data is archived daily containing experiments produced by ECMWF's Research Department or experiments run by Member States at ECMWF. Basically, an experiment can address any of the areas of Meteorology and it is archived accordingly.

Users wanting to retrieve Research experiments need to know in advance the **name of the specific experiment and its nature**. For this information, please, contact [User Support](#).

Input data to ECMWF models or for verification/scores

Fields

ECMWF archives a selection of products, both **Analysis** and **Forecast**, from other Centres: **Bracknell, Melbourne, Montreal, Offenbach, Toulouse, Tokyo** and **Washington**. Most of these products exist on the GTS and they are combined to create global fields with **ECMWF local GRIB header extensions** added. A convention exists to create empty (dummy) fields when there are missing data (e.g. if problems arise at one of the Centres or on the GTS)

There are other fields which are used as input for the ECMWF forecasting system (e.g. **Sea Surface Temperatures from Washington** which are used by ECMWF's Analysis).

Images from Meteosat and GOES Satellites are coded in GRIB and archived in MARS

Observations

Observations used as input to the assimilation system are also archived in MARS, as well as the **Feedback** on how the observations are used in the analysis. There are, amongst others:

- **Surface data**, conventional and satellite data covering: synoptic observations, soil and earth temperature, buoy information and PAOB observations (pseudo surface pressure observations in the Southern Hemisphere).
- **Vertical soundings**, conventional and satellite data: PILOT, TEMP, ROCOB, RTOVS, ATOVS, SATEM.
- **Upper-air data**, conventional observations such as AIREP and ACARS, together with satellite upper-air SATOB, high resolution winds and geostationary radiances.

MARS also holds the observations in the form of files as they are presented to the Analysis (**Analysis Input**), as well as the feedback files (**Analysis Feedback**). These are mainly used to allow to reproduce in the future any past operational run. There is one set of files per synoptic time.

2.2 Data formats

Archived data is stored in two WMO formats:

WMO FM 92-IX Ext GRIB (GRId in Binary)

WMO defined format for meteorological field data, or (more generally) any regularly spaced gridded data. All ECMWF model output is in GRIB format with ECMWF local extensions in their headers. GRIB data is archived in one of the following spatial coordinate systems:

- **Spherical Harmonics (SH)**, mainly for upper air fields
- **Gaussian Grid (GG)**, mainly for surface data, although some upper air fields as well
- **Latitude/Longitude (LL)**, other centre's data, wave and ocean data

The correspondence between the three types of grid resolutions is as follows:

Spectral	Gaussian	Latitude/Longitude
T63	N48	1.875
T106	N80	1.125
T159	N80	1.125
T213	N160	0.5625
T319	N160	0.5625
T511	N256	0.351
T639	N320	0.28125
T799	N400	0.225

The GRIB format is handled via the [GRIBEX](#) subroutine, part of the [libemos library](#) at ECMWF.

BUFR (Binary Universal Form Representation)

WMO defined format for point data (irregularly spaced). Archived observations are in BUFR format.

The BUFR format is handled via the [BUFRDC](#) subroutine, part of the [libemos library](#) at ECMWF.

3 Language

A good approach to the MARS language is to consider that **each field or observation in the archive has a name**. The purpose of this section is to show how to name uniquely each individual field using the MARS vocabulary.

The interaction with MARS is done via a **MARS request** using this language. Fields are described (named) by MARS requests. Fields having similar names can be described using a single MARS request.

3.1 Request format

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 format:

```
verb,  
    parameter 1 = value 1,  
    ...         = ...,  
    parameter N = value N
```

Where:

- **verb**, identifies the **action** to be taken, e.g. `retrieve` or `list`
- **parameter**, a **predefined keyword**, e.g. `type` or `date`
- **value**, the **value assigned to the parameter**, e.g. `Analysis` or `temperature`

Notes on MARS request syntax:

- Leading spaces and tab characters are ignored.
- **verb** and pairs `parameter=value` are separated by commas. Note the last pair is not terminated with a comma. MARS considers that as the end of a request.
- Directives may be in upper or lower case or a mixture of both; MARS converts everything except filenames 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 filenames.
- Abbreviations may be used for verbs, parameters and values, providing 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.

3.2 Retrieve request

The verb used is `retrieve`, and it is meant to retrieve data from MARS. This is an example:

```
retrieve,  
    class      = od,  
    stream     = oper,  
    expver    = 1,  
    date      = -1,  
    time      = 00,
```

```

type           = analysis,
levtype       = model levels,
levelist      = 1/to/31,
param        = temperature,
grid         = 2.5/2.5,
area         = europe,
use          = infrequent,
target       = analysis.12

```

The list of pairs parameter=value can be grouped as follows:

Identification directives

- Directives to identify the archive
- Directives related to date and time
- Directives for meteorological fields
- Directives for observations
- Directives for satellite images

Data storage directives

Post-processing directives

Execution control directives

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 [on-line Archive catalogue](#). Data manipulation (post-processing) directives and execution control directives are optional, depending on user's needs.

3.3 Identify the Archive

These parameters are a **ECMWF classification**, needed to distinguish among different characteristics of the same meteorological data, like different origins, different models which produced the data, different runs of the same model or different representations of the same data.

class

Specifies the ECMWF classification given to the data.

Archived data may arise from routine Operations ([od](#)), from Research experiments ([rd](#)), Re-Analysis data sets ([er](#) for ERA 15, [e4](#) for ERA 40), European Climate Network Support ([cs](#)), DEMETER project ([dm](#)) or experiments created by Member State's users (internet country code, e.g. [uk](#), [es](#),...). Operations and Re-Analysis are the most frequently used data classes

stream

Identifies the origin of the data or the Project that generated the data when the same Meteorological types are archived.

For example, [oper](#) is used for the Operational Atmospheric model and [wave](#) for the Wave model. Both models produce Analysis and Forecast data types.

expver

It is the version of the data.

Each experiment is assigned a unique code (version). Production data is assigned [1](#) or [2](#), and experimental data in Operations [11](#), [12](#),... Research or Member State's experiments have a four letter experiment identifier.

repres

Selects the representation of the archived data ([sh](#) for spherical harmonics, [gg](#) for gaussian grid and [ll](#) for latitude/longitude). Users must be aware `repres` is not a parameter for data manipulation but it identifies the representation of the archived data: some datasets contain the same data archived in 2 different representations, although it is not common practise.

In general, users can ignore this directive.

domain

In most of the cases, it refers to the Globe ([G](#)), i.e., the whole Earth. Domain is not a manipulation parameter.

Depending on the desired product, domain has different meanings:

- For Cluster and Tube products, it specifies the area over which the cluster computation is carried out. Working options for this parameter are General European Area ([G](#)), North West Europe ([A](#)), North East Europe ([B](#)), South West Europe ([C](#)) and South East Europe ([D](#)).
- For Wave data, it represents the geographical domain for which the data was produced. Working options for this parameter are Globe ([G](#)) and Mediterranean ([M](#)) for the Limited Area Wave Model. Northern Hemisphere ([N](#)) and Southern Hemisphere ([S](#)) can be used for old 2D-wave spectra products, parameter code 250.
- Fastex ([F](#)) - domain of the FASTEX experiment.

origin

Specifies the origin of the data, usually as the **CCCC** WMO centre identifier. It is used for datasets classified as [supplementary data](#) or data used or produced in the [Multi-Analysis Ensemble](#) project, or output of the [multi-model seasonal forecast](#) system or the [monthly forecast](#) system.

system

Labels the version of the operational forecast system for seasonal forecast related products. For example, the ECMWF seasonal forecasting system uses `system=1` for cycle 15r8 used between 1997 and 2002, and `system=2` for the cycle 23r4 based system introduced in 2002. For research seasonal forecast experiments, system is set to 0 (not used).

method

Used for seasonal forecast related products. For a given system or experiment, specifies how the analyses and forecasts were produced: `method=0` denotes the case when no ocean data assimilation was used to prepare ocean initial conditions; `method=1` is the standard case of data assimilation. Other methods may use [2](#),[3](#)...

3.4 Date and time directives

These parameters specify the base date of the model run for fields or the date of the observations. For a given date, they specify:

- a point in time for fields (`time` or `time + step`)
- a point in time for observations (`time`)
- observations for a period of time (starting from `time` extending `range`)

date

Specifies the Analysis date, the Forecast base date or Observations date.

Allowed formats are:

- **Absolute** as `YYYY-MM-DD`, `YYYYMMDD`. The day of the year can also be used: `YYYY-DDD`
- **Relative** as `-n`; `n` is the number of days before today (i.e., `-1 = yesterday`)
- Name of month (e.g. `january` for Climatology data)
- Operational monthly means are retrieved by setting date to `00`.

time

Specifies the time of the data in hours and minutes. Valid values depend on the type of data:

- Analysis time, Forecast base time or First guess verification time (all usually at synoptic hours: `00`, `06`, `12` and `18`).
- Observation time (any combination in hours and minutes is valid, subject to data availability in the archive).

The syntax is `HHMM` or `HH:MM`. If `MM` is omitted it defaults to `00`.

step

Specifies the forecast time step from forecast base time.

Valid values are hours (`HH`) from forecast base time. It also specifies the length of the forecast which verifies at First Guess time.

refdate

Specifies the date of the real-time forecast associated to the hindcast.

In the Monthly Forecasting system, hindcasts are run on the same day, same month for the last n years, and are associated to the real-time forecast by storing them under the same reference date.

reference

Specifies the reference forecast time step (in hours) at which the assignment of Ensemble forecast members to Tube products is carried out.

The tubing categorisation made at the reference Forecast time-steps is applied to a number of preceding forecast time steps. These differ according to the reference forecast time step. The

table below shows the values you can specify in step for each of the allowed values of reference, or conversely, the values you can specify in reference for a given value of step:

Reference	Step
96	48/72/96
144	96/120/144
168	72/96/120/144/168
192	144/168/192
240	192/216/240

range

For observations, it denotes the period, in minutes, starting from `time`. It can be used to specify periods of time over different dates. (i.e., `time=2100`, `range=360`, denotes observations from `21:00` to `03:00` next day).

For fields, it is mostly used in ocean data. For a time series product, range defines the values in time over which the time series extends. For a time averaged product, range is the length of the period over which the averaging has been made. Range is generally used with step, which defines the end point of the relevant time interval. For example, for type forecast, step gives the number of hours at the end of the time series, and range gives the number of hours between the beginning and the end of the time-series.

fcmonth

For certain datasets like the seasonal forecast monthly means, it specifies the complete calendar month which follows the forecast base date. For example, `fcmonth=2` is the second whole calendar month. For a forecast starting 1st January this would be February. For a forecast starting 2nd January this would be March.

fcperiod

For certain datasets like the monthly forecast monthly means, it defines the period over which the field has been averaged. Its units are dates. For example, `fcperiod=04-10` represents the average of days 4 to 10 of the forecast.

3.5 Directives for fields

type

It determines the type of fields to be retrieved. This parameter makes the selection between observations, images or fields, and it determines some of the valid remaining parameters of the request. Examples of fields are: Analysis (`an`), Forecast (`fc`), Perturbed Forecast (`pf`), Control Forecast (`cf`). Many other types will retrieve fields, and will need parameters related to fields. For example, note that Forecast will need directive `step`, Perturbed Forecast will need directive `number`, and so on.

levtype

Denotes type of levels. Its value has a direct implication on valid [levelist](#) values. Common values are: model level ([ml](#)), pressure level ([pl](#)), surface ([sfc](#)), potential vorticity ([pv](#)), potential temperature ([pt](#)) and depth ([dp](#)). Note ocean data is archived with [levtype=dp](#), and wave data is archived with [levtype=sfc](#).

levelist

Specifies the required levels. Valid values have to correspond to the selected levtype. For example, model levels can range [1](#) to [60](#). Pressure levels are specified in hPa, e.g. [1000](#) or [500](#). Potential vorticity levels are specified in units of $10^{-9} \text{ m}^2 \text{ s}^{-1} \text{ K kg}^{-1}$. For ocean fields, values are specified in metres below sea level, e.g, [5](#).

number

Selects the member in ensemble forecast run. It has different meaning depending on the type of data:

- for Perturbed Forecasts, specifies the Ensemble forecast member.
- for Cluster products, specifies the number of the cluster(s) of Ensemble Forecasts for which a cluster mean or a cluster standard deviation can be obtained.
- for Ensemble Forecast Probabilities, specifies the number of the Ensemble Forecast Probability.
- for Tube products, specifies the number of the tube.

Number is also used in those products that have an ensemble nature, like the seasonal forecast or monthly forecast datasets.

param

Specifies the meteorological parameter. The list of MARS parameters is extensive. Their availability is directly related to its meteorological meaning and, therefore, the rest of directives specified in the MARS request. In the search of parameter numbers, mnemonics or units, the [GRIB documentation](#) is a valuable source of information, as it describes ECMWF's local table 2 versions. For a given parameter, the [Data Finder](#) application will help in discovering from which data set and when they are available.

Meteorological parameters can be specified by their GRIB code ([param=130](#)), their mnemonic ([param=t](#)) or full name ([param=temperature](#)).

channel

It represents the frequency band of the data. This attribute is used for very specific data: brightness temperatures ([param=194](#)) from Errors in First Guess data type ([type=ef](#)).

diagnostic

Specifies the diagnostic function number code in sensitivity forecast products:

- [0](#), is the initial analysis or the control forecast.
- [1](#), is the 2 day forecast error (2 day fc minus verifying analysis)
- [2](#), is the difference between the 2 day forecast from day d, and the 1 day forecast from d+1.

- `5`, is the difference between the 2 day forecast from day `d` at 12z, and the 36h forecast from `d+1` at 00z

iteration

Specifies the step number in the algorithm for the minimisation of the `diagnostic` function for sensitivity forecast products. Usual iteration values range from `0` to `3`.

frequency

Specifies the required frequency components of **wave model spectral fields**. These wave model fields are discretised in a number `NFRE` of frequencies and a number `NANG` of directions at each grid point.

direction

Specifies the required direction components of **wave model spectral fields**. These wave model fields are discretised in a number `NFRE` of frequencies and a number `NANG` of directions at each grid point.

product

For ocean fields, it denotes how the archived field is defined in time. The product may be an instantaneous field (`inst`), a field accumulated in time from the start of the integration (`tacc`), a field averaged in time over a given range (`tavg`) or a time series (`tims`).

section

For ocean fields, denotes the spatial orientation of the archived field. The field might be horizontal (`h`), if vertically oriented it might be meridional (`m`) or zonal (`z`) or time series, or it might be a vertically oriented time series (`v`).

latitude

For ocean fields, on a zonally oriented or time series product, denotes the location in latitude. Value is in degrees, positive values for north.

longitude

For ocean fields, on a meridionally oriented or time-series product, denotes the location in longitude. Value is in degrees east, in the range 0-360.

3.6 Directives for observations and satellite images

type

It determines the type of observations to be retrieved. For example Observations (`ob`), Feedback (`fb`), Analysis Input (`ai`) or Analysis Feedback (`af`) will retrieve BUFR data, and the request will need parameters related to observations. Satellite images are coded in GRIB, and they are retrieved using `type=im`.

obstype

Specifies the observation subtype (BUFR code) or the satellite channel for images.

This data is not produced at ECMWF but received from all sorts of sources around the world. Availability is directly related to its reception from those sources.

ident

Specifies the **satellite identification number** for Satellite data or the **WMO Station number** when retrieving observations. Examples of Satellite identification number are [52](#) for METEOSAT 5 or [53](#) for METEOSAT 6. WMO Station number can be specified, e.g. [03953](#).

duplicates

Specifies if duplicated observations are to be kept or removed. The archive may contain duplicated observations, which are removed ([duplicates = remove](#)) or kept ([duplicates = keep](#)) in the target file according to the value of this parameter.

block

Specifies WMO Block number(s) for observations, e.g. [02](#).

instrument

For satellite data, it specifies a filter upon instrument. Instruments can be specified by name, e.g. [hirs2](#), [ssu](#), [msu](#), [amsu-a](#), [amsu-b](#). The BUFR instrument code is also valid, e.g. [605](#), [627](#), [623](#), [570](#), [574](#).

3.7 Data Storage directives

target

Specifies a Unix file into which retrieved data is to be written after manipulation.

source

Specifies a Unix file from which data is to be read.

fieldset

Denotes a MARS variable into which retrieved data is to be written after manipulation. It is a name used to represent a temporary collection of fields which can be manipulated with further MARS requests.

MARS uses temporary storage to hold fieldsets. Once MARS finishes its execution, this storage is removed, thus the data is lost if it has not been saved with a [write](#) request.

3.8 Post-processing directives

grid

Specifies the grid mesh for Latitude/Longitude output or the number of latitude lines between the Pole and the Equator for Gaussian output.

For Latitude/Longitude output, the format is in degrees, i.e. [grid = 2.5/2.5](#)., where the first number denotes the east-west resolution (longitude) and the second denotes the north-south resolution (latitude).

For Gaussian output the format is an integer representing the number of lines from Pole to Equator, i.e. [grid = 80](#).

One cannot just use any number for the lat/lon grid, but an integer fraction of 90 degrees, latitude range from the Equator to the Pole, e.g. [2](#), [2.25](#), [2.5](#), [2.8125](#), [3](#), [3.5](#), [3.75](#), [4.5](#), [5](#), etc.,.

gaussian

Specifies the desired type of Gaussian grid for the output. Valid Gaussian grids are quasi-regular ([reduced](#)) or [regular](#). Directive gaussian can only be specified **together with directive grid**. Gaussian without grid has no effect.

When [gaussian = reduced](#) is selected, only certain values are valid for parameter grid, according to the definitions for [quasi-regular Gaussian](#) grids used at ECMWF.

area

Specifies the desired sub-area of data to be extracted.

Some predefined areas exist (Europe, and four quadrants for each hemisphere). Alternatively, you may specify an area of your own, by inputting the area limits as [North/West/South/East](#); if South > North, the values are swapped and a warning issued; southern latitudes and western longitudes must be given as negative numbers. For example, Europe may be defined by [75/-20/10/60](#).

Areas can be defined to wrap around the globe, e.g., in order to retrieve all longitudes but Europe one would specify [area=75/60/10/-20](#). Sub-area extraction works on regular latitude/longitude fields, regular gaussian fields and wave fields. For observations, it is used as a filter.

There are some [area definition examples](#) on the Data Services documentation.

frame

Specifies the number of points to be selected from a sub-area inwards. It works together with area parameter. For example, [area=europe](#), [grid=2/2](#), [frame=2](#) will select a frame with outer limits [74/60/10/-20](#) and inner limits [72/58/8/-18](#).

resol

Specifies the desired **triangular truncation** of retrieved data, before carrying out any other selected post-processing. The default is automatic truncation ([auto](#)), by which the lowest resolution compatible with the value specified in [grid](#) is automatically selected for the retrieval.

Users wanting to perform post-processing from full spectral resolution should specify Archived Value ([av](#)). The following are examples of existing resolutions found in the archive: [63](#), [106](#), [159](#), [213](#), [255](#), [319](#), [399](#), [511](#) or [799](#).

This parameter has no meaning/effect if the archived data is not in spherical harmonics form.

rotation

Specifies a rotation for the output fields. The values for this parameter consist of the latitude of the southern pole and the longitude of the southern pole in degrees, e.g. [rotation = -30.0/10.0](#).

accuracy

Specifies the **number of bits per value** to be used in the generated GRIB coded fields. A positive integer may be given to specify the preferred number of bits per packed value. This must not be greater than the number of bits normally used for a Fortran integer on the processor handling the request (typically 32 or 64).

packing

Specifies the packing method of the output fields. Valid values for latitude/longitude output are simple (**si**) or second order (**so**). Valid values for spectral fields are complex packing (**co**) or simple packing (**si**). In order to preserve the packing of the archived data one would specify archived value (**packing = av**).

specification

Specification of reduced gaussian grid to use. Can only be used with N80 or N160 **reduced gaussian grids** to force an old grid definition to be used (Research Department designated _12) which has 12 points in the northernmost and southernmost latitudes, e.g. **specification = 12**.

style

Specifies the style of post-processing. A valid value is **style = dissemination**, in order to instruct MARS to perform postprocessing in a way as close to the dissemination system as possible. For example:

- sub-area extraction will not be expanded but contracted when **area** and **grid** don't fit
- interpolation is performed to the finest grid-point allowed in dissemination according to the operational setup (let's say it could be 0.5 by 0.5) and then by selecting alternate points to reach the desired resolution (for a 2 x 2 degree grid, it will select every fourth point).
- rotation of vector components (winds) is performed using hirlam routines.

bitmap

Specifies a UNIX filename containing a **bitmap definition**, in the format defined in appendix 3 of ECMWF Meteorological Bulletin M3.1: The dissemination of ECMWF products to Member States, D.Jokic.

3.9 Execution control directives

expect

Specifies the number of fields the request is expecting to retrieve. Please, note that some datasets do not have all parameters at all levels. Use this directive in case a data set includes data gaps, e.g if a data set only had parameter geopotential at 1000/500 and temperature at 850, you would specify:

```
param = 129/130, levelist = 1000/850/500, expect = 3
```

If **expect = any** is provided, MARS will retrieve any fields or observations for the given request, but it will not fail in case it does not find any data (in this case, an empty target file is produced).

padding

For historical reasons, each GRIB message retrieved is padded to a multiple of 120 bytes, and each BUFR message to a multiple of 8 bytes. This directive can specify a number of bytes to use for padding, e.g., `padding=0` for no padding.

database

Specifies in which of the available databases the search for the required data should be carried out. This feature is meant to be used by analysts or Operational suites, and its use is best avoided by the general user.

use

Provides MARS with a hint about the use of the retrieved data. If `use = infrequent`, the MARS system will not keep an internal cache copy of the data once it has satisfied your request. This is particularly recommended when retrieving large datasets.

3.10 Considerations on Parameter values

Parameters may consist of 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 keyword `to` as well as `/` and `by`.

Examples of different formats for values are:

format	directive	example
single value	<code>param</code>	<code>temperature</code>
list of values	<code>step</code>	<code>12/24/48</code>
range of values	<code>date</code>	<code>19990104/to/19990110/by/2</code>

The number of fields described in a MARS request is computed by multiplying the number of values of its identification parameters 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 parameters accept `all` as a valid value. In this case, MARS will deliver data available matching the rest of the request, e.g, `param = all`.

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

Note that not all possible combinations `parameter = value` will name a field in the archive. Therefore, some syntactically correct requests will not retrieve anything if the data they represent has not been produced/archived. Furthermore, certain directives have only a meaning when retrieving specific datasets and should be ignored or set to `off` for other datasets.

3.11 Language hints for ECMWF fields

This section presents some hints in MARS terms on how to name the different products. This is meant as a guideline, and shows the parameters that select a specific data set, but not all parameters necessary to retrieve the data. An [on-line archive overview](#) is available in order to help users navigate through the MARS catalogue. Once in the catalogue, you can view the batch request.

Usually, operational archive data can be identified with the following MARS directives:

```
class          = od
domain         = g
expver         = 1
```

Atmospheric deterministic forecast

```
stream        = oper
```

Analysis

```
type          = an
• Surface
  levtype     = sfc
• Model levels
  levtype     = ml
  levelist    = 1/to/60
• Pressure levels
  levtype     = pl
  levelist    = 1000/850/500...
• Isentropic levels
  levtype     = pv/pt
```

Forecast

```
type          = fc
step          = 12/to/240/by/12
```

First Guess

```
type          = fg
step          = 6
```

Four-dimensional variational Analysis

```
type          = 4v
```

Errors in Analysis

```
type          = ea
```

Errors in First Guess

```
type          = ef
```

Wave deterministic models

stream = wave

Ensemble Forecast System

stream = enfo/waef

Test system for severe weather prediction

time = 00

expver = 12

Control Forecast

type = cf

step = 12/to/504/by/12

Perturbed Forecast

type = pf

number = 1/to/50

step = 12/to/240/by/12

Forecast Probabilities

type = fp

step = 24/to/240/by/24

Forecast mean probabilities

type = fp

step = 144-168/144-240/192-240

Ensemble means

type = em/es

Clusters

type = cm/es

Tubes

type = tu

step = 96

reference = 48/72/96

Boundary conditions

stream = scda

Multi-Analyses Ensemble

stream = maed/amap

Seasonal Forecast

system = 2

method	= 0/1
<i>atmospheric</i>	
stream	= seas
levtype	= ml/pl/sfc
<i>wave</i>	
stream	= wasf
<i>ocean</i>	
stream	= seas
levtype	= dp
product	= inst/tims/tacc/tavg
section	= h/m/z/v
Monthly Forecast	
stream	= mofc/wamf
Monthly datasets	
<i>deterministic</i>	
stream	= mnth/wamo
<i>seasonal forecast</i>	
stream	= sfmm/swmm
<i>monthly forecast</i>	
stream	= mfmm/wmfm
<i>Climatology</i>	
type	= cl
Special projects	
<i>Re-Analysis 15 years</i>	
class	= er
<i>Re-Analysis 40 years</i>	
class	= e4
<i>Demeter</i>	
class	= dm
<i>Provost</i>	
class	= pv

ECSN-Hyretics

class = cs

Research or Member State's experiments

class = rd

class = at/be/hr/dk/fi/fr/de/gr/hu/is/ie/it/nl/
no/pt/si/es/se/ch/tr/uk

expver = aaaa

Other Centres data

stream = ammc/cwao/edzw/egrr/lfpw/rjtd/kwbc

3.12 Language hints for ECMWF observations

Observations

type = ob

obstype = synop land

time = 09:30

range = 180

Feedback

type = fb

obstype = synop land

time = 11:45

range = 360

Analysis input

type = ai

obstype = bufr/tovh/atov/grad/wavescat/scar/ralt/
ssbt/reo3/storm/qscat

time = 00/06/12/18

Analysis Feedback

type = af

obstype = fbconv/fbtovs/fbrad1c/fbgeos/fbscat/
fbssmi/fbreo3

time = 00/06/12/18

Obstypes available

MARS abbreviation	Observation type
LSD	Land surface data

SSD	Sea surface data
VSNS	Vertical soundings (not satellite)
VSS	Vertical soundings (satellite)
SLNS	Single level upper-air (not satellite)
SLS	Single level upper-air (satellite)
OD	Oceanographic data.
DD	Derived data
SDS	Surface Data (Satellite)

Subtype number (obstype)	MARS abbreviation	Observation type
--------------------------	-------------------	------------------

	LSD	Land surface data
1	S	Synop land
2	S2	Synop record 2 land
3	SA	Synop auto land
4	SA2	Synop auto record 2 land
140	METAR	Metar

	SSD	Sea surface data
9	SAB	Synop ship abbreviated
11	SS	Synop ship
12	SS2	Synop record 2 ship
13	SAS	Synop auto ship
14	SAS2	Synop auto record 2 ship
19	SH	Shred
20	RCD	Reduced COADS data
21	DSU	Dribu surface
22	BSU	Bathy surface
23	TSU	Tesac surface

	VSNS	Vertical soundings (not satellite)
91	P	Pilot land
92	PS	Pilot ship
95	WP	Wind profiler
96	EWP	European wind profile
97	EWTP	European wind & temp profile
101	T	Temp land
102	TS	Temp ship
103	TD	Temp drop
104	R	Rocob land
105	RS	Rocob ship
106	TM	Temp mobile
107	CT	Climate temp

	VSS	Vertical soundings (satellite)
51	HR	High resolution
53	RTOV	RTOVS
54	TOVB	TOVS1b
55	ATOV	ATOVS
56	ATOP	ATOVS products
61	SMT	Satem mean temp.
62	SW	Satem water
63	SHL	Satem higher level mean
65	SLM	Satem levels merged
71	MT	Mean temp.
72	W	Water
73	HL	Higher level
75	TLM	TOVS levels merged

	SDS	Surface Data (Satellite)
121	WAS	Wave Scatterometer

122	WIS	Wind Scatterometer
123	RAL	Radar Altimeter
124	SST	Sea Surface Temperature
125	RW	Reprocessed Wind

	SLNS	Single level upper-air (not satellite)
141	COD	Codar
142	AIR	Airep
143	COL	Colba
144	AM	AMDAR
145	AC	ACARS

	SLS	Single level upper-air (satellite)
82	SAT2	Satob section 2
83	SAT3	Satob section 3
84	SAT4	Satob section 4
85	SAT5	Satob section 5
86	SAT6	High resolution winds
87	SAT7	High density unified format satellite wind
88	GEOR	Geostationary radiances
89	GCSR	Geostationary clear sky radiances

	OD	Oceanographic data
131	DO	Dribu oceanographic
132	BO	Bathy
133	TO	Tesac

	DD	Derived data
164	PA	PAOBS

	TRMM	TRMM
129	TBT	TMI brightness temperature
130	TI3R	TMI instantaneous 3D rainfall structure

		SSMI
126	SSMI	Microwave images
127	SSBT	Seas surface brightness temperature

	QSCAT	QuikScat
137	QSCA	Quikscat
138	QSSW	Quikscat sea winds
139	QS50	Quikscat sea winds 50 km.

	REO3	Ozone
206	REO3	Retrieved ozone

	CC	Chemical constituent
162	OVS	Ozone vertical sounding

	SF	Synoptic features
31	STOR	Storm data

4 Architecture

The MARS system is based on a **client/server architecture**, which makes it more flexible and adaptive to changes in a complex environment like ECMWF's.

The interaction between clients and servers is based on MARS requests. Clients send MARS requests to a server which looks for the data on-line or off-line, depending on the capabilities of the given server. If it fails to satisfy the request, the client will contact another server, according to a list of pre-configured servers.

Cache mechanisms existed in the past to move some activity away from the main archive. Since the implementation of the new Data Handling system in 1997, these caches have not been needed, but they could re-appear if necessary.

4.1 Clients

The MARS client is a program written in C language linked with the LIBEMOS library (which contains the GRIBEX routines and GRIB interpolation software).

The MARS source code is also embedded in other applications, such as Metview.

Access at ECMWF

MARS can be executed either in batch or interactive mode. Usually, clients issue requests from:

- Workstations and Servers, suitable for most retrievals
- Supercomputers, retrieving data to be used as input for models
- Metview, for interactive/batch visualization/plotting and manipulation

It is recommended to use workstations or workstation servers for data retrieval instead of the supercomputers. This avoids unnecessary idle time when data has to be read from tape.

Member States/Co-operating States access

Most of the access from Member State users comes via the ecgate system, by logging in or submitting batch jobs to ECMWF's computers.

Moreover, the MARS client software is also available to access ECMWF's MARS archive system remotely by installing it at Member State sites. It removes user's need to login. Access and transfer of the data is transparent to the user, although data transfer may be slow. In addition, it is possible to create secondary cache systems at Member State sites in order to minimise remote access and transfer for popular data.

4.2 Servers

Reports Data Base

Contains on-line observations received via ECMWF's acquisition system. This system has been interfaced with MARS to allow real-time observation access. Access to this server is meant for monitoring and operational archive purpose only.

Fields Data Base

This is where models running at ECMWF write their outputs. It contains data produced by the most recent cycles.

Depending on the configuration and disk resources, it can contain up to **several days of operational data** and more recent research experiments. Also, there are many Fields Databases, several per supercomputer or workstation server able to run ECMWF's models.

It is meant to provide very fast access as all the data resides on-line. This makes it very suitable for model input data retrieval or last cycles data access.

Main Archive

This is the core of the MARS System and consists of the following hardware and software:

Hardware

- Multiprocessor dedicated servers.
- Several Terabytes of disk space, used for temporary storage before data is written to tape and for caching purposes while data is read from tape.
- Tape silos

Software

- A set of applications written in C++ linked with the LIBEMOS library.
- Tivoli Storage Management software (TSM, formerly known as ADSM) which manages the tape robots. In 2002, a migration process to the High Performance Storage System (HPSS) will be carried out, expected to last until end 2004.

Some characteristics of the Main Archive system such as request scheduling and data collocation are very important for users in order to optimise data retrieval.

4.3 Interaction client/server: request execution

The MARS client has a configuration of servers to access when looking for data. The following is a schematic view of actions the MARS client performs per request:

- Check the **request syntax** with the help of the MARS language.
- **Print the request** that it is going to process and the number of fields expected
- Query the **cache system** (if configured)
- Query the supercomputers' **Fields DataBases** (if the data are not cached)
- Query the **main archives** (if data are not in the FDB)
- **Transfer** the data to the client and post-process (if needed)
- Caches the data (if the cache is present)
- Reports on the result

Note that post-processing is done while the data is being transferred and before writing to disk.

5 Retrieving data

5.1 Request scheduling

The main archive implements a **queueing system** necessary to satisfy all type of requests: operational tasks carrying out time-critical work, interactive users retrieving a few fields for visualization, batch users retrieving Gigabytes of data, etc....

Resources available on the main archive impose some limitations on the number of requests processed simultaneously: these resources are basically the number of Unix processes MARS can start and the number of drives available to read data from tapes. The **cost of a request** is evaluated on arrival in terms of number of fields requested and their location, either on disk or on tape. The higher the number of tapes a request has to access, the more likely that it will be queued. It is possible that requests will be queued if the MARS server reaches certain limit of concurrent requests.

The queueing system handles **priorities**: the highest priority will be chosen next for execution. This priority is computed from the age of a request since it entered the queue. An artificial mechanism of ageing requests allows to prioritise operational work or interactive users.

[View the MARS queue.](#)

Data collocation

A request is scheduled more efficiently if it minimises the number of tapes it has to access in order to be satisfied. In order to do so, users must know how the data is organised.

The MARS system is organised in a tree fashion based on MARS parameters. At the bottom of the tree there are what we called the **hypercubes** or **archive objects**, that are aimed to be in a single file on tape. There is a compromise between the amount of data these cubes hold and the relationship between the data they contain. By studying data access patterns, we have come up with the following rules:

- **1 file per month of Analysis** (1 type of level, all times, levels, and parameters), for example, have a look at [January 2003 model level analysis](#) in the catalogue.
- **1 file per Forecast** (1 type of level, all steps, levels and parameters), for example, have a look at [1st of January 2003 model level forecast at 12 UTC](#) in the catalogue.
- **1 file per Ensemble Forecast** (1 type of level, all steps, members, levels and parameters), for example, have a look at [1st of January 2003 pressure level 50 member ensemble at 12 UTC](#) in the catalogue.

Different projects have different needs, and this rule may vary. Users are encouraged to visit the [MARS archive catalogue](#) in order to inspect how much related data an archive object contains. The description above is the rule, but resources available at certain times might cause to break it, e.g., 1 month of Analysis may be in 2 files because at that particular month the MARS system was short of disk space and data had to be written to tape earlier than desired.

5.2 Post-processing

Any requested data manipulation or post-processing is carried out by the MARS client, except in the case of a local Member State's client where data is first processed at ECMWF prior to its transmission over the network. The post-processing is carried out by a set of routines present in the libemos library. Please, refer to the [field interpolation software](#) routines for in-depth details about such post-processing.

Sub-area extraction

Most of the data at ECMWF is global. A subarea can be created using the `area` directive by defining its latitude and longitude boundaries: `North/West/South/East`.

Global grids have a grid mesh implicitly based on (0 West, 0 North); the grids do not wrap around at (0 E, 360 W). For example, a 3x3 degree grid has latitudes at 90N, 87N,..., 87S, 90S and longitudes at 0E, 3E,..., 354W, 357W. Subareas are created from a global grid using the same implicit origin and spacing; if necessary, the subarea boundaries are adjusted to fit on the grid mesh by enlarging it.

Polar latitudes are made up of repeated grid-point values; a wind V-component has an adjustment for longitude. Gaussian grids do not have a latitude at either Pole or at the Equator.

Sub-area extraction is possible for regular gaussian and latitude longitude fields (including wave). They cannot be applied if the resulting field is in spherical harmonics or reduced (quasi-regular) representation.

Conversions

The following facilities are available:

- reduce the triangular truncation of spherical harmonic fields, e.g. `resol=319`
- change the resolution of Gaussian grids, e.g. `grid=42`
- change the resolution of latitude/longitude grids, e.g. `grid=5/5`
- convert spherical harmonic to Gaussian grids, e.g., `grid=160, gaussian=reduced`
- convert spherical harmonics to latitude/longitude grids, e.g. `grid=2.5/2.5`
- convert quasi-regular Gaussian grids to regular Gaussian grids, same or different resolution, e.g., `grid=160, gaussian=regular`
- convert Gaussian to latitude/longitude grids, e.g., `grid=2.5/2.5`

Derived fields

Some fields are not archived but are derived from others. This is the case for wind components (U and V), which are derived from vorticity and divergence.

Accuracy

GRIB coded fields have a specified number of bits per packed value. This value can be varied using the accuracy directive; there is no gain in precision by using a value higher than the number of bits originally used for archiving.

Truncation before interpolation

Spectral fields are automatically truncated before interpolation to grid fields to reduce data volumes and spurious aliased values. The truncation can be controlled using the `resol` directive. Default truncations are detailed below:

Grid increment	Truncation
2.5 and greater	T63
1.5 to 2.5	T106
0.6 to 1.5	T213
0.4 to 0.6	T319
0.3 to 0.4	T511
0.15 to 0.3	T799
0 to 0.15	T1023

Users wanting to post-process at the full archived resolution can specify `resol = av` in the request. Note that high resolutions might need more resources to carry out post-processing.

Rotation

Fields on spherical harmonics or gaussian grid can be rotated with the directive `rotation`. The rotation is performed prior to any other conversion. Therefore:

- Spherical harmonics fields can be retrieved as rotated spherical harmonics or converted to rotated latitude/longitude or gaussian grid fields.
- Gaussian grid fields can be retrieved as rotated latitude/longitude or rotated Gaussian grid fields.

If the input is a spectral field and the output is a rotated spectral field, a file of rotation coefficients is generated and used in the processing. The convergence of the algorithm for rotating the spectral fields is sensitive to the spectral truncation. For some high resolutions and some large angles of rotation, it is necessary to split the rotation into three steps (two forwards and one backwards!). The size of the rotation coefficient files can be very large.

If the input is a spectral field and the output is a rotated grid point field, the spectral field is interpolated to a non-rotated gaussian grid which is then transformed to the required rotated grid point field using a 12-point interpolation based on the FULL-POS scheme used in the ECMWF forecast model.

A rotated grid-point field is created from an input grid-point field by finding for each rotated grid-point its nearest four neighbours in the unrotated field and carrying out a bilinear interpolation.

5.3 Retrieval efficiency

This section presents some hints users should follow when retrieving data.

Local Disk

Whenever possible, use a local filesystem to store the target file. NFS has proved to be very slow and its use may lead to an unnecessary increase in network traffic. MARS will issue a warning!

Amount of data

Make an estimate of the data to be retrieved before issuing a request. It is easy to retrieve Gigabytes of data. Check that computer resources and limits are adequate for the amount of data to retrieve/interpolate in order to avoid unnecessary processing (MARS will fail if a quota is exceeded or in the case of any Unix problem regarding resources such as memory or CPU time).

Number of fields

Make an estimate of the number of fields to be retrieved before issuing a request. Try to retrieve a sensible number, up to tens of thousands of fields.

Reduce the number of tapes involved

The number of tapes a MARS request is going to access will have an impact on its scheduling on the server. As a rule of thumb, two or more separate requests accessing files on different tapes are scheduled more efficiently than a single request accessing two or more tapes. A large number of tapes implies more waiting time.

Large datasets

When retrieving large datasets (e.g. Re-Analysis), try to retrieve as many data from the same tape file as possible. Then, avoid caching the data on the server by specifying: `use = infrequent`

5.4 Trouble-shooting

Syntactic errors

If a comma is missing or an unknown MARS keyword is specified, MARS will stop processing and report a syntax error.

Semantic errors

Data not found

Usually means the MARS directives do not specify archived data.

Expected xx, got yy

The server transferred some data and the client failed; usually the client expected more fields than were sent by the server. Either some data do not exist, are missing on the server or the request asked for a syntactically correct parameter which is not in the archive.

Inconsistency in field ordering

This error occurs when a server sends a field which does not correspond to the MARS request. Operational servers are not likely to deliver inconsistent data, but it may well happen in test environments.

System limits

Some system resource limits can be controlled by the user (consult the man pages of your shell if running in interactive mode or documentation about the software used for batch mode).

Error writing to file

It usually indicates the user has exceeded his/her quota on the filesystem holding the target file or that filesystem became full.

Memory allocation failed

It usually indicates MARS needs more memory than the available for user processes in order to execute a request.

CPU time limit

It usually indicates the MARS process has exceeded the CPU time limit. The kind of post-processing and the number of fields retrieved have a direct implication of the CPU time needed by MARS to satisfy a request.

MARS parameter = all

By using the value all on certain parameters, MARS is asked to retrieve ALL data available which matches the rest of the request.

In some cases, all data available is not all the data that the user expects (e.g. if an operational archive has not yet been completed, MARS will deliver only what it currently holds). The use of [all](#) is best avoided.

It should be noted that retrievals from the Fields DataBase do not accept [all](#) as a valid value.

ANS4301E file data unavailable on server

These are TSM errors, meaning that the data is unavailable from tapes for some time. These errors are passed to the client and make your request fail. As a rule, re-run your request before reporting the error. If the failure is consistent, please inform ECMWF's Call Desk.

Debugging

Some failures are not evident to explain from the MARS report. In such cases, the user can turn on debug messages by setting environment variable **MARS_DEBUG** to any value different from 0.

Please, note that running in debug mode can generate large amount of output. Users are advised to re-direct MARS output to a file.

Default values

MARS will guess unspecified parameters in a request and will assign default values to them. Some default values are not valid in the context of all possible retrieve requests. Minimise their use.

No semantic check

MARS does not perform any semantic check on the request. A MARS request can be syntactically correct but may not describe any archived data.

Set of working requests

It is advisable to have a catalogue of working requests and re-use or modify them as needed.

MARS messages

MARS has the following levels for messages it prints on execution:

- **INFO**, request being processed and a report on the execution at the end.
- **WARNING**, any unusual aspect of the execution.
- **ERROR**, system or data errors which do not stop MARS execution.
- **FATAL**, terminates the execution of MARS.

Report

At the end of each request, MARS will print a report on all the aspects of its execution:

- which server delivered the data
- number of fields/reports which have been retrieved/interpolated
- time spent in retrieving/interpolating
- network transfers
- memory usage
- the most wanted message: No errors reported

This report gives users an idea about the resources needed for a request, and can be used for future reference when retrieving similar datasets.

Field order

Users should not expect retrieved fields to be in any specific sequence in the target file. Depending on the MARS configuration, fields can be retrieved differently. Therefore, user programs processing the target file must take this into account.

It is possible to obtain always the same sequence, i.e. the order of the parameters in the request, by using a fieldset.

Avoid sub-archives

Avoid creating sub-archives in file based storage systems, e.g. ECMWF's ECFS. This system does not provide data collocation, and a future access to the same data will be slower than retrieving it again from MARS.

Unless data has been post-processed, do not store Gigabytes of data in ECFS.

5.5 Data deficiencies in the archive

There are some known deficiencies in the archives (suspect or missing data). Those are listed in the [Data Services information pages](#) and in the [Changes in the archive](#) pages.

5.6 Frequently Asked Questions

I don't find operational First Guess anymore in the catalogue. Where is it?

With the introduction of 12 hour 4DVAR on 12th September 2000, the data type First Guess has not been produced any longer on the operational deterministic model. That data is available as type Forecast from previous cycles and time step validating at First Guess time.

The correspondance between First Guess and Forecast is as follows:

type = First Guess	type = Forecast
date=today,time=00,step=06	date=yesterday,time=12,step12
date=today,time=06,step=06	date=today,time=00,step=06
date=today,time=12,step=06	date=today,time=00,step=12
date=today,time=18,step=06	date=today,time=12,step=06

My retrieve request fails with “data not found”. What can I do?

It means that what MARS has understood of your request is not found in the archive. That could happen for several reasons:

- **wrong defaults:** you have not specified all necessary directives and MARS has filled some with default values, but they are wrong. For example, if you specify levtype=ml and no levelist, the default is a pressure level levelist. Users are advised not to rely on default values
- **data not yet produced:** a common mistake is to specify a range of dates that end in a future date, or to retrieve data for current date when the suite that produces the data has not run yet. Date should be the first directive to check when such error occurs.
- **data not archived:** the data doesn't identify any fields in the archive, or it was discontinued. For example, first guess data was discontinued on September 2000.

It is recommended to carry out the following actions:

- Check the request as it is **processed by MARS**. MARS will print out the request it is processing. It sometimes changes directives, or inherits directives from previous requests, etc., and the resulting request doesn't represent any field in the archive.
- Visit the on-line [archive catalogue](#), following the directives of the request printed by MARS. After some point in the navigation tree, you could find a dead end.

If after all you still don't know why your retrieve failed, please, seek advice from [User Support](#).

I get the error “unexpected charater in input”, but can't see anything wrong!

It is almost certain it is due to a missing comma in the line above. Remember that each directive block (`parameter=value`) must terminate with a comma, except the last line. Also check for misplaced = or /.

I cannot retrieve yesterday's observations. Why?

Observations are archived in MARS with 2 days delay. As a rule of thumb, you can always access `date = -3`, as the time at which observations archiving is carried out may vary from day to day depending on suite schedule.

I do not seem to have permission to access some data. Why and what can I do ?

Not all the data held in MARS are freely available to everyone. For example, the message:

```
restricted access to current forecast data
```

means you do not have permission to access the latest forecast, i.e. those which are still valid. External users may need a licence agreement from their national meteorological service before they can access such data. Contact your Member State Computing Representative in such cases.

I cannot retrieve recent Initialised Analyses. Why ?

Production of Initialised Analyses finished on 29/01/1996. After that date only Analysis is archived.

Parameter packing doesn't seem to have any effect if no other post-processing is carried out.

That's correct. At the moment you need to specify other post-processing for `packing` to work.

What is the most efficient way to loop over several requests ?

If you loop in a script, then do so in the following hierarchy:

```
date (outer loop)
  time
    step
      number (EPS only)
        level
          parameter (inner loop)
```

In any loop always put the mars command at the highest level possible, i.e., try to avoid putting the mars command in the innermost loop.

Thus, for example, if you wish to retrieve 3 months of analysis data (April 1999 to June 1999), then loop at the date level:

```
for month in 19990401/to/19990430 19990501/to/19990531
19990601/to/19990630
do
  mars <<EOF
    retrieve,
      ....
      type= an,
      ...
      date= $month,
      ...
```

EOF
done

How can I filter Analysis Input observations ?

Analysis Input/Feedback is archived and retrieved as used/produced by the Analysis, therefore, no post-processing is carried out on retrieval. However, users can use MARS to extract observations from already retrieved files.

This can be achieved by changing the type to observations and specifying a retrieval from source. For example, if we had retrieved all the conventional observations used in yesterday's 12 Z analysis, one could extract synop land data, 1 hour around 12 Z with the following request:

```
retrieve,  
    type           = ob,  
    obstype        = synop land,  
    time           = 1130,  
    range          = 60,  
    date           = -1,  
    source         = conventional.12z,  
    target         = synop.1hour
```

6 Advanced use

This section is intended to explain some features of MARS which might help to organise scripts or to manipulate already retrieved data. In the MARS language section, the verb `retrieve` was presented. MARS accepts other verbs: `read`, `write`, `list` or `compute`.

6.1 Listing archive contents

The verb `list` queries the MARS archive about the data availability for a given request. It is suitable for interactive or batch use to check availability of certain datasets and its collocation in the archive.

Note that default values are not supplied by MARS. However, some parameters (namely `class`, `expver`, `stream`, `type` and `date`) are necessary in order to avoid listing the whole archive (which may take a long time!).

Users can specify a `target` in order to keep the report in a Unix file. Otherwise, MARS will display it to the standard output.

Here is a sample request with the report output. For this particular data set, not all parameters are present at all levels. The report shows there are 6 fields available, they are in 1 file, and they represent a total of 86.6 Kilobytes.

```
list,  
  class      = od,  
  type       = an,  
  expver     = 1,  
  stream     = kwbc,  
  date       = 19990222,  
  time       = 0000/1200,  
  levtype    = pl,  
  levelist   = 1000/850/500,  
  param      = 129/130
```

Provides the following output:

```
class      = od  
cost       = 6 fields, 86.6484 Kbytes on 1 tape  
expver     = 1  
file[0]    = marsa:/marsodkwbc:/1/an/19990201/pl/62694:/  
20000331.022830  
file[1]    = -  
id         = 62694  
levtype    = pl  
stream     = kwbc  
type       = an  
date       file length levelist offset  param  time  
1999-02-22 0    16540  1000    1751614 129.128 00:00:00  
1999-02-22 0    11284   850     1768154 130.128 00:00:00  
1999-02-22 0    16540   500     1779438 129.128 00:00:00  
1999-02-22 0    16540  1000    1795978 129.128 12:00:00
```



```
1999-02-22 0      11284   850          1812518 130.128 12:00:00
1999-02-22 0      16540   500          1823802 129.128 12:00:00
```

```
Grand Total:
=====
```

```
Entries          : 6
Total            : 88,728 (86.6484 Kbytes)
```

6.2 Manipulate already retrieved data

`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 directive `source`. This data can either be stored in a `fieldset` for further manipulation or written to a Unix file specified with the parameter `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
```

Verb `retrieve` can also be used to extract data from a source Unix 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,
  class          = od,
  type           = an,
  date           = 20010101,
  time           = 12,
  param          = z/t,
  levtype        = pl,
  levelist       = 1000,
  grid           = 2.5/2.5,
  source         = myfile,
  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 it has to specify all the directives, and the user might not know what are the exact contents of the source file.

6.3 Retrieve data into multiple targets

MARS supports automatic generation of target files using MARS keywords enclosed in square brackets []. The target name has to be enclosed in double quotes.

It is a very useful way of organising target files depending on the fields to retrieve. Instead of making different calls to MARS that may access several times the same tape, one could retrieve as much data from the same tape as possible, and make use of this facility to have different target files.

The following example retrieves Pressure level Analysis for the four main synoptic hours, storing them in four different target files, one per synoptic time:

```
retrieve,  
    type           = an,  
    expver         = 1,  
    levtype        = pl,  
    levelist       = 1000/850/700/500/400/300,  
    param          = 129,  
    date           = 19990222,  
    time           = 0000/0600/1200/1800,  
    target         = "analysis.[time]"
```

6.4 Multiple requests in one call to MARS

MARS can accept more than one request in the same invocation. This allows users 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.

There are some considerations when using multiple requests:

Parameter Inheritance

When two or more `retrieve` requests are used in a single call to MARS, parameters not explicitly set in the second request are inherited from the first (and so on). 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/60,  
    param= temperature,  
    target= analysis.ml
```

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

There can be cases in which unwanted parameters are inherited. They can be removed from subsequent requests by specifying 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 for 40 days ago and append it at the end of the target file.

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

6.5 Fieldset

It is a temporary storage for fields, either retrieved from MARS or read from a UNIX file. Fieldsets can be seen as variables in a MARS request, and they can be referenced in subsequent MARS requests in the same execution. This is an example how to retrieve data into a fieldset named `analysis`:

```
retrieve,  
  class      = od,  
  type       = analysis,  
  stream     = oper,  
  expver     = 1,  
  date       = -1,  
  time       = 12,  
  levtype    = pressure levels,  
  levelist   = 1000/850/500,  
  param      = temperature,  
  fieldset   = analysis
```

At the end of the execution, all fieldsets are released. Therefore, data in unsaved fieldsets will be lost. Fieldsets can be saved as UNIX files with `write` requests, like in the example:

```
write,
    fieldset      = x,
    target        = "data"
```

Note that fieldsets are in fact using temporary disk storage on the host where MARS is executed, usually in \$TMPDIR directory. When dealing with big fieldsets, users are advised to make such variable point to a suitable directory where temporary data can be stored.

6.6 Compute

Mathematical computations can be done on retrieved grid point GRIB fields using the `compute` verb. It uses directive `fieldset` to store the result of the computation and `formula` which specifies the actual computation. Here is an example:

```
compute,
    fieldset      = Z,
    formula       = "X/2+log(Y)*X"
```

In this example, X and Y are the two input fieldsets, Z is the output fieldset.

Conventions used in this documentation:

- X upper case characters denote fieldsets
- X^k is the kth field of fieldset X
- X_{ij}^k is the point (column i, line j) of field X^k
- a lower case characters are real scalars
- '...' is a character string
- f(...) is a function call
- formula = "..." is the mars formula, what is actually computed

Any mathematical formula can be reduced to a series of elementary computations involving only one or two values. So the formula in the example can be reduced to:

```
[1] compute X/2
[2] compute log(Y)
[3] compute [2]*X
[4] compute [1]+[3]
```

For each of the fields X^k and Y^k of X and Y, a field Z^k is computed using the `formula` and added to the fieldset Z.

Mixing fields and scalars

If the two operands are two fields, the operation is carried out between each pair of corresponding grid points in the two fields. The result is a field.

If one operand is a scalar and the other a field, the operation is carried out by applying the scalar to each grid point. The result is a field.

If the two operands are both scalars, the result is a scalar.

Calling functions

The same logic applies to functions. If the argument of a function is a field, the result is a field where each grid point is the result of the function at the corresponding grid point in the input field.

If the argument is a scalar the output is also a scalar.

Examples:

```
compute,  
  fieldset      = X,  
  formula       = "A * B"
```

Creates a fieldset X where $X_{ij}^k = A_{ij}^k * B_{ij}^k$. Note that * is scalar multiplication and not matrix multiplication.

```
compute,  
  fieldset      = X,  
  formula       = "A * 2"
```

Creates a fieldset X where $X_{ij}^k = A_{ij}^k * 2$

```
compute,  
  fieldset      = X,  
  formula       = "(A>10) * B + (A<=10)*C"
```

Creates a fieldset X where:

- $X_{ij}^k = B_{ij}^k$ if $A_{ij}^k > 10$
- $X_{ij}^k = C_{ij}^k$ if $A_{ij}^k \leq 10$

Bitmaps (Missing values)

Some fields may contain a bitmap, i.e., there is no data (missing values) for certain grid points. Those missing values are not considered in any computation, but copied to the resulting field. Only valid data values are computed. Functions `bitmap` and `nobitmap` enable users to handle bitmaps on fieldsets.

Examples:

```
compute,  
  fieldset      = X,  
  formula       = "bitmap(A,0)"
```

Creates a fieldset X where:

- $X_{ij}^k = A_{ij}^k$ if A_{ij}^k not 0
- $X_{ij}^k =$ missing value if $A_{ij}^k = 0$

```
compute,  
  fieldset      = X,  
  formula       = "bitmap(A,B)"
```

If B is a field with a bitmap, this request creates a fieldset X, where X^k is a copy of A^k with bitmap of B.

```
compute,  
  fieldset      = X,  
  formula       = "nobitmap(A,1)"
```

Creates a fieldset X where:

- $X_{ij}^k = A_{ij}^k$ if A_{ij}^k not missing value
- $X_{ij}^k = 1$ if $A_{ij}^k =$ missing value

Missing Fields

Any operation involving a missing field will results in a missing field.

Operators

The following operators are implemented:

Boolean operators

- = equal to
- > greater than
- < less than
- >= greater or equal to
- <= less than or equal to
- <> not equal to

All the boolean operators return 1 if the test is true, 0 otherwise.

Arithmetic operators

- + addition
- subtraction
- * multiplication
- / division
- ^ power

The sign '-' can also be used to denote negation. In this case it is actually a function, not an operator.

Sub fieldsets

One can extract fields from a fieldset using square brackets. If x is a fieldset, $x[2]$ is the second field, $x[1,10]$ comprises the first 10 fields, and $x[2,8,2]$ comprises every second field of x starting from 2 up to 8.

Order of precedence

MARS follows the mathematical order of operations:

1. function calls
2. ^
3. * and /
4. + and -
5. boolean operators

By using brackets, one can change the order of computation.

Available functions

Name	# of param.	Param. type	Result type	Description
max	any	any	fieldset if one of the parameters is a fieldset, scalar otherwise	Maximum
min	any	any	fieldset if one of the parameters is a fieldset, scalar otherwise	Minimum
merge	any	fieldset	fieldset	Merges several fieldsets into a unique fieldset
maxvalue	1	fieldset	scalar	Maximum grid point value of all fields
minvalue	1	fieldset	scalar	Minimum grid point value of all fields
sgn	1	any	same as parameter	0 if the value is 0, +1 if the value is positive, -1 if the value is negative.
int	1	any	same as parameter	Integer part
exp	1	any	same as parameter	Exponential
log	1	any	same as parameter	Natural logarithm
sin	1	any	same as parameter	Sine
cos	1	any	same as parameter	Cosine
tan	1	any	same as parameter	Tangent
asin	1	any	same as parameter	Arc-sine
acos	1	any	same as parameter	Arc-cosine
atan	1	any	same as parameter	Arc-tangent
abs	1	any	same as parameter	Absolute value
sqrt	1	any	same as parameter	Square root
count	1	fieldset	scalar	Number of fields in the fieldset
sum	1	fieldset	1 field	Sum of all the fields in the fieldset.
mean	1	fieldset	1 field	Mean of all the fields in the fieldset.
rms	1	fieldset	1 field	Root mean square of all the fields in the fieldset.
stdev	1	fieldset	1 field	Standard deviation of all the fields in the fieldset.
bitmap	2	fieldset, scalar	fieldset	Creates missing values for those grid points equal to scalar
bitmap	2	fieldset, field	fieldset	Creates missing values for those missing values of the second field

Name	# of param.	Param. type	Result type	Description
bitmap	2	fieldset, fieldset	fieldset	Creates missing values on the first parameter for those grid points with missing values of the second fieldset
nobitmap	2	fieldset, scalar	fieldset	Removes bitmap on the fieldset by changing missing values by the scalar provided
repeat	2	field, scalar	fieldset	Repeats field as many times as indicated by scalar

Warnings

No check is done on the type of data combined in the formulae. It is the user's responsibility to make sure the computations have a physical meaning.

GRIB header of the resulting fields is copied from the first field encountered. The resulting fields' GRIB headers are marked as being created with the `compute` facility by setting the model identification number to 255. Users can preserve the model identification number by setting the environment variable `MARS_COMPUTE_FLAG` to 0 before issuing the call to `MARS`.

Example: Compute wind speed

```

retrieve,
    type           = analysis,
    time           = 12,
    date           = -1,
    grid           = 2.5/2.5,
    param          = u,
    fieldset       = u

retrieve,
    param          = v,
    fieldset       = v

compute,
    formula        = "sqrt(u*u + v*v)",
    fieldset       = speed

write,
    fieldset       = speed,
    target         = "speed"

```