Migration to GRIB2 -> 50r2

UEF2026

Sébastien Villaume

GRIB2 Migration technical lead

ECMWF

Questions at any time via Slido

www.slido.com

#2414240

Context

-> GRIB1 has been deprecated by WMO more than 2 decades ago.

-> ECMWF is required to publish in GRIB2 a set of core parameters on the WIS2 by March 2027.

-> future horizontal resolution increase of IFS will likely hit the GRIB1 limitations.

-> In 2011, IFS increased the number of model levels above 125 levels (137 levels) -> only model levels were migrated.

-> recommendation of the TAC subgroup on the direction of GRIB at ECMWF

Context

-> GRIB1 has been deprecated by WMO more than 2 decades ago.

-> ECMWF is required to publish in GRIB2 a set of core parameters on the WIS2 by March 2027.

-> future horizontal resolution increase of IFS will likely hit the GRIB1 limitations.

-> In 2011, IFS increased the number of model levels above 125 levels (137 levels) -> only model levels were migrated.

-> recommendation of the TAC subgroup on the direction of GRIB at ECMWF

Context

-> GRIB1 has been deprecated by WMO more than 2 decades ago.

-> ECMWF is required to publish in GRIB2 a set of core parameters on the WIS2 by March 2027.

-> future horizontal resolution increase of IFS will likely hit the GRIB1 limitations.

-> In 2011, IFS increased the number of model levels above 125 levels (137 levels) -> only model levels were migrated.

-> recommendation of the TAC subgroup on the direction of GRIB at ECMWF

Context

-> GRIB1 has been deprecated by WMO more than 2 decades ago.

-> ECMWF is required to publish in GRIB2 a set of core parameters on the WIS2 by March 2027.

-> future horizontal resolution increase of IFS will likely hit the GRIB1 limitations.

-> In 2011, IFS increased the number of model levels above 125 levels (137 levels) -> only model levels were migrated.

-> recommendation of the TAC subgroup on the direction of GRIB at ECMWF

Context

-> GRIB1 has been deprecated by WMO more than 2 decades ago.

-> ECMWF is required to publish in GRIB2 a set of core parameters on the WIS2 by March 2027.

-> future horizontal resolution increase of IFS will likely hit the GRIB1 limitations.

-> In 2011, IFS increased the number of model levels above 125 levels (137 levels) -> only model levels were migrated.

-> recommendation of the TAC subgroup on the direction of GRIB at ECMWF

Context

-> GRIB1 has been deprecated by WMO more than 2 decades ago.

-> ECMWF is required to publish in GRIB2 a set of core parameters on the WIS2 by March 2027.

-> future horizontal resolution increase of IFS will likely hit the GRIB1 limitations.

-> In 2011, IFS increased the number of model levels above 125 levels (137 levels) -> only model levels were migrated.

-> recommendation of the TAC subgroup on the direction of GRIB at ECMWF

Terms of reference

-> Scope

  -> Migrate all parameters

  -> Only new data produced, no conversion of existing data

  -> Migrate our entire software stack: I/O routines in models, interpolation, plotting, indexing, archiving, dissemination

-> Requirements

  -> Minimize disruptions to our operations and to our users

  -> Maintain consistency with existing data

  -> Provide tools to ease the migration

  -> Take advantage of the rich metadata

  -> Support Member states

  -> Fully WMO compliant, avoid local encoding

Terms of reference

-> Scope

-> Migrate all parameters

-> Only new data produced, no conversion of existing data

-> Migrate our entire software stack: I/O routines in models, interpolation, plotting, indexing, archiving, dissemination

-> Requirements

-> Minimize disruptions to our operations and to our users

-> Maintain consistency with existing data

-> Provide tools to ease the migration

-> Take advantage of the rich metadata

-> Support Member states

-> Fully WMO compliant, avoid local encoding

Terms of reference

-> Scope

-> Migrate all parameters

-> Only new data produced, no conversion of existing data

-> Migrate our entire software stack: I/O routines in models, interpolation, plotting, indexing, archiving, dissemination

-> Requirements

-> Minimize disruptions to our operations and to our users

-> Maintain consistency with existing data

-> Provide tools to ease the migration

-> Take advantage of the rich metadata

-> Support Member states

-> Fully WMO compliant, avoid local encoding

Migration preparation: New templates, new parameters

-> Templates (~80 new templates added to GRIB2)

- -> Wave spectra (frequency/direction) and wave period templates
- -> Anomalies, EFI, SOT templates
- -> Tile templates for land surface modelling
- -> Probability templates with focal statistics
- -> Quantiles
- -> Large ensembles (>255 members)

-> Parameters

- -> Heatwaves/thermal stress related parameters
- -> Soil/snow/ice model levels/layers
- -> Hydrological parameters
- -> Ocean, ice and waves
- -> Destination Earth and ERA6 parameters

Migration preparation: New templates, new parameters

-> Templates (~80 new templates added to GRIB2)

- -> Wave spectra (frequency/direction) and wave period templates
- -> Anomalies, EFI, SOT templates
- -> Tile templates for land surface modelling
- -> Probability templates with focal statistics
- -> Quantiles
- -> Large ensembles (>255 members)

-> Parameters

- -> Heatwaves/thermal stress related parameters
- -> Soil/snow/ice model levels/layers
- -> Hydrological parameters
- -> Ocean, ice and waves
- -> Destination Earth and ERA6 parameters

Migration preparation: New templates, new parameters

-> Templates (~80 new templates added to GRIB2)

- -> Wave spectra (frequency/direction) and wave period templates
- -> Anomalies, EFI, SOT templates
- -> Tile templates for land surface modelling
- -> Probability templates with focal statistics
- -> Quantiles
- -> Large ensembles (>255 members)

-> Parameters

- -> Heatwaves/thermal stress related parameters
- -> Soil/snow/ice model levels/layers
- -> Hydrological parameters
- -> Ocean, ice and waves
- -> Destination Earth and ERA6 parameters

GRIB2

-> richer, more explicit metadata and parameter semantics

-> extensible template framework for new products and encodings

-> stronger WMO alignment for interoperable data exchange

-> cleaner indexing, filtering, and parameter identification

GRIB2

-> richer, more explicit metadata and parameter semantics

-> extensible template framework for new products and encodings

-> stronger WMO alignment for interoperable data exchange

-> cleaner indexing, filtering, and parameter identification

GRIB2

-> richer, more explicit metadata and parameter semantics

-> extensible template framework for new products and encodings

-> stronger WMO alignment for interoperable data exchange

-> cleaner indexing, filtering, and parameter identification

GRIB2

-> richer, more explicit metadata and parameter semantics

-> extensible template framework for new products and encodings

-> stronger WMO alignment for interoperable data exchange

-> cleaner indexing, filtering, and parameter identification

GRIB2

-> richer, more explicit metadata and parameter semantics

-> extensible template framework for new products and encodings

-> stronger WMO alignment for interoperable data exchange

-> cleaner indexing, filtering, and parameter identification

Re-designing `paramId` & `shortName`

probability of accumulated total precipitation at surface of at least 25mm

Re-designing `paramId` & `shortName`

probability of accumulated total precipitation at surface of at least 25mm

GRIB1 (ECMWF - local)

table2Version=131 (technical - no meaning)
indicatorOfParameter=98 (probability of accumulated total precipitation at surface of at least 25mm)
timeRangeIndicator=4 (accumulation)
typeOfLevel=1 (surface)

Re-designing `paramId` & `shortName`

probability of accumulated total precipitation at surface of at least 25mm	
GRIB1 (ECMWF - local)	GRIB2 (WMO)
table2Version=131 (technical - no meaning) indicatorOfParameter=98 (probability of accumulated total precipitation at surface of at least 25mm) timeRangeIndicator=4 (accumulation) typeOfLevel=1 (surface)	productDefinitionTemplateNumber = 9 (probability) discipline = 0 (meteorology) parameterCategory = 1 (Moisture) parameterNumber = 52 (Total* precipitation rate) typeOfFirstFixedSurface = 1 (surface) typeOfStatisticalProcessing = 1 (Accumulation) probabilityType = 3 (above lower limit) scaledValueOfLowerLimit = 25 scaleFactorOfLowerLimit = 0 (no scaling)

Re-designing `paramId` & `shortName`

probability of accumulated total precipitation at surface of at least 25mm	
GRIB1 (ECMWF - local)	**GRIB2 (WMO)**
table2Version=131 (technical - no meaning) indicatorOfParameter=98 (probability of accumulated total precipitation at surface of at least 25mm) timeRangeIndicator=4 (accumulation) typeOfLevel=1 (surface)	productDefinitionTemplateNumber = 9 (probability) discipline = 0 (meteorology) parameterCategory = 1 (Moisture) parameterNumber = 52 (Total* precipitation rate) typeOfFirstFixedSurface = 1 (surface) typeOfStatisticalProcessing = 1 (Accumulation) probabilityType = 3 (above lower limit) scaledValueOfLowerLimit = 25 scaleFactorOfLowerLimit = 0 (no scaling)

In ecCodes, these 2 GRIB representations share:

- the same `paramId`: 131098
- the same `shortName`: tpg25

Parameters changes across the migration

-> All parameters containing a time range will be deprecated:

  -> In the last XX hours

  -> Since previous post-processing

  -> The time range will be handled by the new keyword `timespan`

-> Parameters without canonical WMO units will be deprecated

  -> Replaced by variants with WMO units

-> Some parameters will be indexed differently in MARS:

  -> Multi levels/layers:

    -> A single Soil temperature parameter on soil levels 1 to 4, rather than four parameters on level type `surface`

  -> Parameters at a specific height in metres: 50m, 100m, 200m, etc.

    -> Single generic parameter on the new levtype height level "hl" rather than many different parameters on levtype sfc.

    -> BUT parameters <=10m will not change, e.g. 2t and 10u/v

Parameters changes across the migration

-> All parameters containing a time range will be deprecated:

  -> In the last XX hours

  -> Since previous post-processing

  -> The time range will be handled by the new keyword `timespan`

-> Parameters without canonical WMO units will be deprecated

  -> Replaced by variants with WMO units

-> Some parameters will be indexed differently in MARS:

  -> Multi levels/layers:

    -> A single Soil temperature parameter on soil levels 1 to 4, rather than four parameters on level type `surface`

  -> Parameters at a specific height in metres: 50m, 100m, 200m, etc.

    -> Single generic parameter on the new levtype height level "hl" rather than many different parameters on levtype sfc.

    -> **BUT** parameters <=10m will not change, e.g. 2t and 10u/v

Parameters changes across the migration

-> All parameters containing a time range will be deprecated:

-> In the last XX hours

-> Since previous post-processing

-> The time range will be handled by the new keyword `timespan`

-> Parameters without canonical WMO units will be deprecated

-> Replaced by variants with WMO units

-> Some parameters will be indexed differently in MARS:

-> Multi levels/layers:

-> A single Soil temperature parameter on soil levels 1 to 4, rather than four parameters on level type `surface`

-> Parameters at a specific height in metres: 50m, 100m, 200m, etc.

-> Single generic parameter on the new levtype height level "hl" rather than many different parameters on levtype sfc.

-> **BUT** parameters <=10m will not change, e.g. 2t and 10u/v

Parameters changes across the migration

-> All parameters containing a time range will be deprecated:

-> In the last XX hours

-> Since previous post-processing

-> The time range will be handled by the new keyword `timespan`

-> Parameters without canonical WMO units will be deprecated

-> Replaced by variants with WMO units

-> Some parameters will be indexed differently in MARS:

-> Multi levels/layers:

-> A single Soil temperature parameter on soil levels 1 to 4, rather than four parameters on level type `surface`

-> Parameters at a specific height in metres: 50m, 100m, 200m, etc.

-> Single generic parameter on the new levtype height level "hl" rather than many different parameters on levtype sfc.

-> **BUT** parameters <=10m will not change, e.g. 2t and 10u/v

Parameters changes across the migration

-> All parameters containing a time range will be deprecated:

-> In the last XX hours

-> Since previous post-processing

-> The time range will be handled by the new keyword `timespan`

-> Parameters without canonical WMO units will be deprecated

-> Replaced by variants with WMO units

-> Some parameters will be indexed differently in MARS:

-> Multi levels/layers:

-> A single Soil temperature parameter on soil levels 1 to 4, rather than four parameters on level type `surface`

-> Parameters at a specific height in metres: 50m, 100m, 200m, etc.

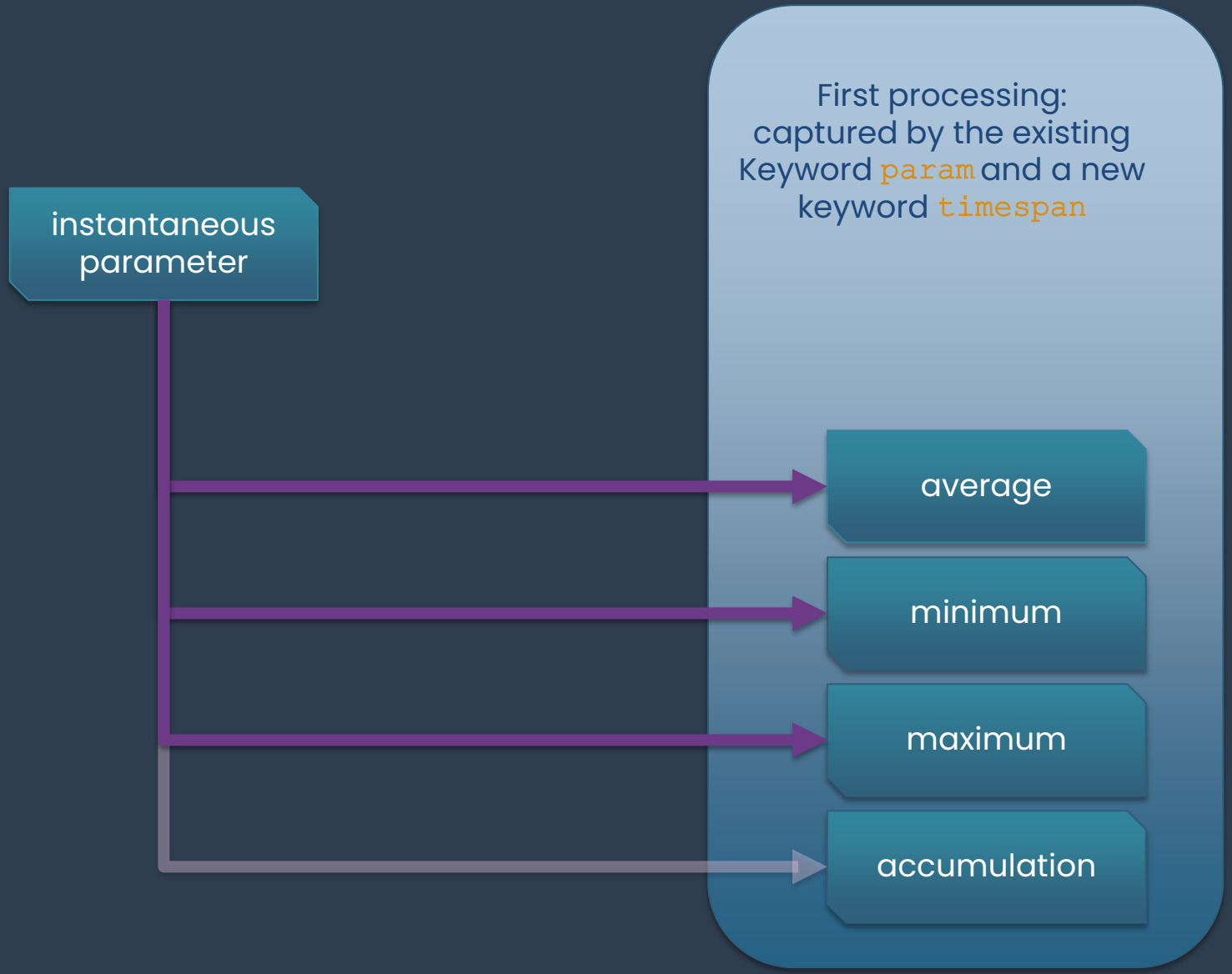
-> Single generic parameter on the new levtype height level "hl" rather than many different parameters on levtype sfc.

-> **BUT** parameters <=10m will not change, e.g. 2t and 10u/v

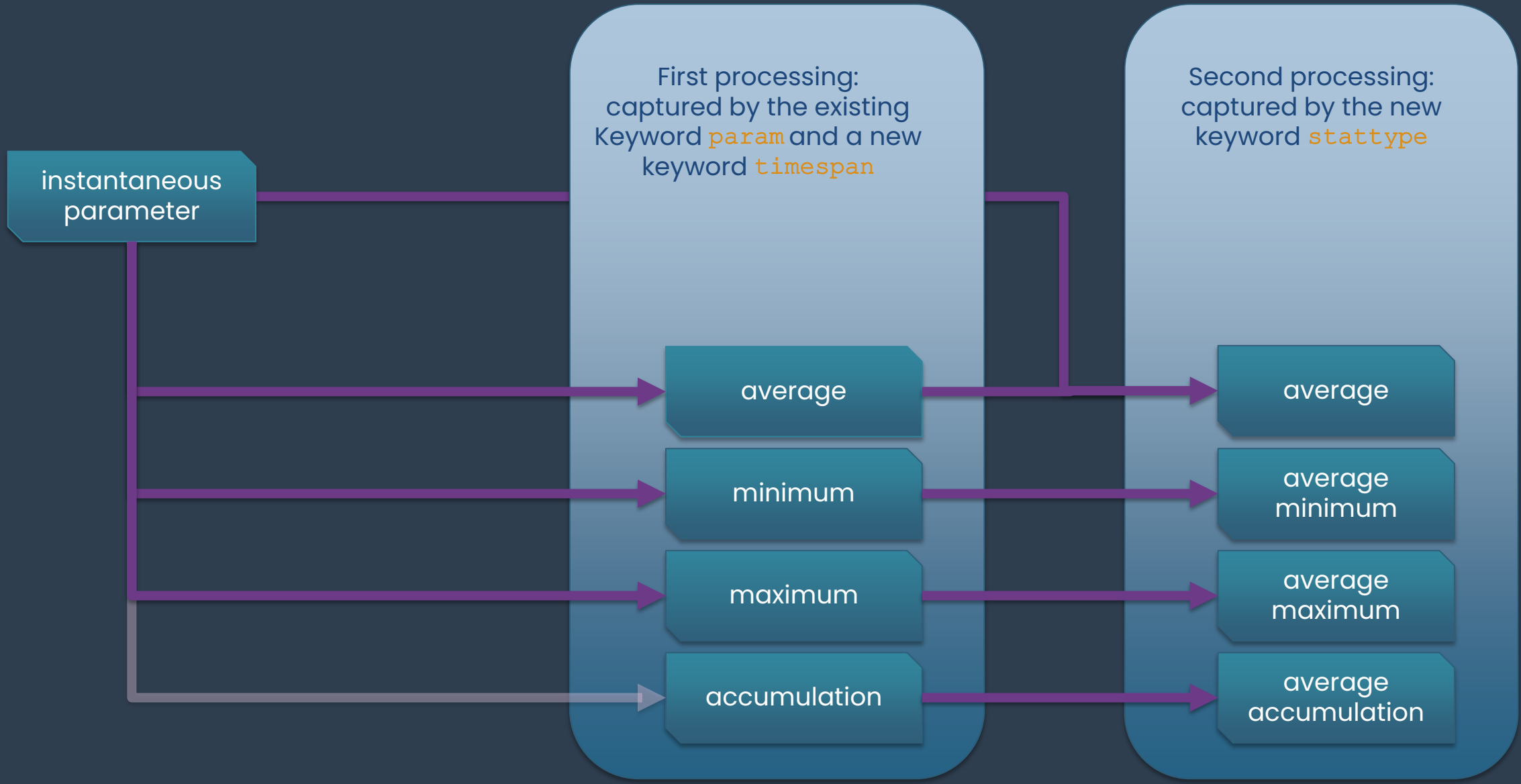
New model for time processing

instantaneous parameter

New model for time processing



New model for time processing



Standardisation of `shortName` and time post-processing

-> The `shortName` will be standardized and use prefixes to indicate their first level of type of time processing:

2t -> min_2t | max_2t | avg_2t

-> The span of the first level of time processing is captured by a new keyword `timespan`:

-> `timespan=1h` (hourly),

-> `timespan=3h` (3-hourly) ,

-> `timespan=from-start` (special, for accumulations)

-> Any additional time processing (type and length) will be captured a new keyword `stattype`:

-> `statype=` Monthly means, monthly maxima, monthly maxima

-> Similar `stattype` for weekly means, daily means, 5-days maximum, etc.

-> This will allow future development of other processing and gives a flexible and generalisable structure.

Standardisation of `shortName` and time post-processing

-> The `shortName` will be standardized and use prefixes to indicate their first level of type of time processing:

2t -> min_2t | max_2t | avg_2t

-> The span of the first level of time processing is captured by a new keyword `timespan`:

-> `timespan=1h` (hourly),

-> `timespan=3h` (3-hourly) ,

-> `timespan=from-start` (special, for accumulations)

-> Any additional time processing (type and length) will be captured a new keyword `stattype`:

-> `statype=` Monthly means, monthly maxima, monthly maxima

-> Similar `stattype` for weekly means, daily means, 5-days maximum, etc.

-> This will allow future development of other processing and gives a flexible and generalisable structure.

Standardisation of `shortName` and time post-processing

-> The `shortName` will be standardized and use prefixes to indicate their first level of type of time processing:

2t -> min_2t | max_2t | avg_2tq2

-> The span of the first level of time processing is captured by a new keyword `timespan`:

-> `timespan=1h` (hourly),

-> `timespan=3h` (3-hourly) ,

-> `timespan=from-start` (special, for accumulations)

-> Any additional time processing (type and length) will be captured a new keyword `stattype`:

-> `statype=` Monthly means, monthly maxima, monthly maxima

-> Similar `stattype` for weekly means, daily means, 5-days maximum, etc.

-> This will allow future development of other processing and gives a flexible and generalisable structure.

Standardisation of `shortName` and time post-processing

-> The `shortName` will be standardized and use prefixes to indicate their first level of type of time processing:

2t -> min_2t | max_2t | avg_2tq2

-> The span of the first level of time processing is captured by a new keyword `timespan`:

-> `timespan=1h` (hourly),

-> `timespan=3h` (3-hourly) ,

-> `timespan=from-start` (special, for accumulations)

-> Any additional time processing (type and length) will be captured a new keyword `stattype`:

-> `statype=` Monthly means, monthly maxima, monthly maxima

-> Similar `stattype` for weekly means, daily means, 5-days maximum, etc.

-> This will allow future development of other processing and gives a flexible and generalisable structure.

Standardisation of `shortName` and time post-processing

-> The `shortName` will be standardized and use prefixes to indicate their first level of type of time processing:

2t -> min_2t | max_2t | avg_2tq2

-> The span of the first level of time processing is captured by a new keyword `timespan`:

-> `timespan=1h` (hourly),

-> `timespan=3h` (3-hourly) ,

-> `timespan=from-start` (special, for accumulations)

-> Any additional time processing (type and length) will be captured a new keyword `stattype`:

-> `statype=` Monthly means, monthly maxima, monthly maxima

-> Similar `stattype` for weekly means, daily means, 5-days maximum, etc.

-> This will allow future development of other processing and gives a flexible and generalisable structure.

Standardisation of `shortName` and time post-processing

-> The `shortName` will be standardized and use prefixes to indicate their first level of type of time processing:

2t -> min_2t | max_2t | avg_2tq2

-> The span of the first level of time processing is captured by a new keyword `timespan`:

-> `timespan=1h` (hourly),

-> `timespan=3h` (3-hourly) ,

-> `timespan=from-start` (special, for accumulations)

-> Any additional time processing (type and length) will be captured a new keyword `stattype`:

-> `statype=` Monthly means, monthly maxima, monthly maxima

-> Similar `stattype` for weekly means, daily means, 5-days maximum, etc.

-> This will allow future development of other processing and gives a flexible and generalisable structure.

Consequence: steps are simpler! step=0-3/3-6/6-9 -> step=3/6/9,timespan=3h

Handling legacy encoding (ongoing work)

-> GRIB2 prescribes the units of parameters

  -> Precipitation in kg m^{-2} in GRIB2 vs m in 50r1

  -> Cloud cover in % in GRIB2 vs fraction in 50r1

  ....

-> Conversion is luckily straightforward

  -> 1 kg m^{-2} ~ 1mm

  -> Conversion factor ~10^3

-> In GRIB1 (and sometimes legacy GRIB2), instantaneous `paramId`s have been used to encode monthly means -> we break this practice

  -> Special handling to continue to decode these parameters correctly

  -> Prevent writing more of these: some parameters become read-only

Handling legacy encoding (ongoing work)

-> GRIB2 prescribes the units of parameters

-> Precipitation in kg m^{-2} in GRIB2 vs m in 50r1

-> Cloud cover in % in GRIB2 vs fraction in 50r1

....

-> Conversion is luckily straightforward

-> 1 kg m^{-2} ~ 1mm

-> Conversion factor ~10^3

-> In GRIB1 (and sometimes legacy GRIB2), instantaneous `paramId`s have been used to encode monthly means -> we break this practice

-> Special handling to continue to decode these parameters correctly

-> Prevent writing more of these: some parameters become read-only

Handling legacy encoding (ongoing work)

-> GRIB2 prescribes the units of parameters

-> Precipitation in kg m^{-2} in GRIB2 vs m in 50r1

-> Cloud cover in % in GRIB2 vs fraction in 50r1

....

-> Conversion is luckily straightforward

-> 1 kg m^{-2} ~ 1mm

-> Conversion factor ~10^3

-> In GRIB1 (and sometimes legacy GRIB2), instantaneous `paramId`s have been used to encode monthly means -> we break this practice

-> Special handling to continue to decode these parameters correctly

-> Prevent writing more of these: some parameters become read-only

Handling legacy encoding (ongoing work)

-> GRIB2 prescribes the units of parameters

-> Precipitation in kg m^{-2} in GRIB2 vs m in 50r1

-> Cloud cover in % in GRIB2 vs fraction in 50r1

....

-> Conversion is luckily straightforward

-> 1 kg m^{-2} ~ 1mm

-> Conversion factor ~10^3

-> In GRIB1 (and sometimes legacy GRIB2), instantaneous `paramId`s have been used to encode monthly means -> we break this practice

-> Special handling to continue to decode these parameters correctly

-> Prevent writing more of these: some parameters become read-only

Wave streams

-> discontinued from 50r2

-> Move regular wave parameters and 2D wave spectra into respective atm streams:

`wave` -> `oper` , `waef` -> `enfo`, etc.

-> Still logically separated from atmo parameters in the MARS catalogue using a `paramtype` node in the MARS catalogue

-> `paramtype` = `base` -> atmospheric parameters

-> `paramtype` = `wave` -> regular wave parameters

-> `paramtype` = `wave_spectra` -> 2D wave spectra

-> Other use of `paramtype`: `chemical`, `optical`, `tile`, `etc.`

Wave streams

-> discontinued from 50r2

-> Move regular wave parameters and 2D wave spectra into respective atm streams:

`wave` -> `oper` , `waef` -> `enfo`, etc.

-> Still logically separated from atmo parameters in the MARS catalogue using a `paramtype` node in the MARS catalogue

-> `paramtype` = `base` -> atmospheric parameters

-> `paramtype` = `wave` -> regular wave parameters

-> `paramtype` = `wave_spectra` -> 2D wave spectra

-> Other use of `paramtype`: `chemical`, `optical`, `tile`, `etc.`

Wave streams

-> discontinued from 50r2

-> Move regular wave parameters and 2D wave spectra into respective atm streams:

`wave` -> `oper` , `waef` -> `enfo`, etc.

-> Still logically separated from atmo parameters in the MARS catalogue using a `paramtype` node in the MARS catalogue

-> `paramtype` = `base` -> atmospheric parameters

-> `paramtype` = `wave` -> regular wave parameters

-> `paramtype` = `wave_spectra` -> 2D wave spectra

-> Other use of `paramtype`: `chemical`, `optical`, `tile`, `etc.`

Wave streams

-> discontinued from 50r2

-> Move regular wave parameters and 2D wave spectra into respective atm streams:

`wave` -> `oper` , `waef` -> `enfo`, etc.

-> Still logically separated from atmo parameters in the MARS catalogue using a `paramtype` node in the MARS catalogue

-> `paramtype` = `base` -> atmospheric parameters

-> `paramtype` = `wave` -> regular wave parameters

-> `paramtype` = `wave_spectra` -> 2D wave spectra

-> Other use of `paramtype`: `chemical`, `optical`, `tile`, `etc.`

Wave streams

-> discontinued from 50r2

-> Move regular wave parameters and 2D wave spectra into respective atm streams:

`wave` -> `oper` , `waef` -> `enfo`, etc.

-> Still logically separated from atmo parameters in the MARS catalogue using a `paramtype` node in the MARS catalogue

-> `paramtype` = `base` -> atmospheric parameters

-> `paramtype` = `wave` -> regular wave parameters

-> `paramtype` = `wave_spectra` -> 2D wave spectra

-> Other use of `paramtype`: `chemical`, `optical`, `tile`, `etc.`

`ChemId` for atmospheric composition

-> The number of `paramId`s used in atmospheric composition is problematic

-> Conceptually these parameters are 2-dimensional:

  -> A physical property -> mass mixing ratio, emission rate, etc.

  -> chemical species and aerosols -> CH_4 , CO_2

-> We are splitting the existing `paramId`s into a pair of keywords:

  -> `paramId` : now only representing the property

  -> `chemId` : representing the chemical species and aerosols

  -> Example: `param`=mmr , `chem`=CH4

-> We also considered extracting the emission sources into a separate keyword but decided not to do so

  -> Example: `param`=emission from volcanoes vs ~~`param`=emission, `source`=volcanoes~~

ChemId for atmospheric composition

-> The number of paramIds used in atmospheric composition is problematic

-> Conceptually these parameters are 2-dimensional:

-> A physical property -> mass mixing ratio, emission rate, etc.

-> chemical species and aerosols -> CH_4 , CO_2

-> We are splitting the existing paramIds into a pair of keywords:

-> paramId : now only representing the property

-> chemId : representing the chemical species and aerosols

-> Example: param=mmr , chem=CH4

-> We also considered extracting the emission sources into a separate keyword but decided not to do so

-> Example: param=emission from volcanoes vs ~~param=emission, source=volcanoes~~

ECMWF

ChemId for atmospheric composition

-> The number of paramIds used in atmospheric composition is problematic

-> Conceptually these parameters are 2-dimensional:

  -> A physical property -> mass mixing ratio, emission rate, etc.

  -> chemical species and aerosols -> CH_4 , CO_2

-> We are splitting the existing paramIds into a pair of keywords:

  -> paramId : now only representing the property

  -> chemId : representing the chemical species and aerosols

  -> Example: param=mmr , chem=CH4

-> We also considered extracting the emission sources into a separate keyword but decided not to do so

  -> Example: param=emission from volcanoes vs ~~param=emission, source=volcanoes~~

`ChemId` for atmospheric composition

-> The number of `paramId`s used in atmospheric composition is problematic

-> Conceptually these parameters are 2-dimensional:

  -> A physical property -> mass mixing ratio, emission rate, etc.

  -> chemical species and aerosols -> CH_4 , CO_2

-> We are splitting the existing `paramId`s into a pair of keywords:

  -> `paramId` : now only representing the property

  -> `chemId` : representing the chemical species and aerosols

  -> Example: `param`=mmr , `chem`=CH4

-> We also considered extracting the emission sources into a separate keyword but decided not to do so

  -> Example: `param`=emission from volcanoes vs ~~`param`=emission, `source`=volcanoes~~

ChemId for atmospheric composition

-> The number of paramIds used in atmospheric composition is problematic

-> Conceptually these parameters are 2-dimensional:

  -> A physical property -> mass mixing ratio, emission rate, etc.

  -> chemical species and aerosols -> CH_4 , CO_2

-> We are splitting the existing paramIds into a pair of keywords:

  -> paramId : now only representing the property

  -> chemId : representing the chemical species and aerosols

  -> Example: param=mmr , chem=CH4

-> We also considered extracting the emission sources into a separate keyword but decided not to do so

  -> Example: param=emission from volcanoes vs ~~param=emission, source=volcanoes~~

`ChemId` for atmospheric composition

-> The number of `paramId`s used in atmospheric composition is problematic

-> Conceptually these parameters are 2-dimensional:

  -> A physical property -> mass mixing ratio, emission rate, etc.

  -> chemical species and aerosols -> CH_4 , CO_2

-> We are splitting the existing `paramId`s into a pair of keywords:

  -> `paramId` : now only representing the property

  -> `chemId` : representing the chemical species and aerosols

  -> Example: `param`=mmr , `chem`=CH4

-> We also considered extracting the emission sources into a separate keyword but decided not to do so

  -> Example: `param`=emission from volcanoes vs ~~`param`=emission, `source`=volcanoes~~

under `paramtype=chemical`

`Wavelength` keyword for optical parameters

-> We are introducing a new `wavelength` keyword to capture the dependency of the optical parameters into a separate degree of freedom.

-> This will have an impact on the existing optical parameters, particularly in atmospheric composition:

  -> optical depths, etc.

  -> Parameters “at XXX nm”

-> It will support other IFS developments: simulated satellite images, etc.

  -> `Wavelength` ranges

  -> Colour channels

  -> etc.

`Wavelength` keyword for optical parameters

-> We are introducing a new `wavelength` keyword to capture the dependency of the optical parameters.

-> This will have an impact on the existing optical parameters, particularly in atmospheric composition:

  -> optical depths, etc.

  -> Parameters “at XXX nm”

-> It will support other IFS developments: simulated satellite images, etc.

  -> `Wavelength` ranges

  -> Colour channels

  -> etc.

`Wavelength` keyword for optical parameters

-> We are introducing a new `wavelength` keyword to capture the dependency of the optical parameters.

-> This will have an impact on the existing optical parameters, particularly in atmospheric composition:

- -> optical depths, etc.
- -> Parameters “at XXX nm”

-> It will support other IFS developments: simulated satellite images, etc.

- -> `Wavelength` ranges
- -> Colour channels
- -> etc.

`Wavelength` keyword for optical parameters

-> We are introducing a new `wavelength` keyword to capture the dependency of the optical parameters.

-> This will have an impact on the existing optical parameters, particularly in atmospheric composition:

  -> optical depths, etc.

  -> Parameters “at XXX nm”

-> It will support other IFS developments: simulated satellite images, etc.

  -> `Wavelength` ranges

  -> Colour channels

  -> etc.

`Wavelength` keyword for optical parameters

-> We are introducing a new `wavelength` keyword to capture the dependency of the optical parameters.

-> This will have an impact on the existing optical parameters, particularly in atmospheric composition:

  -> optical depths, etc.

  -> Parameters "at XXX nm"

-> It will support other IFS developments: simulated satellite images, etc.

  -> `Wavelength` ranges

  -> Colour channels

  -> etc.

under `paramtype=optical`

`modelName` and `modelVersion`

-> We are introducing 2 new ecCodes helper keys to enrich the metadata:

  -> `modelName` (IFS, AIFS, EFAS, NEMO, etc.)

  -> `modelVersion` (cy50r2, v2.0, etc.)

-> In specific cases, these can be used as a MARS keyword (model)

  -> Land Data Assimilation System (LDAS)

  -> Hydrology models

  -> wildfire prediction models

  -> Etc.

`modelName` and `modelVersion`

-> We are introducing 2 new ecCodes helper keys to enrich the metadata:

  -> `modelName` (IFS, AIFS, EFAS, NEMO, etc.)

  -> `modelVersion` (cy50r2, v2.0, etc.)

-> In specific cases, these can be used as a MARS keyword (model)

  -> Land Data Assimilation System (LDAS)

  -> Hydrology models

  -> wildfire prediction models

  -> Etc.

`modelName` and `modelVersion`

-> We are introducing 2 new ecCodes helper keys to enrich the metadata:

  -> `modelName` (IFS, AIFS, EFAS, NEMO, etc.)

  -> `modelVersion` (cy50r2, v2.0, etc.)

-> In specific cases, these can be used as a MARS keyword (model)

  -> Land Data Assimilation System (LDAS)

  -> Hydrology models

  -> wildfire prediction models

  -> Etc.

Tools to support the migration

-> Reference Static Dataset

  -> First prototype available – will be incrementally updated to cover all operational data.

  -> Available on the HPC: `/ec/vol/marsdev4/MTG2_sample_dataset`

  -> Available at `https://data.ecmwf.int/mtg2_sample/`

-> MARS request translator (soon)

  -> Provide a tool to help users migrate their requests

  -> Support conversion of dissemination requests

-> Robust conversion tool: GRIB1 -> GRIB2 (soon)

  -> We will not support the reverse conversion

-> We are available for technical discussions to answer and help member states teams and commercial customers working on the migration.

Tools to support the migration

-> Reference Static Dataset

  -> First prototype available – will be incrementally updated to cover all operational data.

  -> Available on the HPC: **`/ec/vol/marsdev4/MTG2_sample_dataset`**

  -> Available at **`https://data.ecmwf.int/mtg2_sample/`**

-> MARS request translator (soon)

  -> Provide a tool to help users migrate their requests

  -> Support conversion of dissemination requests

-> Robust conversion tool: GRIB1 -> GRIB2 (soon)

  -> We will not support the reverse conversion

-> We are available for technical discussions to answer and help member states teams and commercial customers working on the migration.

Tools to support the migration

-> Reference Static Dataset

-> First prototype available – will be incrementally updated to cover all operational data.

-> Available on the HPC: `/ec/vol/marsdev4/MTG2_sample_dataset`

-> Available at `https://data.ecmwf.int/mtg2_sample/`

-> MARS request translator (soon)

-> Provide a tool to help users migrate their requests

-> Support conversion of dissemination requests

-> Robust conversion tool: GRIB1 -> GRIB2 (soon)

-> We will not support the reverse conversion

-> We are available for technical discussions to answer and help member states teams and commercial customers working on the migration.

Tools to support the migration

-> Reference Static Dataset

-> First prototype available – will be incrementally updated to cover all operational data.

-> Available on the HPC: `/ec/vol/marsdev4/MTG2_sample_dataset`

-> Available at `https://data.ecmwf.int/mtg2_sample/`

-> MARS request translator (soon)

-> Provide a tool to help users migrate their requests

-> Support conversion of dissemination requests

-> Robust conversion tool: GRIB1 -> GRIB2 (soon)

-> We will not support the reverse conversion

-> We are available for technical discussions to answer and help member states teams and commercial customers working on the migration.

Tools to support the migration

-> Reference Static Dataset

  -> First prototype available – will be incrementally updated to cover all operational data.

  -> Available on the HPC: `/ec/vol/marsdev4/MTG2_sample_dataset`

  -> Available at `https://data.ecmwf.int/mtg2_sample/`

-> MARS request translator (soon)

  -> Provide a tool to help users migrate their requests

  -> Support conversion of dissemination requests

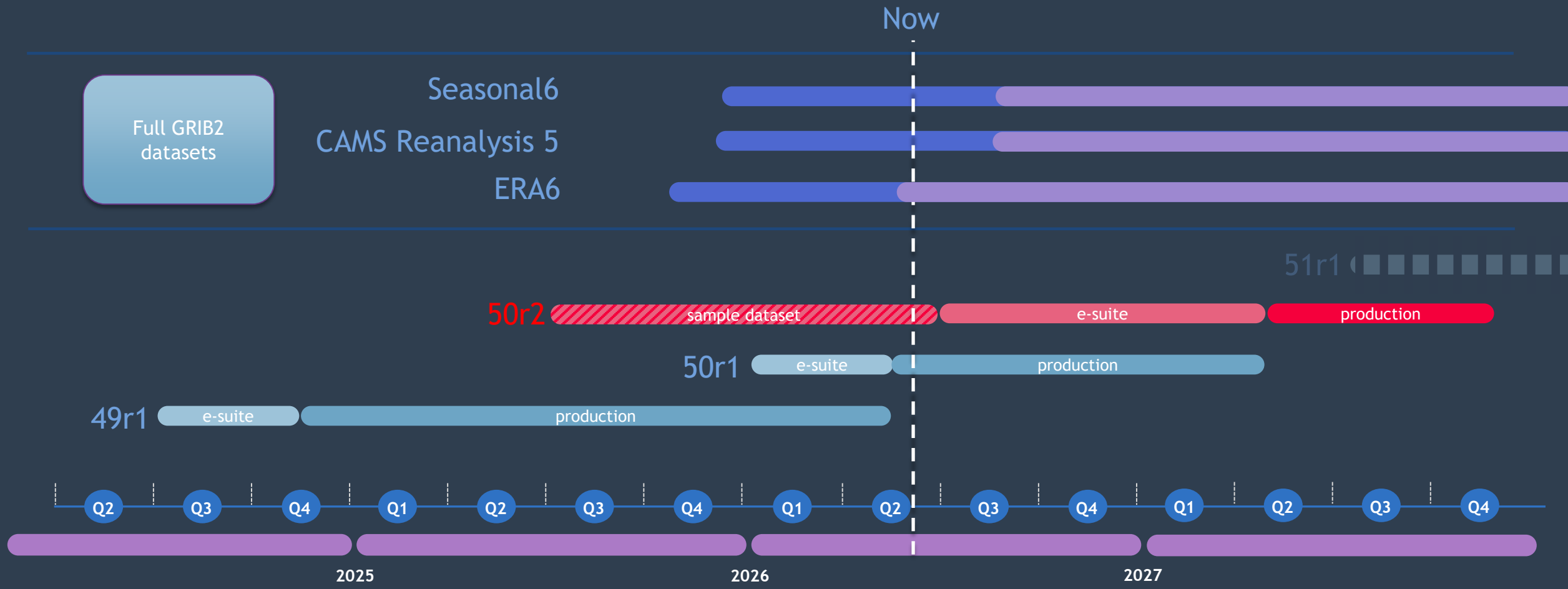
-> Robust conversion tool: GRIB1 -> GRIB2 (soon)

  -> We will not support the reverse conversion

-> We are available for technical discussions to answer and help member states teams and commercial customers working on the migration.

Timeline

50r1: scientific cycle, GRIB mix
50r2: technical cycle, full GRIB2
51r1: scientific cycle, full GRIB2



ECMWF

Timeline

ERA6 in production
not released!!! 😀

50r1: scientific cycle, GRIB mix
50r2: technical cycle, full GRIB2
51r1: scientific cycle, full GRIB2

Now

Full GRIB2 datasets

Seasonal6

CAMS Reanalysis 5

ERA6

51r1

50r2 sample dataset e-suite production

50r1 e-suite production

49r1 e-suite production

Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4

2025 2026 2027

ECMWF

StayTuned! Thank you!

ecCodes release notes

MTG2 homepage

Parameter changes

Subscribe MTG2 mailing list

MTG2 reference dataset

ECMWF