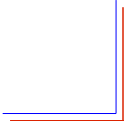




GRIB API

Anna Chiara Goglio
Palmanova, Gennaio 2017

ARPA FVG – CRMA
Centro Regionale di Modellistica Ambientale
crma@arpa.fvg.it





INDICE della presentazione



1. Codice **GRIB**

- **Struttura e caratteristiche**
- **Edizioni 1 e 2**

2. Applicativi **GRIB API**

- Definizione
- Interfacce e Struttura
- Caratteristiche
- Logica di accesso ai GRIB
- Chiavi
 - Caratteristiche
 - Esempi
- Indici

3. Interfaccia **FORTRAN 90**

- Regole generali
- Struttura dei programmi
- Funzioni disponibili
- Gestione degli errori

4. Esempi

- Estrazione campo in un punto lat/lon
- Estrazione su tutta la griglia
- Creazione e conversione file GRIB

5. Documentazione

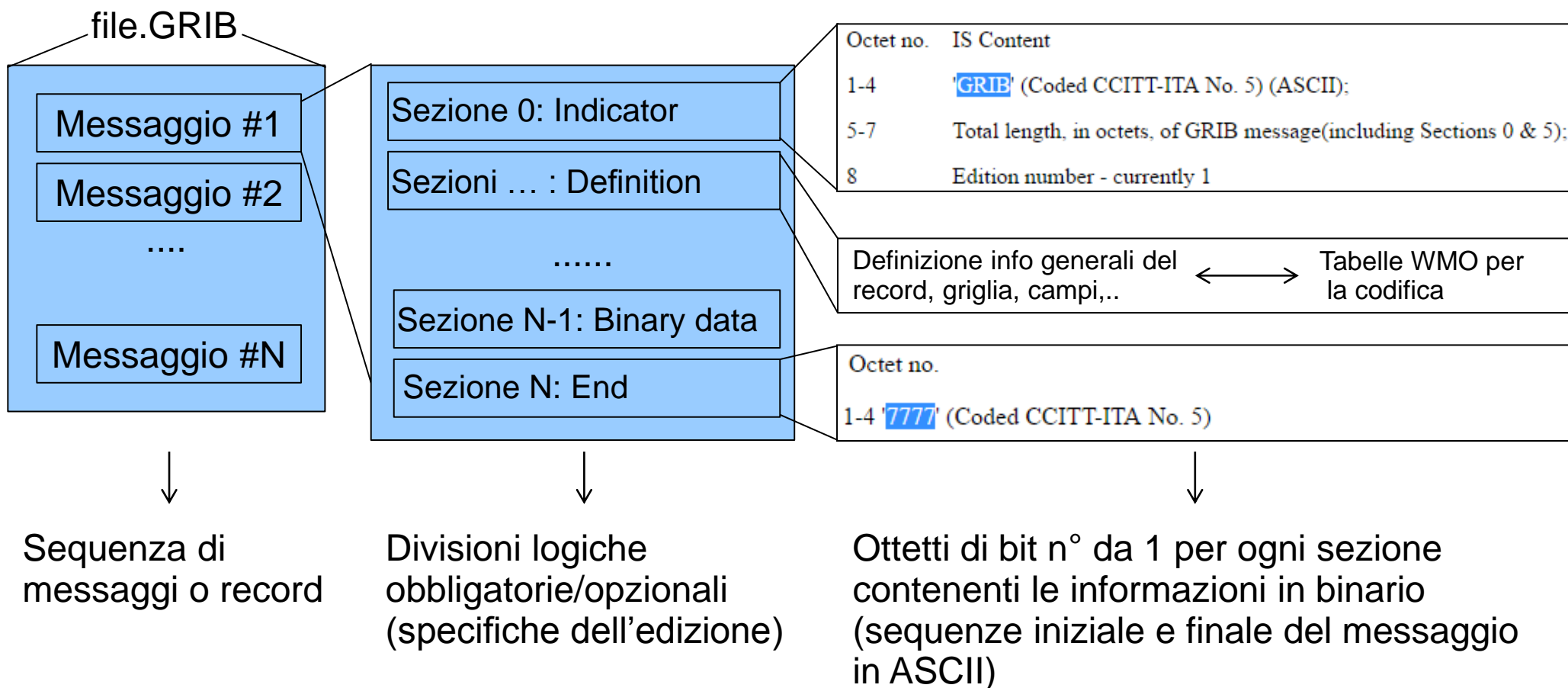


Codice GRIB (GRIdded Binary)



GRIB → Formati standard WMO (World Meteorological Organization) per la trasmissione e l'archiviazione di campi definiti su grigliato regolare

- ✓ Compatti
- ✓ Veloci da codificare/decodificare
- ✓ Auto-descrittivi
- ✓ Flessibili





Edizioni GRIB 1 e GRIB 2



GRIB 1 (1985)

GRIB1

SECTION 0 Indicator section
 SECTION 1 Product definition section
 SECTION 2 **(Grid description section)**
 SECTION 3 (Bit-map section)
 SECTION 4 Binary data section
 SECTION 5 7777 (End section)

GRIB 2 (2001)

GRIB2

SECTION 0 Indicator Section
 SECTION 1 Identification Section
 SECTION 2 (Local Use Section) }
 SECTION 3 Grid Definition Section }
 SECTION 4 Product Definition Section } (repeated)
 SECTION 5 Data Representation Section }
 SECTION 6 (Bit-map Section) }
 SECTION 7 Binary Data Section }
 SECTION 8 End Section }

Miglioramenti edizione GRIB 2

- ✓ Diversa codifica ⇒ + compatto
- ✓ Sezioni di definizione separate ⇒ + flessibile
- ✓ Possibilità di creare multi-GRIB (tempi,campi,livelli,griglie) ⇒ adatto a long-range/climate run ed ensemble run
- ✓ Codifica di errori e statistica
- ✓ Codifica di immagini radar e dati satellite
- ✓ Codifica dati spettrali
- ✓ Codifica serie temporali



INDICE della presentazione



1. Codice **GRIB**
 - Struttura e caratteristiche
 - Edizioni 1 e 2
2. Applicativi **GRIB API**
 - **Definizione**
 - **Interfacce e Struttura**
 - **Caratteristiche**
 - **Logica di accesso ai GRIB**
 - **Chiavi**
 - **Caratteristiche**
 - **Esempi**
 - **Indici**
3. Interfaccia **FORTRAN 90**
 - Regole generali
 - Struttura dei programmi
 - Funzioni disponibili
 - Gestione degli errori
4. Esempi
 - Estrazione campo in un punto lat/lon
 - Estrazione su tutta la griglia
 - Creazione e conversione file GRIB
5. Documentazione



GRIB API (Application Program Interface)

Le **GRIB API** (prodotte da ECMWF) sono applicativi per la gestione di dati in formato GRIB utilizzabili tramite varie interfacce

- ✓ **altamente performanti** → 35% di guadagno in velocità rispetto alle CDO
- ✓ **estremamente versatili** → ~ 40 funzioni per interrogare, copiare, modificare e convertire
- ✓ **molto flessibili** → Indipendenti dall'edizione GRIB

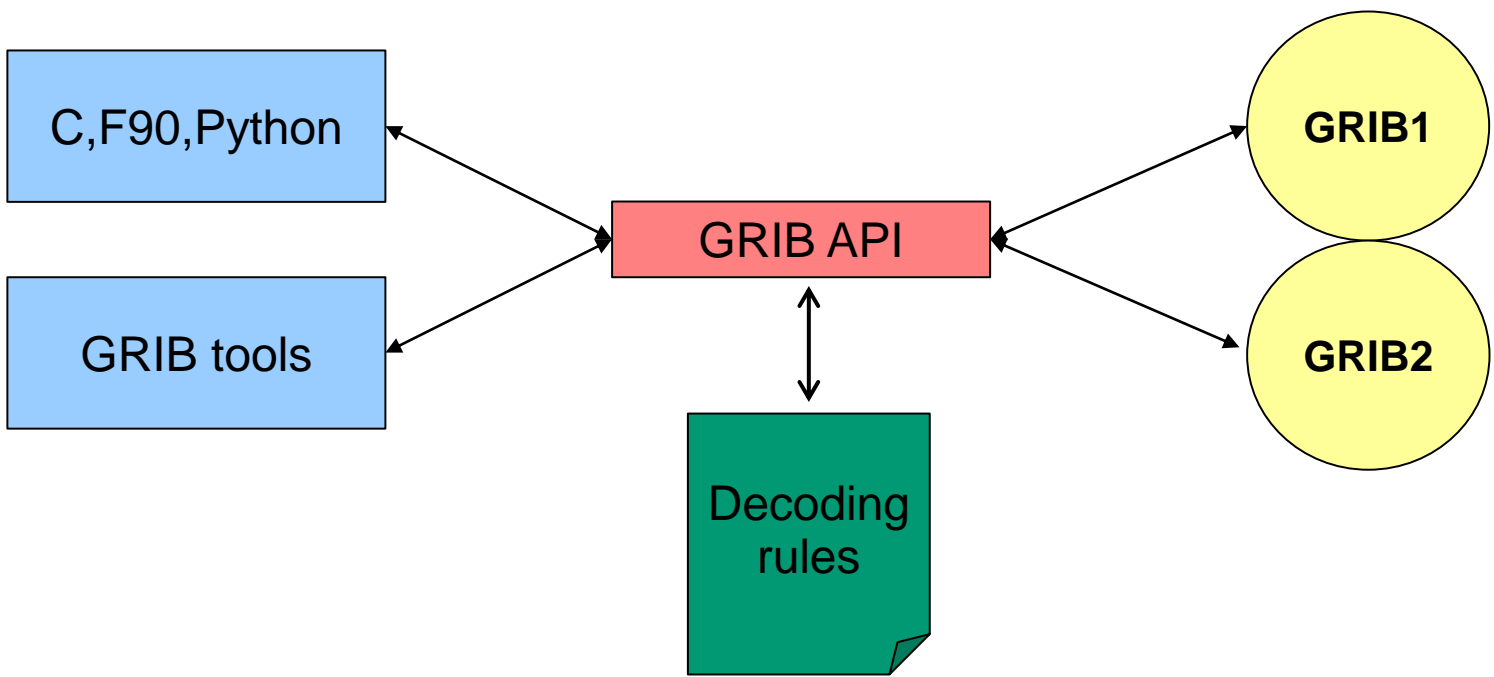
- ✗ scarsa documentazione

- ✗ approccio poco intuitivo



Interfacce e struttura

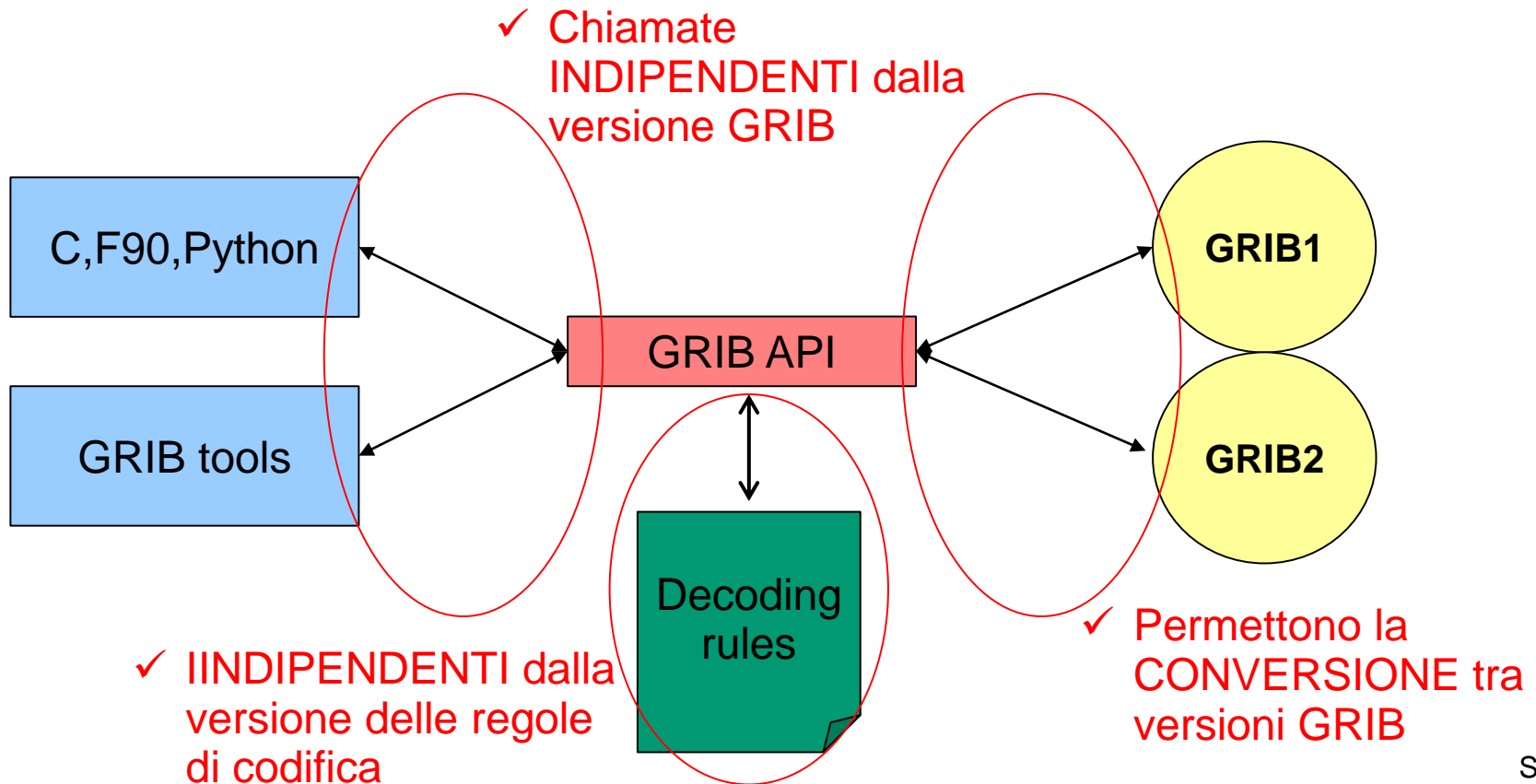
- Interfacce disponibili:
- C (nativo)
 - **Fortran 90**
 - Phyton
- } gestione tramite *chiamate a funzioni*
- + Performanti
+ Versatili
-
- GRIB tools: Applicativi pronti
- } gestione tramite *comandi di linea*
- + Intuitivi





Interfacce e caratteristiche

- Interfacce disponibili:
- C (nativo)
 - **Fortran 90**
 - Python
- } gestione tramite *chiamate a funzioni*
- + Performanti
+ Versatili
-
- GRIB tools: Applicativi pronti
- } gestione tramite *comandi di linea*
- + Intuitivi

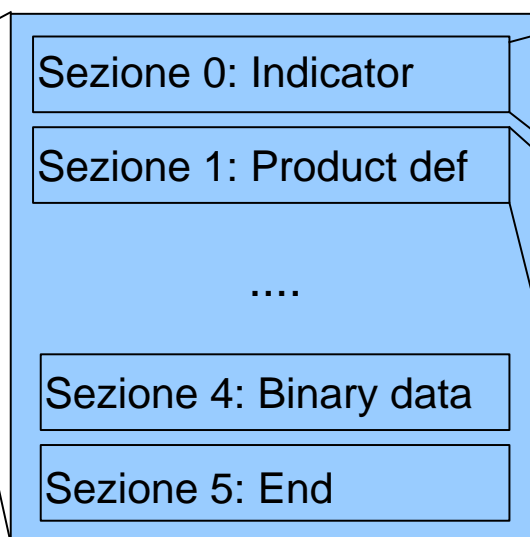




Accesso ai GRIB



file.GRIB



| Octet no. | IS Content |
|-----------|---|
| 1-4 | 'GRIB' (Coded CCITT-ITA No. 5) (ASCII); |
| 5-7 | Total length, in octets, of GRIB message(including Sections 0 & 5); |
| 8 | Edition number - currently 1 |

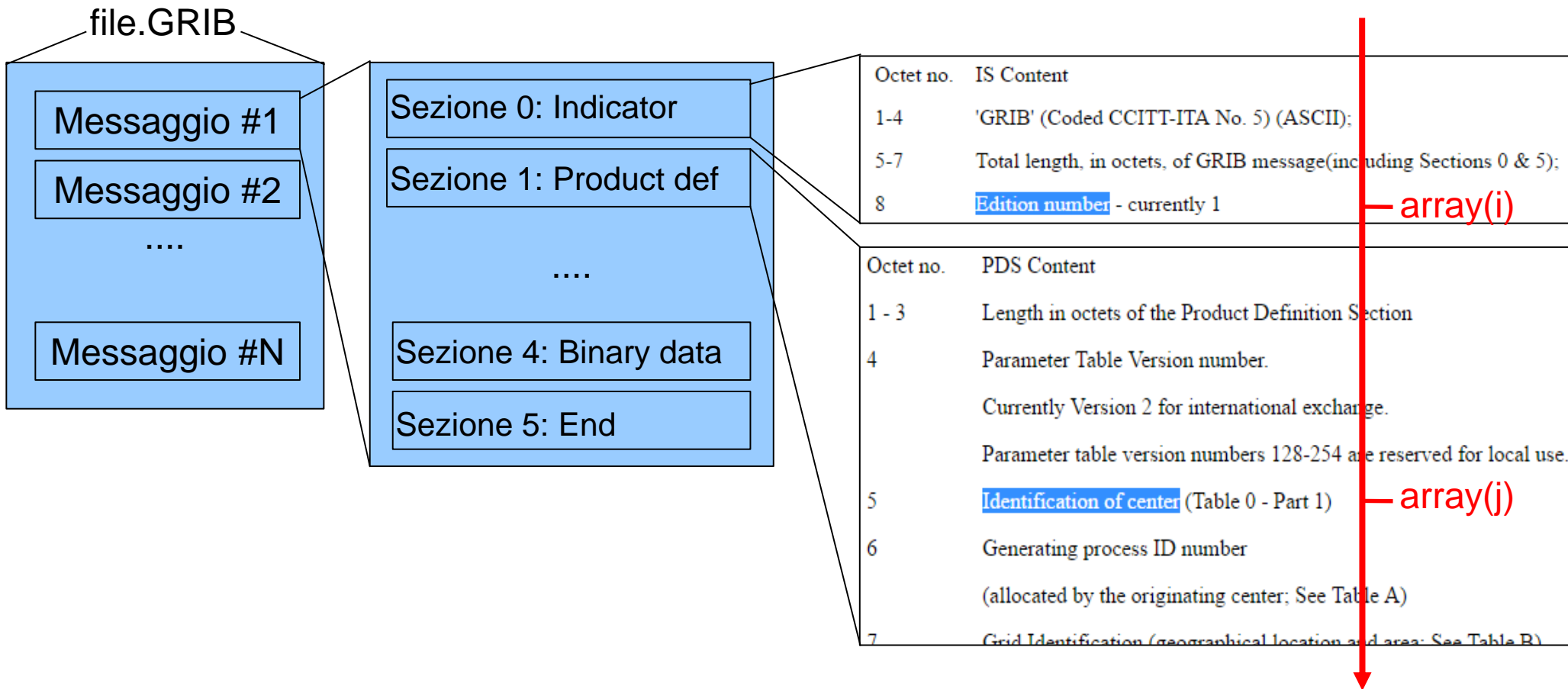
| Octet no. | PDS Content |
|-----------|---|
| 1 - 3 | Length in octets of the Product Definition Section |
| 4 | Parameter Table Version number. Currently Version 2 for international exchange. Parameter table version numbers 128-254 are reserved for local use. |
| 5 | Identification of center (Table 0 - Part 1) |
| 6 | Generating process ID number (allocated by the originating center; See Table A) |
| 7 | Grid Identification (geographical location and area; See Table B) |

grib_dump (grib TOOLS)

```

===== MESSAGE 1 ( length=80360 ) =====
1-4  identifier = GRIB
5-7  totalLength = 80360
8    editionNumber = 1
===== SECTION_1 ( length=52, padding=0 ) =====
1-3  sectionLength = 52
4    table2Version = 128
5    centre = 98 [European Centre for Medium-Range Weather Forecasts (grib1/0.table) ]
6    generatingProcessIdentifier = 145
7    gridDefinition = 255
8    section1Flags = 128 [10000000]
9    indicatorOfParameter = 172 [Land-sea mask ((0 - 1) (grib1/2.98.128.table) ]
10   indicatorOfTypeOfLevel = 1 [Surface (of the Earth, which includes sea surface) (grib1/local/ecmf/3.table , grib1/3.table) ]
11-12 level = 0
  
```

Accesso key-based - Accesso **array-based**



❖ GRIBEX, CDO, ...

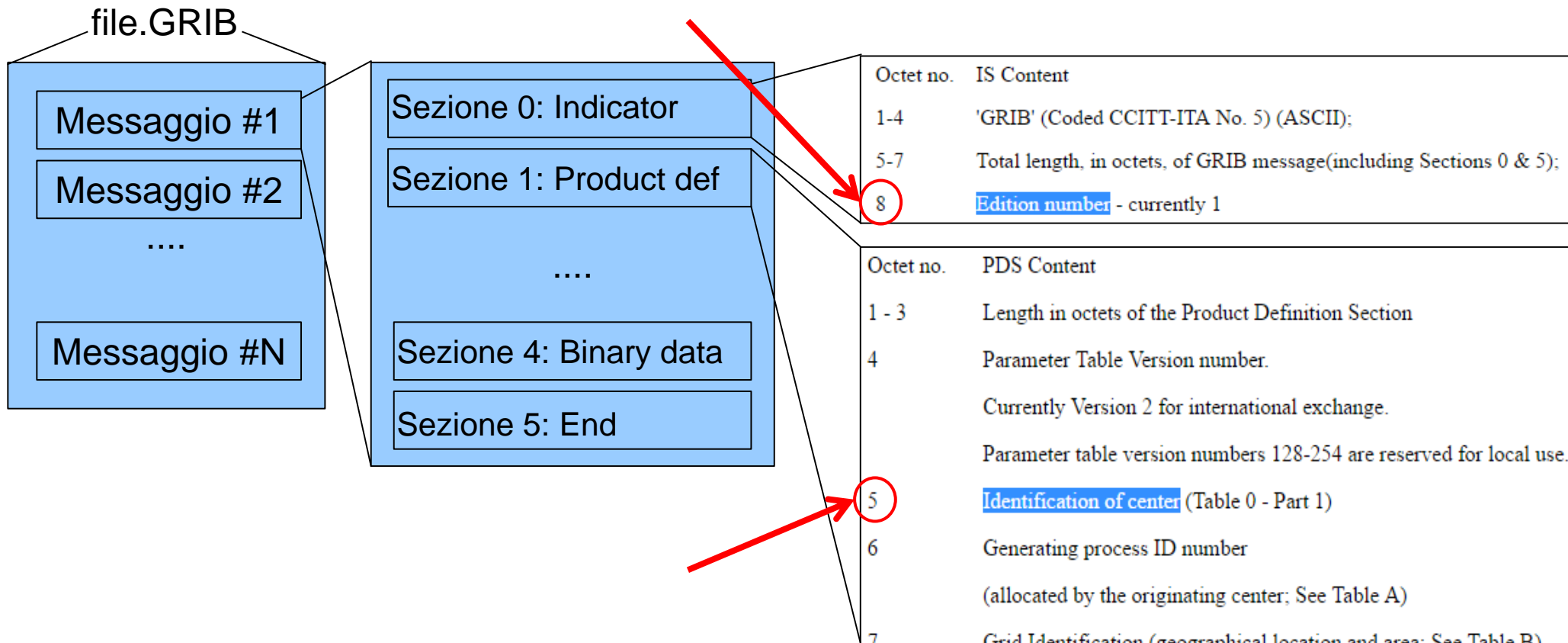
Ricerca per elemento in **array**
(GRIB caricato in memoria)

❖ GRIB API

Ricerca valore della **chiave**
(accesso solo agli ottetti corrispondenti)

+ Veloce

Accesso **key-based** - Accesso array-based



❖ GRIBEX, CDO, ...

Ricerca per elemento in array
(GRIB caricato in memoria)

❖ GRIB API

Ricerca per posizione della **chiave**
(accesso SOLO agli **ottetti corrispondenti**)

+ Veloce

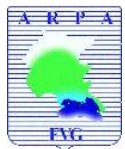


GRIB API keys - caratteristiche

- NOME** Chiavi individuate tramite nome (basato sulla documentazione WMO)
- FORMATO del valore** Formato nativo (real, integer o string)
- ✘ non è noto (solo l'interfaccia in C restituisce il tipo)
 - ✓ in alcuni casi si può convertire (*nome:i*, *nome:l*, *nome:d*, *nome:s*)
- TIPO** Codificate/Calcolate
- ACCESSIBILITA' del valore** Modificabili/READ-ONLY
- DISPONIBILITA'** Le chiavi in un messaggio variano a seconda di
- contenuto del messaggio (per es. forecastTime/dataTime)
 - versione GRIB
 - valore attribuito ad alcune di esse (per es. *edition*)

Elenco keys disponibili: [grib_ls](#), [grib_dump](#) (grib TOOLS)

```
edition  centre  typeOfLevel  level  dataDate  stepRange  dataType  shortName  packingType  gridType
1        ecmf     surface      0      20151126  0          an        lsm        grid_simple  regular_ll
1 of 1 grib messages in /lustre/arpa/goglioa/tmp/GRIB_api/prova_maschera.GRIB
```



GRIB API keys - esempi



| Categoria (namespace) | Keys GRIB1/GRIB2 | Es. di valori |
|-----------------------|---|---|
| Is | <ul style="list-style-type: none"> - editionNumber - dataType | 1,2 an,fc |
| parameter | <ul style="list-style-type: none"> - name - shortName - indicatorOfParameter/parameterNumber - indicatorOfTypeOfLevel/parameterCategory - units/parameterUnits | Temperature 2t,10u 172,167 sfc,100 K, m s-1 |
| vertical | <ul style="list-style-type: none"> - level | 0,500 |
| time | <ul style="list-style-type: none"> - dataDate - dataTime - forecastTime | 20151125 1200 90 |
| geography | <ul style="list-style-type: none"> - iScansNegatively - jScansPositively - latitudeOfFirstGridPointInDegrees - latitudeOfLastGridPointInDegrees - longitudeOfFirstGridPointInDegrees - latitudeOfLastGridPointInDegrees | 0, 1 0, 1 |
| data values | <ul style="list-style-type: none"> - numberOfPoints/numberOfDataPoints | 259920 |



GRIB API **index**: gestione delle keys



INDICE → Gestione messaggi tramite accesso ai soli ottetti relativi alle chiavi scelte

Index_keys='indicatorOfParameter, indicatorOfTypeOfLevel, level, shortName, units'

| indicatorOfParameter: | indicatorOfTypeOfLevel: | level: | shortName: | units: |
|-----------------------|-------------------------|--------|------------|-----------------------|
| 129 | pl | 0 | 10u | % |
| 130 | sfc | 7 | 10v | (0 - 1) |
| 131 | | 10 | 2d | K |
| 132 | | 28 | 2t | Pa |
| 133 | | 30 | ci | kg kg**-1 |
| 134 | | 50 | lsm | m of water equivalent |
| 139 | | 70 | msl | m s**-1 |
| 141 | | 100 | q | m**2 s**-2 |
| 151 | | 150 | r | m**3 m**-3 |
| 157 | | 200 | sd | |
| 164 | | 250 | skt | |
| 165 | | 300 | sp | |
| 166 | | 400 | sst | |
| 167 | | 500 | stl1 | |
| 168 | | 600 | stl2 | |
| 170 | | 700 | stl3 | |
| 172 | | 800 | stl4 | |
| 183 | | 850 | swvl1 | |
| 235 | | 925 | swvl2 | |
| 236 | | 1000 | swvl3 | |
| 31 | | | swvl4 | |
| 34 | | | t | |
| 39 | | | tcc | |
| 40 | | | u | |
| 41 | | | v | |
| 42 | | | z | |



INDICE della presentazione



1. Codice **GRIB**
 - Struttura e caratteristiche
 - Edizioni 1 e 2
2. Applicativi **GRIB API**
 - Definizione
 - Interfacce e Struttura
 - Caratteristiche
 - Logica di accesso ai GRIB
 - Chiavi
 - Caratteristiche
 - Esempi
 - Indici
3. Interfaccia **FORTAN 90**
 - **Regole generali**
 - **Struttura dei programmi**
 - **Funzioni disponibili**
 - **Gestione degli errori**
4. Esempi
 - Estrazione campo in un punto lat/lon
 - Estrazione su tutta la griglia
 - Creazione e conversione file GRIB
5. Documentazione



Interfaccia FORTRAN 90 / Regole generali



- All'inizio del programma è necessario chiamare il modulo grib api:
USE grib_api
 - Tutti gli oggetti sono gestiti tramite un **numero ID**:
 - ID file.GRIB
 - ID messaggi – corrisponde al numero del record
 - ID indice
 - ID iterator
 - Per poter leggere/scrivere un **file.GRIB** è necessario aprirlo/chiuderlo:
CALL grib_open_file/CALL grib_close_file
 - Per creare un **indice** contenente solo le chiavi scelte:
CALL grib_index_create/CALL grib_new_from_index
CALL grib_index_release
- oppure
- per accedere ad un intero **messaggio** questo deve essere caricato/scaricato:
CALL grib_new_from_file
CALL grib_release



Interfaccia FORTRAN 90 / Struttura generale dei programmi



```
PROGRAM nome_programma
```

```
! Modulo grib_api:
```

```
USE grib_api
```

```
IMPLICIT NONE
```

```
.....
```

```
! Apertura file.GRIB
```

```
CALL grib_open_file(id_filein,file.GRIB,open_mode)
```

'r' = read

'w' = write

```
! Apertura di un messaggio completo dal file.GRIB
```

```
CALL grib_new_from_file(id_filein, id_record)
```

```
! Oppure
```

```
! Creazione di un indice dal file.GRIB
```

```
CALL grib_index_create(id_index,file.GRIB,index_keys)
```

```
.....
```

```
! Chiusura messaggio e/o indice
```

```
CALL grib_index_release(id_index)
```

```
CALL grib_release(id_record)
```

Elenco chiavi dell'indice da creare ed eventuali conversioni di tipo, per es.
'indicatorOfParameter,indicatorOfTypeOfLevel:l,level'

```
! Chiusura file.GRIB
```

```
CALL grib_close_file(id_filein)
```

```
END PROGRAM nome_programma
```

Intent(in)
Intent(out)

Interfaccia FORTRAN 90 / Funzioni disponibili



- ~ 40 funzioni disponibili

- Categorie di funzioni disponibili:

- Gestione dei file.GRIB
- Gestione dei messaggi
- Gestione degli indici
- Gestione delle chiavi
- Gestione dei dati
- Gestione degli errori

- Formato argomenti restituiti:

nome_funzione(arg1,arg2,..argN(out),..) —————> argN(out) formato nativo

nome_funzione_formato(arg1,arg2,..argN(out),..) —————> argN(out) *formato*



Interfaccia FORTRAN 90 / Funzioni per la gestione di file.GRIB



Intent(in)
Intent(out)

- Apertura file in modalità di lettura o scrittura
`grib_open_file(id_filein,file.GRIB,open_mode)`
open_mode: 'r'=read (lettura-modifica),'w'=write (creazione)
- Chiusura file
`grib_close_file(id_filein)`
- Scrittura di un messaggio presente in memoria in un file.GRIB
`grib_write(id_mes,id_outfile)`
outfile.GRIB deve essere stato aperto in scrittura
- Lettura di n byte da file.GRIB
`grib_read_bytes(file.GRIB,buffer,nbyte)`
buffer: stringa
- Scrittura n byte in file.GRIB
`grib_write_bytes(file.GRIB,buffer,nbyte)`



Interfaccia FORTRAN 90 / Funzioni per la gestione di messaggi



Intent(in)
Intent(out)

- Conteggio messaggi presenti in un file.GRIB
`grib_count_in_file(id_filein,mes_num)`
- Apertura di un messaggio da file.GRIB
`grib_new_from_file(id_filein, id_record)`
- Chiusura di un messaggio
`grib_release(id_mes)`
- Lettura del numero di byte occupati dal messaggio
`grib_get_message_size(id_mes, byte_size)`
- Copia di un messaggio in memoria
`grib_copy_message(id_mes,array)`
array: character(len=1),dimension(1:byte_size)
- Creazione di un messaggio da un array caricato in memoria
`grib_new_from_message(id_mes, array)`
- Elenco del contenuto
`grib_dump(id_mes)`



Interfaccia FORTRAN 90 / Funzioni per la gestione di indici



Intent(in)
Intent(out)

- Creazione da file GRIB (scelta delle chiavi)
`grib_index_create(id_index,file.GRIB,index_keys)`
- Creazione da un altro indice (selezione valore delle chiavi)
`grib_new_from_index(id_index, id_new_index)`
- Chiusura indice creato
`grib_index_release(id_index)`
- Aggiunta di più file GRIB da gestire tramite un unico indice
`grib_index_add_file(id_index,file.GRIB)`
- Conteggio valori distinti assunti da ciascuna chiave nel file GRIB
`grib_index_get_size(id_index, key, size)`
- Selezione del contenuto in base al valore delle chiavi
`grib_index_select(id_index,key,key_value)`
l'indice NON viene modificato con questa operazione,
è necessario crearne uno nuovo con `grib_new_from_index`
- Salvataggio in file.idx
`grib_index_write(id_index,file.idx)`
- Lettura file.idx
`grib_index_read(id_index,file.idx)`

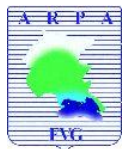


Interfaccia FORTRAN 90 / Funzioni per la gestione di chiavi



Intent(in)
Intent(out)

- Conteggio valori distinti assunti da ciascuna chiave in un indice
`grib_index_get_size(id_index, key, size)`
- Cattura valore/valori di una chiave da un messaggio o da un indice
`grib_get(id_mes, key, value)`
`grib_get(id_index, key, values)`
- Modifica valore di una chiave in un messaggio
`grib_set(id_mes, key, new_value)`
- Controllo dello stato di una chiave in un messaggio (se è missing o meno)
`grib_is_missing(id_mes, key, flag)`
flag: 0=not missing, 1=missing
- Settaggio stato missing per una chiave in un messaggio
`grib_set_missing(id_mes, key)`
- Creazione, cancellazione, interrogazione e scorrimento *iterator* (scorre i valori delle chiavi in un messaggio):
`grib_keys_iterator_new(id_mes, id_iterator, name_space)`
se name_sapace è lasciato vuoto scorre su tutti i tipi di chiavi
`grib_keys_iterator_delete(id_iterator)`
`grib_keys_iterator_get_name(id_iterator, key_name)`
`grib_keys_iterator_next(id_iterator)`
`grib_keys_iterator_rewind(id_iterator)`



Interfaccia FORTRAN 90 / Funzioni per la gestione di chiavi



Intent(in)
Intent(out)

- Conteggio valori distinti assunti da ciascuna chiave in un indice
`grib_index_get_size(id_index, key, size)`

- Cattura valore/valori di una chiave da un messaggio o da un indice

`grib_get(id_mes, key, value)`

`grib_get(id_index, key, values)`

grib TOOLS

grib_get -p key1, key2, ... File.GRIB

- Modifica valore di una chiave in un messaggio

`grib_set(id_mes, key, new_value)`

grib TOOLS

**grib_set -s key1=val1, key2=val2, ...
in.GRIB out.GRIB**

- Controllo dello stato di una chiave in un messaggio (se è missing o meno)

`grib_is_missing(id_mes, key, flag)`

flag: 0=not missing, 1=missing

- Settaggio stato missing per una chiave in un messaggio

`grib_set_missing(id_mes, key)`

- Creazione, cancellazione, interrogazione e scorrimento *iterator* (scorre i valori delle chiavi in un messaggio):

`grib_keys_iterator_new(id_mes, id_iterator, name_space)`

se name_sapace è lasciato vuoto scorre su tutti i tipi di chiavi

`grib_keys_iterator_delete(id_iterator)`

`grib_keys_iterator_get_name(id_iterator, key_name)`

`grib_keys_iterator_next(id_iterator)`

`grib_keys_iterator_rewind(id_iterator)`



Interfaccia FORTRAN 90 / Funzioni per la gestione di dati

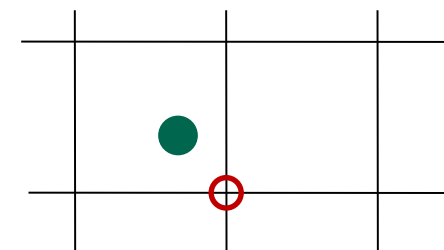


- Cattura coordinate e valore di un messaggio su tutta la griglia
`grib_get_data(id_mes,lats,lons,values)`

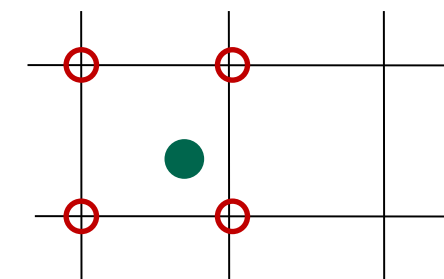
| Latitudine | Longitudine | Valore |
|------------|-------------|--------|
| ... | ... | ... |

Intent(in)
Intent(out)

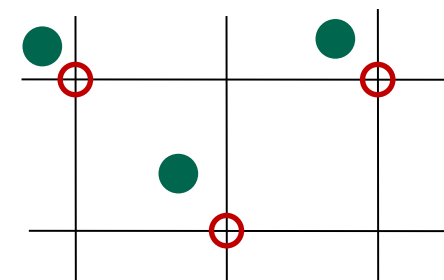
- Cattura il valore campo nel punto di griglia più vicino a una certa coordinata e della distanza di questo dal punto
`grib_find_nearest_single(id_index,ls_flag,inlat,inlon,outlat,outlon,value,distance,index)`



- Cattura il valore campo nei 4 punti di griglia più vicini a una certa coordinata e della distanza di questi dal punto
`grib_find_nearest_four_single(id_index,ls_flag,inlat,inlon,outlats,outlons,values,distances,indexs)`



- Cattura valori del campo del punto di griglia più vicino ad ogni coordinata di un array di coordinate e della distanza di questi
`grib_find_nearest_multiple(id_index,ls_flag,inlats,inlons,outlats,outlons,values,distances,indexs)`



ls_flag: true=considera solo punti di tipo land
false=considera punti land/sea

- ❑ Tutte le funzioni hanno un argomento opzionale per l'**exit status**

grib_...(arg1,arg2,...,ex) \longrightarrow ex \neq 0 : ERRORE nell'esecuzione

- Errore associato al codice ex:
`grib_get_error_string(ex,error_str)`
- Trappola: esce dall'esecuzione se ex corrisponde ad error_string
`grib_check(ex,soubroutine_name,error_string)`

| |
|-----------------------------------|
| <p>Intent(in) Intent(out)</p> |
|-----------------------------------|

- ❑ Se NON si utilizza **ex** l'esecuzione viene interrotta appena ex \neq 0
- ❑ Altro utilizzo di ex (per es. ciclo su tutti i record in un file GRIB):

```

grib_new_from_file(id_file,id_mes,ex)

DO WHILE (ex $\neq$ GRIB_END_OF_FILE)
    ....

    grib_new_from_file(id_file,id_mes,ex)
END DO

```



INDICE della presentazione



1. Codice GRIB

- Struttura e caratteristiche
- Edizioni 1 e 2

2. Applicativi GRIB API

- Definizione
- Interfacce e Struttura
- Caratteristiche
- Logica di accesso ai GRIB
- Chiavi
 - Caratteristiche
 - Esempi
- Indici

3. Interfaccia FORTRAN 90

- Regole generali
- Struttura dei programmi
- Funzioni disponibili
- Gestione degli errori

4. Esempi

- **Estrazione campo in un punto lat/lon**
- **Estrazione su tutta la griglia**
- **Creazione e conversione file GRIB**

5. Documentazione



Interfaccia FORTRAN 90 / Esempi:

1) Estrazione in un punto lat/lon



Intent(in)
Intent(out)

```
PROGRAM point_extraction
! Modulo grib_api:
USE grib_api
```

```
IMPLICIT NONE
INTEGER :: ex, id_filein, id_index, id_sel
REAL(KIND=8) :: inlat=45.000, inlon=13.000
REAL(KIND=8) :: outlat, outlon, index, value, distance
```

! Apertura file.GRIB

```
CALL grib_open_file(id_filein, './data/file.GRIB', 'r', ex)
```

! Creazione di un indice contenente le chiavi relative a nome, tipo e livello delle variabili

```
CALL grib_index_create(id_index, file.GRIB, 'indicatorOfParameter, indicatorOfTypeOfLevel:l, level', ex)
```

! Selezione del campo nell'indice in base a nome, tipo e livello della T2

```
CALL grib_index_select(id_index, 'indicatorOfParameter', 167, ex)
CALL grib_index_select(id_index, 'indicatorOfTypeOfLevel', 100, ex)
CALL grib_index_select(id_index, 'level', 0, ex)
```

! Creo un nuovo indice che punti solo ai valori selezionati

```
CALL grib_new_from_index(id_index, id_sel, ex)
```

! Lettura del valore del campo nel punto di griglia più vicino alle coordinate inlat/inlon e distanza da queste

```
CALL grib_find_nearest_single(id_sel, .false., inlat, inlon, outlat, outlon, value, distance, index, ex)
```

```
WRITE(UNIT=*, FMT='(A,F7.3,A,F7.3,A,F9.3,A,F9.3)') 'outlat=', outlat, 'outlon=', outlon, 'dist=', distance, 'val=', value
```

! Chiusura indici creati

```
CALL grib_index_release(id_index)
CALL grib_index_release(id_sel)
```

! Chiusura file.GRIB

```
CALL grib_close_file(id_filein)
```

```
END PROGRAM point_extraction
```



Interfaccia FORTRAN 90 / Esempi:

1) Estrazione in un punto lat/lon



Intent(in)
Intent(out)

```
PROGRAM point_extraction
! Modulo grib_api:
USE grib_api
```

```
IMPLICIT NONE
INTEGER :: ex, id_filein, id_index, id_sel
REAL(KIND=8) :: inlat=45.000, inlon=13.000
REAL(KIND=8) :: outlat, outlon, index, value, distance
```

! Apertura file.GRIB

```
CALL grib_open_file(id_filein, './data/file.GRIB', 'r', ex)
```

! Creazione di un indice contenente le chiavi relative a nome, tipo e livello delle variabili

```
CALL grib_index_create(id_index, file.GRIB, 'indicatorOfParameter,indicatorOfTypeOfLevel:l,level', ex)
```

! Selezione del campo nell'indice in base a nome, tipo e livello della T2

```
CALL grib_index_select(id_index, 'indicatorOfParameter', 167, ex)
CALL grib_index_select(id_index, 'indicatorOfTypeOfLevel', 100, ex)
CALL grib_index_select(id_index, 'level', 0, ex)
```

ATTENZIONE
al tipo attribuito alle key

! Creo un nuovo indice che punti solo ai valori selezionati

```
CALL grib_new_from_index(id_index, id_sel, ex)
```

! Lettura del valore del campo nel punto di griglia più vicino alle coordinate inlat/inlon e distanza da queste

```
CALL grib_find_nearest_single(id_sel, .false., inlat, inlon, outlat, outlon, value, distance, index, ex)
```

```
WRITE(UNIT=*, FMT='(A,F7.3,A,F7.3,A,F9.3,A,F9.3)') 'outlat=', outlat, 'outlon=', outlon, 'dist=', distance, 'val=', value
```

! Chiusura indici creati

```
CALL grib_index_release(id_index)
CALL grib_index_release(id_sel)
```

! Chiusura file.GRIB

```
CALL grib_close_file(id_filein)
```

```
END PROGRAM point_extraction
```



Interfaccia FORTRAN 90 / Esempi:

2) Estrazione su tutta la griglia



Intent(in)
Intent(out)

```
PROGRAM info
```

```
! Modulo grib_api:
USE grib_api
```

```
IMPLICIT NONE
```

```
INTEGER :: ex, id_filein, id_index, id_se, gridpoints_num, data, ora
REAL(KIND=8), DIMENSION(:), ALLOCATABLE :: lats, lons, values
```

```
...apertura file.GRIB...
```

```
...creazione indice var T2...
```

```
! Lettura di data e ora
```

```
CALL grib_get(id_sel, 'dataDate', data, ex)
```

```
CALL grib_get(id_sel, 'dataTime', ora, ex)
```

```
! Lettura numero dei punti di griglia
```

```
CALL grib_get(id_sel, 'numberOfPoints', gridpoints_num, ex)
```

```
ALLOCATE(lats(gridpoints_num))
```

```
ALLOCATE(lons(gridpoints_num))
```

```
ALLOCATE(values(gridpoints_num))
```

```
! Lettura lat, lon e valori su tutta la griglia
```

```
CALL grib_get_data(id_sel, lats, lons, values, ex)
```

```
! Ciclo su tutti i punti di griglia
```

```
DO i=1, gridpoints_num
```

```
  WRITE(UNIT=*, FMT='(A,A,A,A,A,F7.3,A,F7.3,A,F9.3)') data=' ', data, ora=' ', ora, lat=' ', lats(i), lon=' ', lons(i), val=' ', values(i)
```

```
END DO
```

```
...chiusura indici...
```

```
...chiusura file GRIB...
```

```
END PROGRAM info
```



Interfaccia FORTRAN 90 / Esempi:

2) Estrazione su tutta la griglia



Intent(in)
Intent(out)

```
PROGRAM info
```

```
! Modulo grib_api:
USE grib_api
```

```
IMPLICIT NONE
```

```
INTEGER :: ex, id_filein, id_index, id_se, gridpoints_num, data, ora
REAL(KIND=8), DIMENSION(:), ALLOCATABLE :: lats, lons, values
```

```
...apertura file.GRIB...
```

```
...creazione indice var T2...
```

```
! Lettura di data e ora
```

```
CALL grib_get(id_sel, 'dataDate', data, ex)
CALL grib_get(id_sel, 'dataTime', ora, ex)
```

```
! Lettura numero dei punti di griglia
```

```
CALL grib_get(id_sel, 'numberOfPoints', gridpoints_num, ex)
```

```
ALLOCATE(lats(gridpoints_num))
ALLOCATE(lons(gridpoints_num))
ALLOCATE(values(gridpoints_num))
```

ATTENZIONE
al verso di lettura della griglia

```
! Lettura lat, lon e valori su tutta la griglia
```

```
CALL grib_get_data(id_sel, lats, lons, values, ex)
```

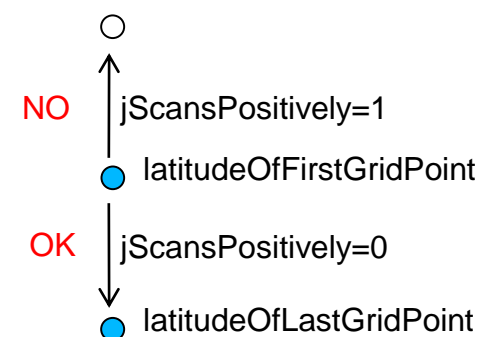
```
! Ciclo su tutti i punti di griglia
```

```
DO i=1, gridpoints_num
  WRITE(UNIT=*, FMT='(A,A,A,A,A,F7.3,A,F7.3,A,F9.3)') data=',data,'ora=',ora,'lat=',lats(i),'lon=',lons(i),'val=',values(i)
END DO
```

```
...chiusura indici...
```

```
...chiusura file GRIB...
```

```
END PROGRAM info
```





Interfaccia FORTRAN 90 / Esempi:

3) Creazione e conversione GRIB



Intent(in)
Intent(out)

PROGRAM edition_conversion

! Modulo grib_api:
USE grib_api

IMPLICIT NONE

INTEGER :: ex, id_filein, id_index, id_sel, id_fileout, id_newmes, mes_size
CHARACTER(LEN=1), DIMENSION(:), ALLOCATABLE :: message

! Apro il file GRIB in lettura

CALL grib_open_file(id_filein, './data/file.GRIB', 'r', ex)

! Apro il file.grib2 in scrittura

CALL grib_open_file(id_outfile, './data/newfile.grib2', 'w', ex)

....creazione indice var T2...

! Leggo le dimensioni del record

CALL grib_get_message_size(id_sel, mes_size, ex)

ALLOCATE(message(mes_size))

! Copio il record nell'array

CALL grib_copy_message(id_sel, message, ex)

! Creo il nuovo record

CALL grib_new_from_message(id_newmes, message, ex)

! Modifico il valore della key relativa all'edizione del GRIB (CONVERSIONE)

CALL grib_set(id_newmes, 'editionNumber', 2, ex)

! Scrivo il record nel nuovo file

CALL grib_write(id_newmes, id_outfile, ex)

...chiusura indici...

...chiusura file GRIB...

END PROGRAM edition_conversion



INDICE della presentazione



1. Codice GRIB

- Struttura e caratteristiche
- Edizioni 1 e 2

2. Applicativi GRIB API

- Definizione
- Interfacce e Struttura
- Caratteristiche
- Logica di accesso ai GRIB
- Chiavi
 - Caratteristiche
 - Esempi
- Indici

3. Interfaccia FORTRAN 90

- Regole generali
- Struttura dei programmi
- Funzioni disponibili
- Gestione degli errori

4. Esempi

- Estrazione campo in un punto lat/lon
- Estrazione su tutta la griglia
- Creazione e conversione file GRIB

5. Documentazione



Documentazione



- Guida GRIB API ECMWF (elenco funzioni, esempi):
<https://software.ecmwf.int/wiki/display/GRIB/Home>
- Slide introduttive GRIB API:
<https://software.ecmwf.int/wiki/display/GRIB/Training+material>
- Chiavi (elenco nomi, esempi):
<https://software.ecmwf.int/wiki/display/GRIB/GRIBEX+keys>
https://software.ecmwf.int/wiki/download/attachments/53515359/gribapi2_keys_2015.pdf?api=v2
- Guida GRIB WMO:
<http://www.wmo.int/pages/prog/www/WDM/Guides/Guide-binary-2.html>