A tutorial for the ECHAM6 boundary condition scheme

Sylvaine Ferrachat
IAC*ETH*

based on the implementation by
Sabine Schröder & Martin Schultz
FZJülich

HAMMOZ
AEROSOL & ATMOSPHERIC CHEMISTRY MODULES FOR ECHAM

Max-Planck-Institut für Meteorologie
JÜLICH FORSCHUNGSZENTRUM
IAC*ETH*
C2SM Centre for Climate Systems Modeling
UNIVERSITY OF OXFORD
FMI

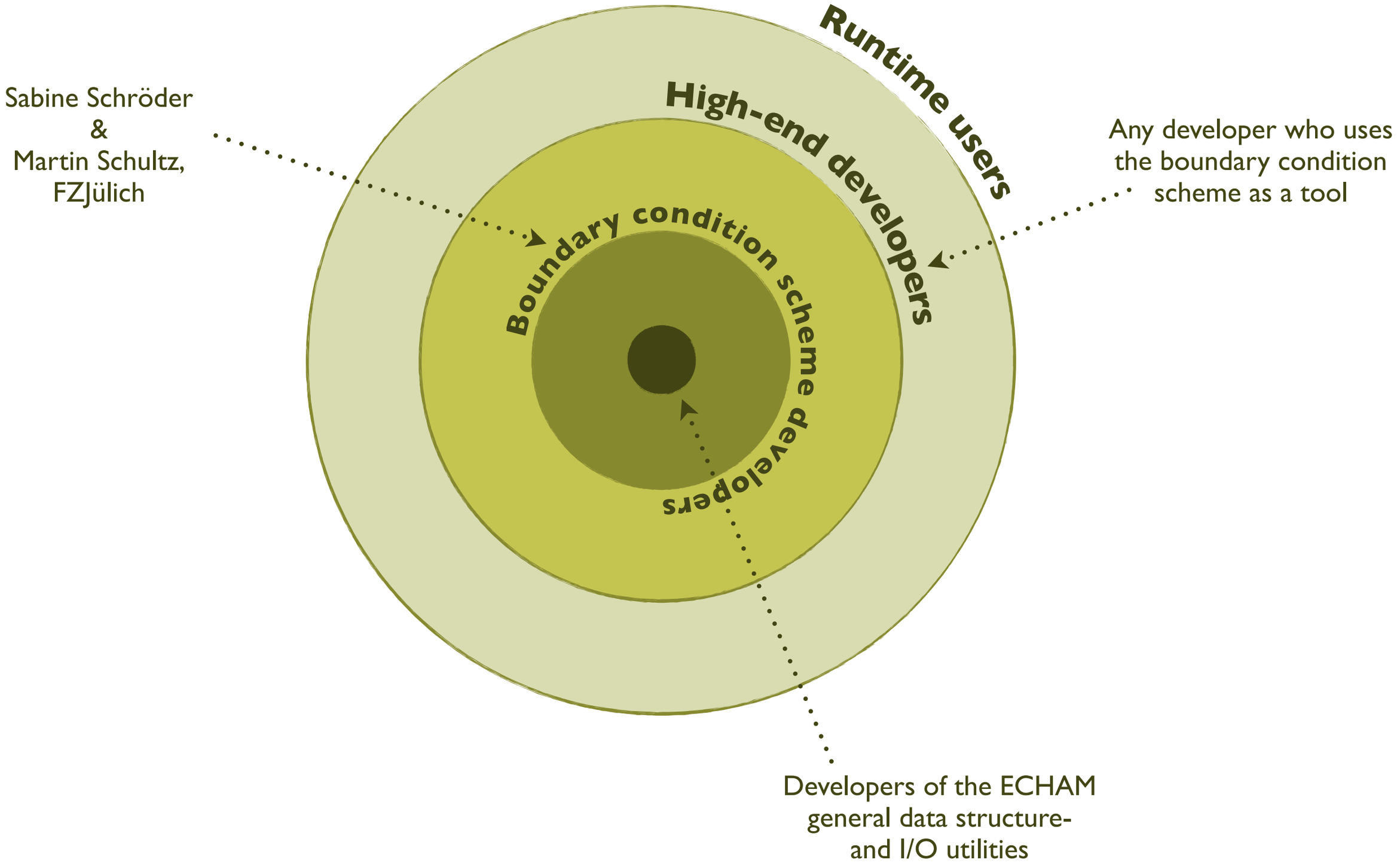
What is this?

- *Boundary condition*:
  - very general meaning [not specifically related to *differential equations solutions*]
- *Scheme*:
  - in this context, this refers to a technical piece of code [not a scheme based on *some given physical parametrization*]
- *Remark:*
  - this is not only for HAMMOZ, but is a part of ECHAM and may be used for any purpose

Rationale for having developed such a scheme

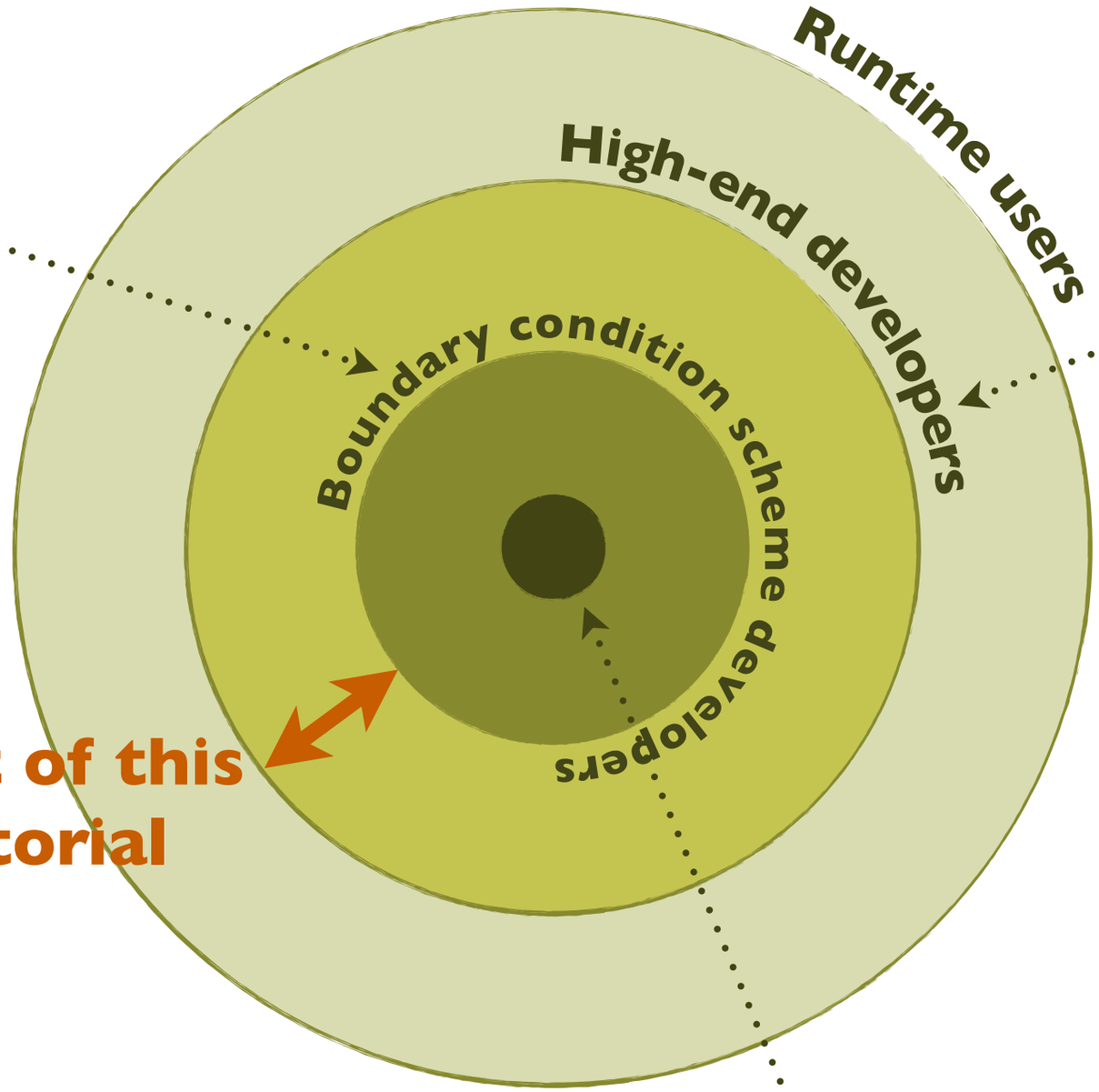
- Decoupling between acquisition and usage
  - several possible acquisition modes for one single usage in code
  - several usages (application modes) originating from one single dataset
- Consistent handling of common concepts
  - e.g. time interpolation, vertical interpolation, etc...
- Avoiding code duplication
  - no multiple netcdf reading subroutines
- Providing extra security
  - e.g. routine checks, etc...
- Offering an easy, safe, efficient and versatile toolbox

An onion-like organization



An onion-like organization

Sabine Schröder
&
Martin Schultz,
FZJülich



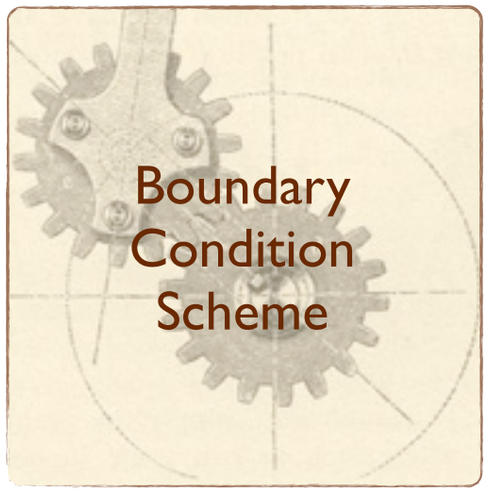
Any developer who uses the boundary condition scheme as a tool

Most of this tutorial

Developers of the ECHAM general data structure- and I/O utilities

HAMMOZ

The boundary condition scheme components



Click to expand

The boundary condition scheme components

I/O utilities
[netcdf reading]

data structure utilities
[streams]

mo_external_field_processor.f90

mo_boundary_condition.f90

mo_emi_matrix.f90

Utilities for file reading
[*ef_type=EF_FILE*]

Main brain

Boundary condition data structure
[bc_nml]

Click to expand

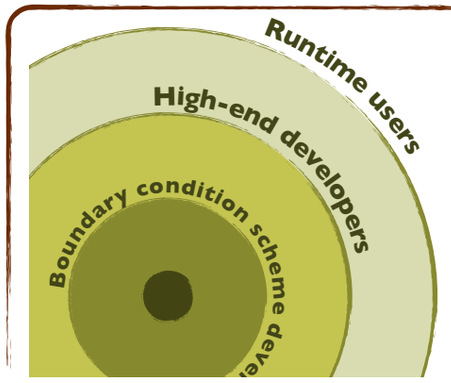
Toolbox for high-end developers

Click to expand

Runtime interface for HAMMOZ emissions

Click to expand

Legend



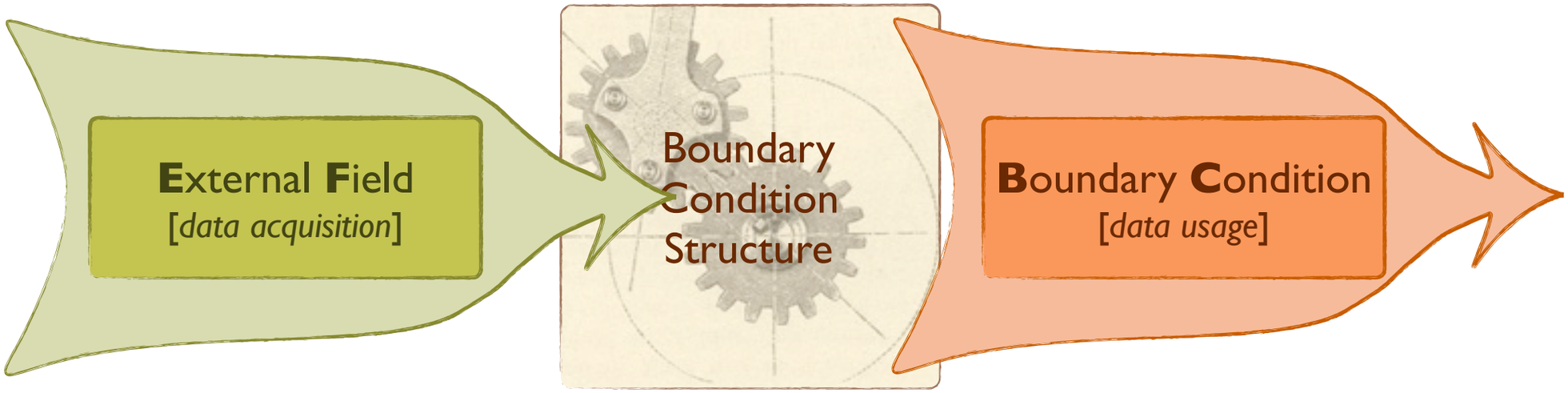
Uses

Provides

Complete documentation

HAMMOZ

The boundary condition structure *[bc_nml]*



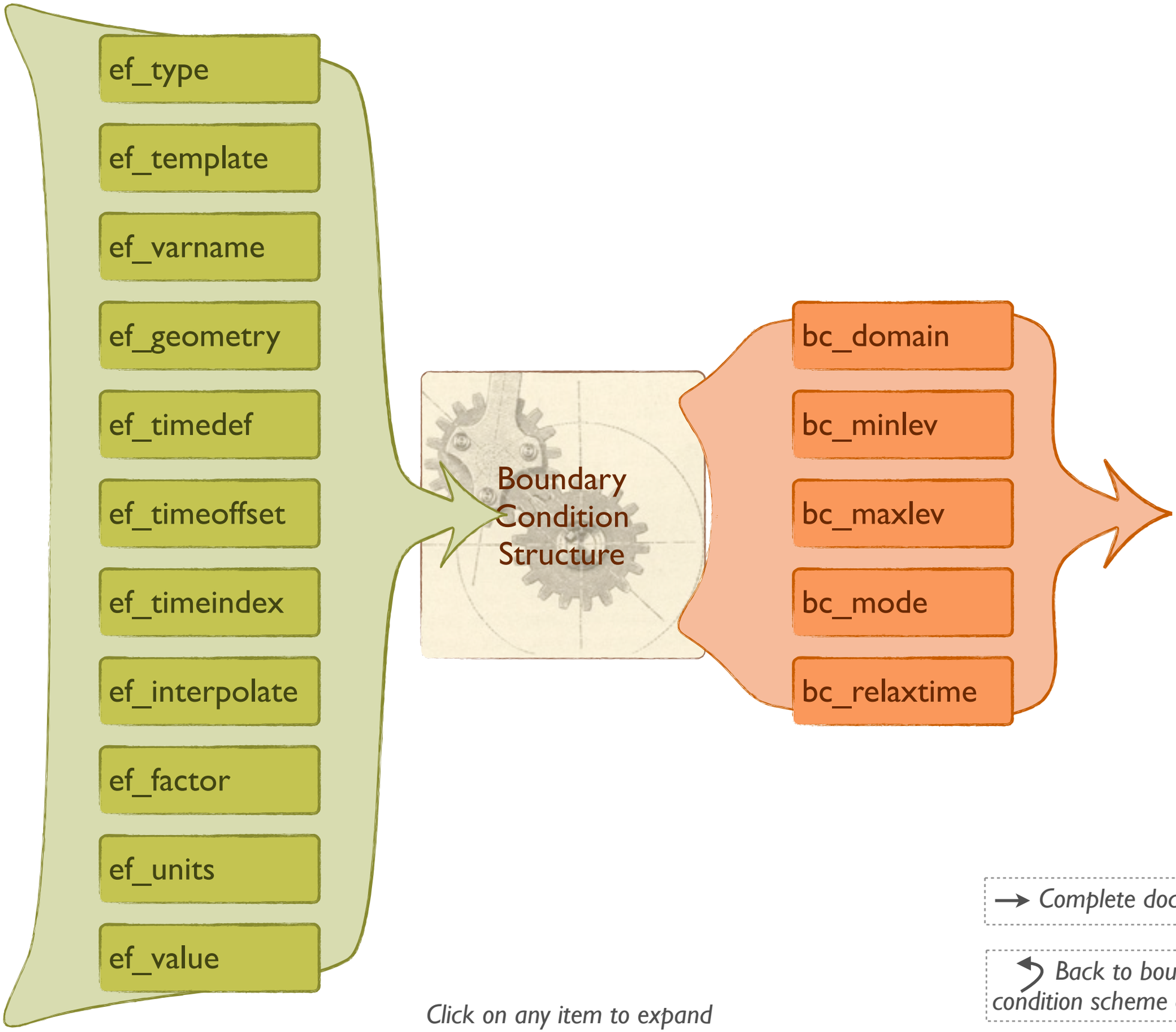
Click to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*



The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

Type of input data
[*origin*]

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

Boundary
Condition
Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

STRING

File name
[*may contain tokens, for e.g. year, resolution, etc...*]

- ef_type
- ef_template
- ef_varname
- ef_geometry
- ef_timedef
- ef_timeoffset
- ef_timeindex
- ef_interpolate
- ef_factor
- ef_units
- ef_value

Boundary Condition Structure

- bc_domain
- bc_minlev
- bc_maxlev
- bc_mode
- bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

STRING

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

Variable name
[*as set in file*]

Boundary
Condition
Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

EF_UNDEFINED

EF_3D

EF_LONLAT

EF_LATLEV

EF_LEV

EF_LAT

EF_SINGLE

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

Spatial dimensionality
[*as set in file*]

Boundary
Condition
Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

EF_TIMERESOLVED

EF_IGNOREYEAR

EF_CONSTANT

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

Type of time dependence

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

REAL

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

Time offset as compared to record [*in file time units*]

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

EF_TIMERESOLVED

EF_IGNOREYEAR

EF_CONSTANT

INTEGER

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

Index of time record in file

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

Complete documentation

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

EF_NOINTER

EF_LINEAR

~~EF_CUBIC~~

Time interpolation

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

REAL

Scaling factor
[*as compared to file values*]

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

STRING

Expected unit from file
[*for information and potential further checks*]

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

The boundary condition structure *[bc_nml]*

EF_INACTIVE

EF_VALUE

EF_FILE

EF_MODULE

ef_type

ef_template

ef_varname

ef_geometry

ef_timedef

ef_timeoffset

ef_timeindex

ef_interpolate

ef_factor

ef_units

ef_value

REAL

Globally uniform value

Boundary Condition Structure

bc_domain

bc_minlev

bc_maxlev

bc_mode

bc_relaxtime

Click on any item to expand

Complete documentation

Back to boundary condition scheme components

The boundary condition structure *[bc_nml]*

Where the boundary condition is applied [*if altitude or pressure, vertical interpolation performed*]

- ef_type
- ef_template
- ef_varname
- ef_geometry
- ef_timedef
- ef_timeoffset
- ef_timeindex
- ef_interpolate
- ef_factor
- ef_units
- ef_value

Boundary Condition Structure

- bc_domain
- bc_minlev
- bc_maxlev
- bc_mode
- bc_relaxtime

- BC_EVERYWHERE
- BC_BOTTOM
- BC_TOP
- BC_LEVEL
- BC_ALTITUDE
- BC_PRESSURE

Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*

Bounds for the range of applicable 'levels' [*height-, pressure-, or model levels*]

- ef_type
- ef_template
- ef_varname
- ef_geometry
- ef_timedef
- ef_timeoffset
- ef_timeindex
- ef_interpolate
- ef_factor
- ef_units
- ef_value

Boundary Condition Structure

- bc_domain
- bc_minlev
- bc_maxlev
- bc_mode
- bc_relaxtime

- BC_EVERYWHERE
- BC_BOTTOM
- BC_TOP
- BC_LEVEL
- BC_ALTITUDE
- BC_PRESSURE

INTEGER

INTEGER

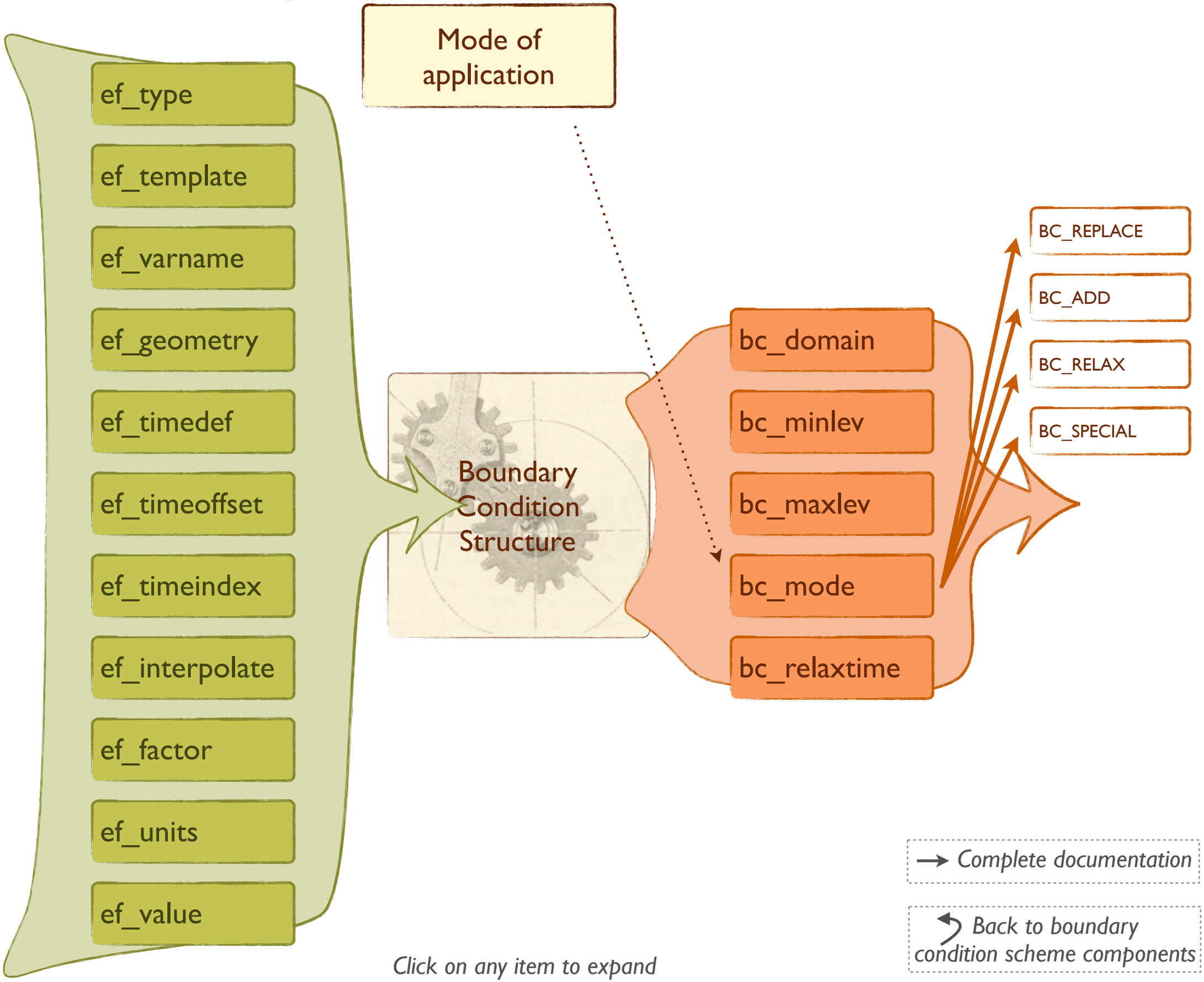
Click on any item to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

The boundary condition structure *[bc_nml]*



The boundary condition structure *[bc_nml]*

Relaxation time in seconds

$$\mathbf{f} = \mathbf{exp}(-\frac{\mathbf{\Delta t}}{\tau_{\mathbf{relax}}})$$

$$\mathbf{x(t + \Delta t)} = \mathbf{f} * \mathbf{x(t)} + \mathbf{(1 - f)} * \mathbf{bc(t)}$$

- ef_type
- ef_template
- ef_varname
- ef_geometry
- ef_timedef
- ef_timeoffset
- ef_timeindex
- ef_interpolate
- ef_factor
- ef_units
- ef_value

Boundary Condition Structure

- bc_domain
- bc_minlev
- bc_maxlev
- bc_mode
- bc_relaxtime

- BC_REPLACE
- BC_ADD
- BC_RELAX
- BC_SPECIAL

REAL

Click on any item to expand

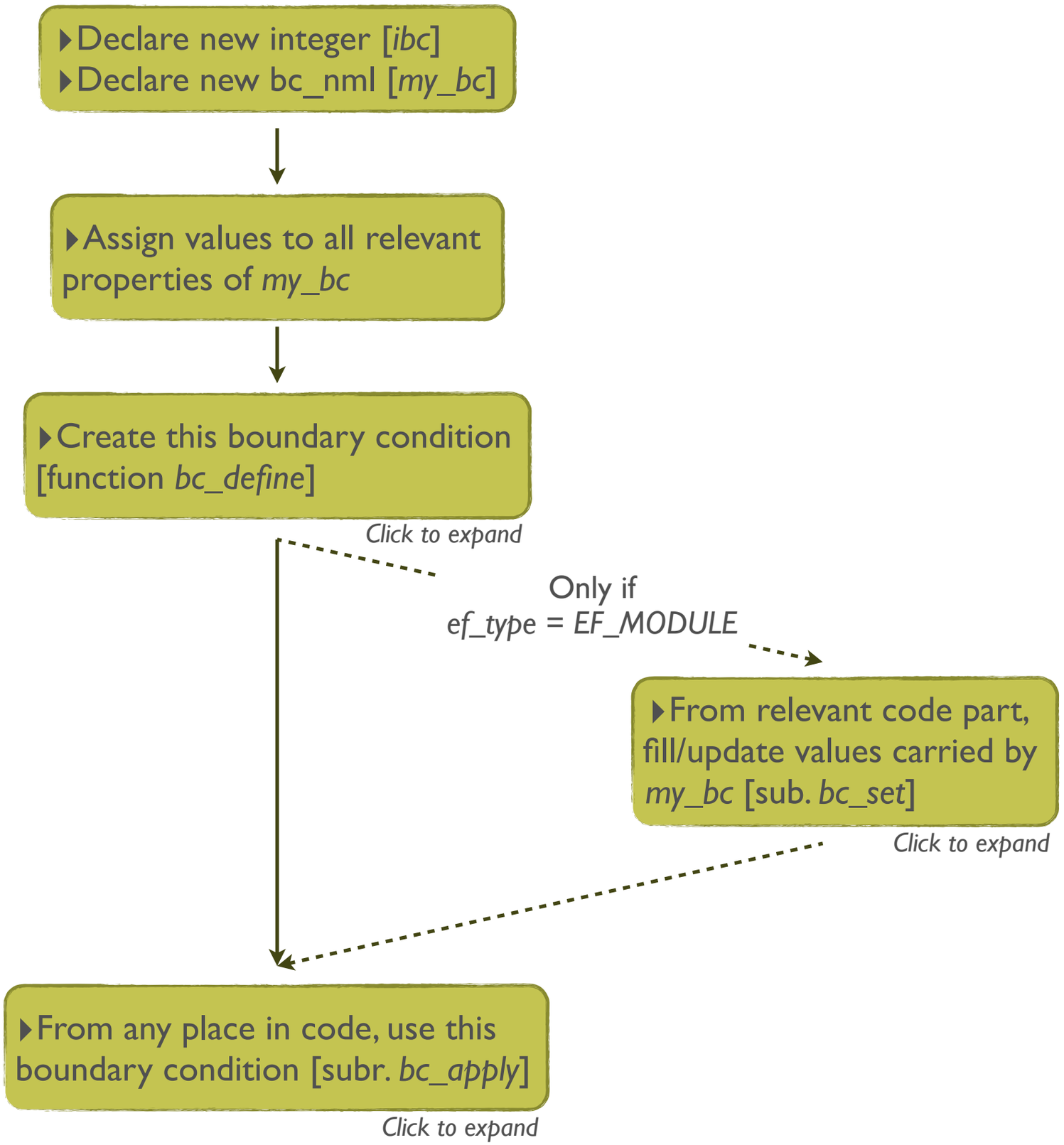
Complete documentation

Back to boundary condition scheme components

HAMMOZ

How to introduce a new boundary condition:

Basic recipe for high-end developers



How to introduce a new boundary condition:

Basic recipe for high-end developers

- Declare new integer [*ibc*]
- Declare new bc_nml [*my_bc*]

↓

- Assign values to all relevant properties of *my_bc*

↓

- Create this boundary condition [function *bc_define*]

Click to expand

```
ibc = bc_define('some name',my_bc,ndims)
```

Only if
ef_type = *EF_MODULE*

- From relevant code part, fill/update values carried by *my_bc* [sub. *bc_set*]

Click to expand

- From any place in code, use this boundary condition [subr. *bc_apply*]

Click to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

How to introduce a new boundary condition:

Basic recipe for high-end developers

- Declare new integer [*ibc*]
- Declare new bc_nml [*my_bc*]

↓

- Assign values to all relevant properties of *my_bc*

↓

- Create this boundary condition [function *bc_define*]

Click to expand

```
REAL(dp) :: z_my_bc_data(dim1,dim2)
...
CALL bc_apply(ibc,dim1,dim2,z_my_bc_data)

zxy(1:dim1,1:dim2) = z_my_bc_data(1:dim1,1:dim2) * ...
```

Only if *ef_type = EF_MODULE*

- From relevant code part, fill/update values carried by *my_bc* [sub. *bc_set*]

Click to expand

- From any place in code, use this boundary condition [subr. *bc_apply*]

Click to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

How to introduce a new boundary condition:

Basic recipe for high-end developers

- Declare new integer [*ibc*]
- Declare new bc_nml [*my_bc*]

↓

- Assign values to all relevant properties of *my_bc*

↓

- Create this boundary condition [function *bc_define*]

Click to expand

Only if *ef_type = EF_MODULE*

```
REAL(dp) :: zefgh(dim1,dim2)
...
zefgh(1:dim1,1:dim2) = ...

CALL bc_set(ibc,dim1,dim2,zefgh)
```

- From relevant code part, fill/update values carried by *my_bc* [sub. *bc_set*]

Click to expand

- From any place in code, use this boundary condition [subr. *bc_apply*]

Click to expand

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

How to introduce a new boundary condition:

Basic recipe for high-end developers

- Declare new integer [*ibc*]
- Declare new bc_nml [*my_bc*]

↓

- Assign values to all relevant properties of *my_bc*

↓

- Create this boundary condition [function *bc_define*]

Click to expand

Only if *ef_type = EF_MODULE*

- From relevant code part, fill/update values carried by *my_bc* [sub. *bc_set*]

Click to expand

- From any place in code, use this boundary condition [subr. *bc_apply*]

Click to expand

- No need to care for concrete I/O
- No need to care for proper update based on the currently used input file

→ *Complete documentation*

Back to boundary condition scheme components

HAMMOZ

An example of runtime-oriented usage: *the emi_spec matrix*

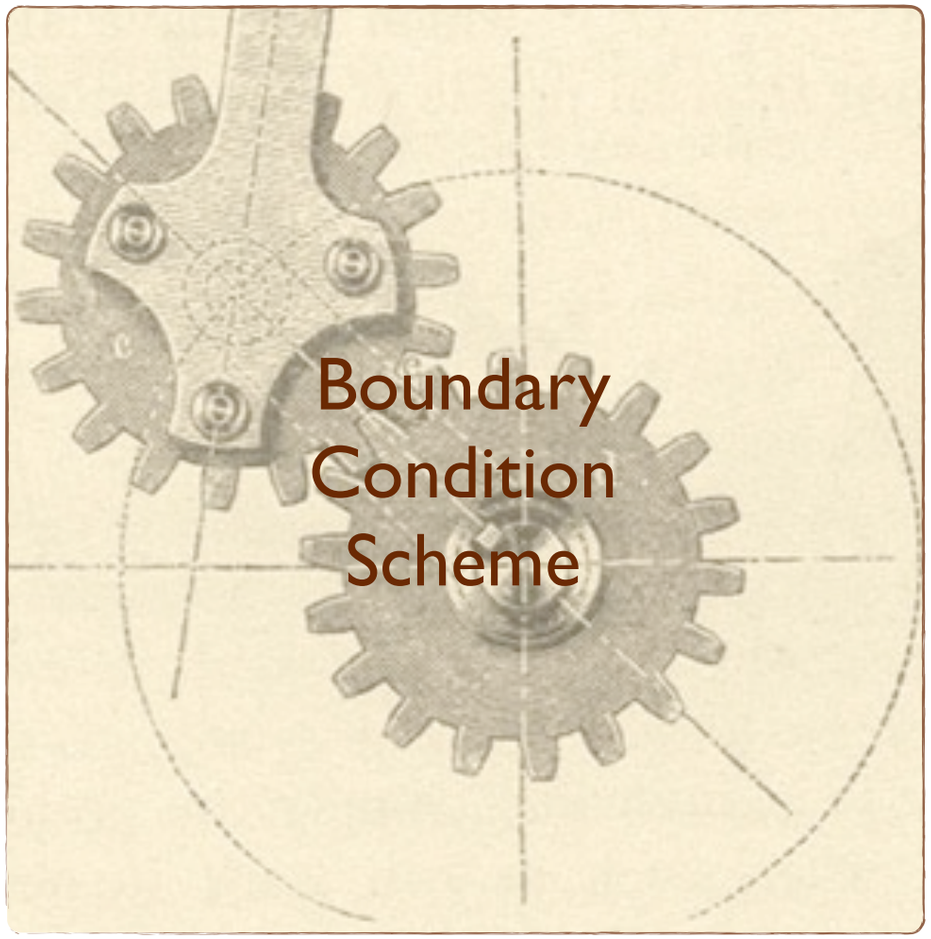
- See Grazia's talk
- Maybe now you understand better the meaning of these *EF_XXX* tags?

```
# Forest fire:
FFIRE=EF_FILE,    %T0/2000/emiss_aerocom_II_%C0_wildfire_2000_%T0.nc,      emiss_ffire,    EF_LONLAT, EF_IGNOREYEAR,
EF_TIMEOFFSET=-14.0, fire
# Grass fire:
GFIRE=EF_FILE,    %T0/2000/emiss_aerocom_II_%C0_wildfire_2000_%T0.nc,      emiss_gfire,    EF_LONLAT, EF_IGNOREYEAR,
EF_TIMEOFFSET=-14.0, fire
# Industry:
IND=EF_FILE,      %T0/2000/emiss_aerocom_II_%C0_anthropogenic_2000_%T0.nc, emiss_ind,      EF_LONLAT, EF_IGNOREYEAR, surface
# Ships:
SHIPS=EF_FILE,    %T0/2000/emiss_aerocom_II_%C0_ships_2000_%T0.nc,         emiss_shp,      EF_LONLAT, EF_IGNOREYEAR, level50m
# Solvent:
SLV=EF_FILE,      %T0/2000/emiss_aerocom_II_%C0_anthropogenic_2000_%T0.nc, emiss_slv,      EF_LONLAT, EF_IGNOREYEAR, surface
# Terrestrial:
TERR=EF_FILE,     %T0/emiss_aerocom_%C0_monthly_CLIM_%T0.nc,               DMS_terr,       EF_LONLAT, EF_IGNOREYEAR, surface
# Transport:
TRA=EF_FILE,      %T0/2000/emiss_aerocom_II_%C0_anthropogenic_2000_%T0.nc, emiss_tra,      EF_LONLAT, EF_IGNOREYEAR, surface
# Waste:
WST=EF_FILE,      %T0/2000/emiss_aerocom_II_%C0_anthropogenic_2000_%T0.nc, emiss_wst,      EF_LONLAT, EF_IGNOREYEAR, surface

# Dust:
DUST=EF_MODULE,    surface
# Oceanic:
OCEANI=EF_MODULE,  surface
# Sea salt:
SEASALT=EF_MODULE, surface
```

→ *Complete documentation*

Back to boundary condition scheme components



Click to expand

Complete documentation:

https://redmine.hammoz.ethz.ch/projects/hammoz/wiki/2_Technical_Documentation#Users-guides

[ECHAM6_Boundary_condition_scheme-v02a.pdf]