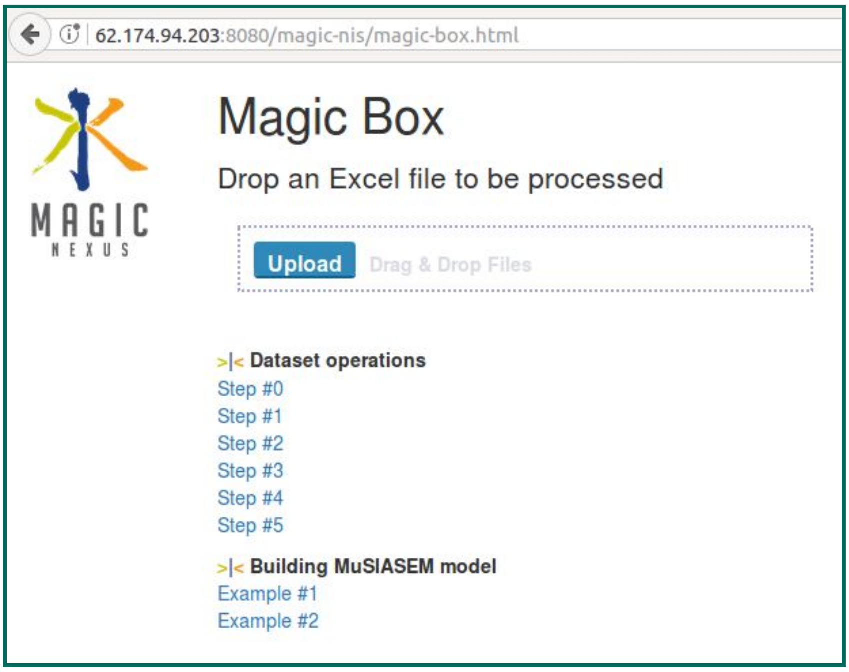
What is Magic Box?

- It is a prototype of parts of NIS, to gather and test ideas on building and analysing MuSIASEM case studies
- Excel used for input and output, to avoid learning a new tool
    - Need to learn how the commands are specified
    - Need to learn how to interpret results
    - and continue adding commands
- Used to identify commands examining already available case studies
    - No separate software analysis track
    - Needs some dialogue with case study authors
- Helps in meeting NIS goals. Mainly:
    - Standardization of MuSIASEM construction and analysis
    - Case studies as FAIR Open Research Data

How to use it?

Open this web site (temporary address) and drop a **specially formatted** Excel file (formatting is explained in the next section) inside a box shaped area (tested using Firefox; Chrome does not work properly). After a while, if everything is ok, you will be returned another Excel file based on the one just uploaded, containing the outcomes of the specified commands. Errors in input commands are signaled with red cells that will have a comment with the reason of the error.



62.174.94.203:8080/magic-nis/magic-box.html

Magic Box

Drop an Excel file to be processed

>|< Dataset operations

Step #0
Step #1
Step #2
Step #3
Step #4
Step #5

>|< Building MuSIASEM model

Example #1
Example #2

The process can be repeated successively using the obtained file as input.

Excel formatting

- The input Excel file (can be LibreOffice, save as ".xlsx" format) is examined for special worksheet names.

- Those commands not recognized are ignored.
- The commands are executed in sequence, from left to right. To use some data -Eurostat dataset, MuSIASEM processors, mapping- it must be available before, to the left of the command where it is used.

Preventive Warning: Excel content out of the standard “xlsx” format, which Microsoft itself sometimes breaks, may be filtered out (removed). An inventory of elements in these sense needs to be identified. The suggested workaround is to have linked spreadsheets, and use a base spreadsheet for NIS interaction and another for analysis using Excel proprietary features.

The current commands are:

“Metadata”

The first worksheet named “Metadata” will allow specifying metadata about the case study. In the future this information will serve to properly track submissions, to name case studies and to generate case-study-level metadata.

This worksheet is not mandatory. If not specified, the commands of the submission will be executed but the submission itself will not be registered under a case study (useful for experimentation).
See “Example #1” (in the web page).

“Dataset_Eurostat_Enumerate”

Obtain the list of all Eurostat datasets: code and English description. **No parameters**.
See “Step #0”.

“Metadata_Eurostat_<dataset name>”

Obtain the dimensions (in a statistical cube meaning) and corresponding code lists from an Eurostat dataset, which will allow exploration, and later the proper elaboration of a request of a slice of the dataset using “Dataset_Eurostat_<dataset name>. The parameter **dataset_name** (must be one of the Eurostat datasets) is specified in the worksheet name. The worksheet itself must be empty.
See “Step #1”, “Step #2”.

“NameMapping_<free name>”

Map categories of an Eurostat dataset code list to MuSIASEM categories. The format is a table, with a header specifying the source Eurostat code list “<dataset_name>”.”<code_list_name>”. The second is the name of the MuSIASEM taxonomy. Below, specific codes from the source code list can be assigned to categories from the MuSIASEM taxonomy. The mapping can be later used to filter datasets (see “Dataset_Eurostat_<dataset_name>”) and to obtain pivot tables (see “PivotTable_<free_short_description>”). Because a code from the source can be repeated, amounts may be doubly accounted (to different MuSIASEM categories), so the specification must be entered carefully. Once it has been checked, the mapping can be easily reused.
See “Step #5”.

“Dataset_Eurostat_<dataset_name>”

Obtain a slice of data from an Eurostat dataset. It is prepared to work in **two steps**:

- First, by only creating a tab with the appropriate name, leaving the worksheet empty, metadata on the dataset is obtained, arranged to prepare the request of the second step. The first line of it will contain the dimension names, and below each column the code lists can be found (in the column to the right, also the description are obtained, just for information). The first line will also contain the parameters for an optional PivotTable execution.
- After the previous, the information can be used to tailor a request, by deleting the unwanted codes (if all codes of a dimension are deleted, the dimension is not considered by the filter, i.e., all its codes are included in the result). PivotTable parameters work the same as with a separate “PivotTable_” tab.

For datasets organized as time series (most of the datasets), two special names:

- **“startPeriod”**. Temporal start of the slice, in units of the FREQ dimension (MuSIASEM normally uses yearly data, FREQ is “A”)
- **“endPeriod”**. Temporal end of the slice, in units of the FREQ dimension.

See “Step #1”, “Step #2”, “Step #3”, “Step #4”, “Step #5”.

“PivotTable_<free short description>”

Elaborate a pivot table (also called dynamic table) using one of the previously elaborated “Dataset_Eurostat…”, “Processors…” or “List…” (see below). What these three processes have in common is that they yield a table of tuples made of one or more dimension realizations and a value, a structure which allows summarization rearrangements in tabular form, commonly known as pivot tables or dynamic tables. The parameters for the pivot table must start in the header (first row) of the worksheet. They are:

- **“Sheet”**. Name of the worksheet containing the base data.
- **“Rows”**. Header names (one or more) whose categories to put in rows. At least one dimension must be specified.
- **“Columns”**. Header names (one or more) whose categories to put in columns. At least one dimension must be specified.
- **“AggFunc”**. One or more of the following:
  - Sum
  - Mean
  - Std
  - Max
  - Min
- **“ShowTotals”**. Add totals to the resulting matrix

It is assumed the existence of a column “**value**”, which is the measure around which the aggregations are performed.

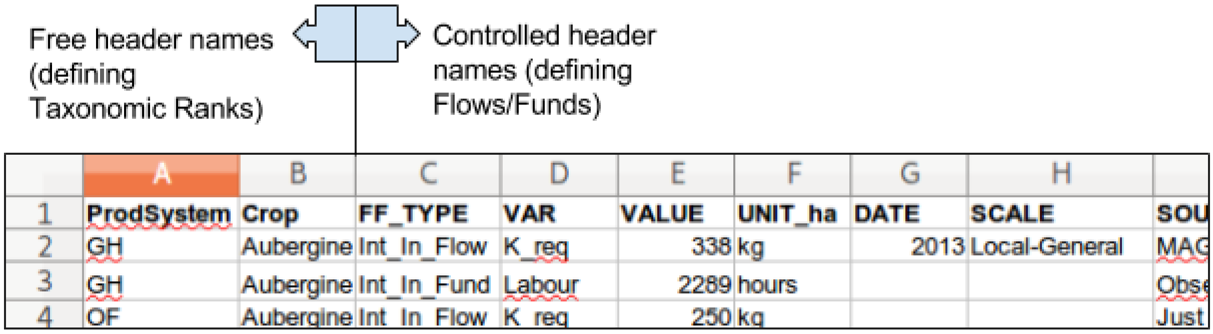
The command “Dataset_Eurostat_<dataset_name>” allows combining the gathering and filtering stage with the pivot table elaboration in a single worksheet. Just perform the first step of

"Dataset_Eurostat_<dataset_name>" and after that specify the pivot table parameters (they will appear to the right of the worksheet), or delete/remove them if not interested. The resulting dataset will be automatically named (internally) "PivotTable_<dataset_name>" for later referencing.
See "Step #4", "Step #5".

"Processors_<name>"

A worksheet specifying a set of intensive or extensive processors. It is possible to write all values or, if a value is equal to the previous one in the column, it can be omitted. Columns will be as follows:

- The header (first row) of the initial columns (starting from the left and up to the first formally named column) will define the names of the taxonomic ranks of the defined processors. This allows the flexibility of having one (minimum) or more columns naming a processor type. The first level of this taxonomy is the <name> provided in the worksheet name. Below the header, each name defined will be incorporated to the taxonomic rank.


Free header names (defining Taxonomic Ranks)
Controlled header names (defining Flows/Funds)

	A	B	C	D	E	F	G	H	
1	ProdSystem	Crop	FF_TYPE	VAR	VALUE	UNIT_ha	DATE	SCALE	SOU
2	GH	Aubergine	Int_In_Flow	K_req	338	kg	2013	Local-General	MAG
3	GH	Aubergine	Int_In_Fund	Labour	2289	hours			Obse
4	OF	Aubergine	Int_In_Flow	K_req	250	kg			Just

- **"FF_TYPE"**. The content of this column serves to indicate the type of factor, names come from Almería case study and are self explanatory. Allowed types are:
  - Int_In_Flow
  - Int_In_Fund
  - Int_Out_Flow
  - Ext_In_Flow
  - Ext_Out_Flow
- **"VAR"**. It will contain the name of the flow or fund.
- **"VALUE"**. Magnitude of the flow or fund.
- **"UNIT"** (for extensive processors) or **"UNIT_<unit_name>"** (for intensive processors). If intensive processors are being specified, the specified unit will be automatically made relative to the unit name in the header. For instance, if the header is "UNIT_ha", "kg" will be converted to "kg/ha". For adimensional amounts (like a percentage), a dash "-" has to be used. Adimensional factors are left out of the upscaling calculations.
- **"DATE"**, **"SOURCE"**, **"COMMENT"**. These columns are recognized (stored). Currently they are not used in the processing.

See "Example #1", "Example #2".

It is possible to map a PivotTable to a "Processors_<name>" by adding a single line following the previous structure, with special syntax. A hash tag "#" will signal the start of these references. The first occurrence of "#" needs to refer to an existing Pivot Table (for instance "PivotTable_1" or "PivotTable_<dataset_name>") followed by a dot "." and some element of the PivotTable. The elements are column or row names and "Value".

See “Step #5”.

“Upscale_<contained processor>_<container processor>”

A worksheet specifying how to embed <contained processors> into <container processor> (both must have been declared previously, using “Processors_”). The contents will be a matrix of values that must sum to one either in columns or in rows (not both). Taxonomic combinations not matching existing processors are ignored. If the container processor is extensive, it automatically converts to extensive the contained processors, in two steps: first applying the upscaling, next applying the fund to descendant processors, top-down in both cases.
See “Example #1”, “Example #2”.

“List_<container processor>_<some contained processor>”

A worksheet that can show the result of previous “Upscale_” worksheets. <contained processor> can be any of the processors contained recursively by the top level container processor.
See “Example #1”, “Example #2”.