# How the FOODEX.19 configuration works

**Configuration table:**

|  |  |
| --- | --- |
| ROOT\_GROUP\_CODE | varchar |
| FORBIDDEN\_PROCS | varchar |

The FOODEX.19 configuration is a table which defines sets of processes which cannot be used for raw commodities, since applying these processes would create a derivative which already exists in the catalogue (the derivative should be used instead as base term).

The rules are different depending on the food group of the raw commodity; therefore it is necessary to identify the food groups in order to assign different rules.

This is what the first column of the configuration does: it defines the code of the term which is the root of a food group. In particular, every descendant of the root will be subjected to the same rules, unless the descendant is itself another food group defined in the configuration. In this case, the sub-food group and its descendants will be subjected to their own rules (no inheritance is applied for the rules).

The second column instead defines for the related food group the forbidden processes. In particular, a row for every forbidden process must be added. For example, imagine the case in which we want to forbid the processes «Smoking (A07JV)» and «Canning / jarring (A0BYP)» for the food group identified by the code A0EZR (Fish meat and products thereof). The configuration will look like the following:

**ROOT\_GROUP\_CODE FORBIDDEN\_PROCS**

A0EZR A07JV

A0EZR A0BYP

However, if there is need of defining a list of forbidden processes only if they are used together (for example, if we want to forbid Smoking and Canning / jarring only if they are both present in the facets), it is possible to define them as a list $-separated in a single entry of the table:

**ROOT\_GROUP\_CODE FORBIDDEN\_PROCS**

A0EZR A07JV$A0BYP

As a general rule, the column FORBIDDEN\_PROCS defines a forbidden **combination** of processes. This gives flexibility to the configuration.

# Examples

## Forbid single processes for a food group

Consider the raw primary commodity (RPC) “Atlantic salmon” (A028P) in the Reporting hierarchy. In this case, the process “Smoking” (A07JV) cannot be applied to it, since the “Smoked salmon” term already exists in the considered hierarchy, and it should be used instead of the RPC as a starting point for creating the FoodEx2 code. The FOODEX2.19 rule allows forbidding the “Smoking” process by setting a proper configuration, which in this case would be the following:

|  |  |
| --- | --- |
| ROOT\_GROUP\_CODE | FORBIDDEN\_PROCS |
| A028P | A07JV |

However, this rule is actually valid for every fish in the Reporting hierarchy, therefore it would be necessary to specify for each fish the **same** forbidden processes, which could be a time consuming and error prone activity. For this reason, it is possible to define a set of rules for an entire food group, by using a root term, as explained in the previous Section. Since every RPC fish in the reporting hierarchy is child of the generic term “Fish meat and products thereof” (A0EZR), it is possible to define the rules on this term to affect every fish with a single configuration. In particular, the following configuration forbids the use of the “Smoking” process for every fish descendent of “Fish meat and products thereof”:

|  |  |
| --- | --- |
| ROOT\_GROUP\_CODE | FORBIDDEN\_PROCS |
| A0EZR | A07JV |

If more than one process should be forbidden, then it is only required to repeat the same logic of a single process by adding new rows in the configuration. For example, if also the “Canning / jarring” (A0BYP) process must be forbidden for fishes (apart from “Smoking”), then two rows should be set in the configuration:

|  |  |
| --- | --- |
| ROOT\_GROUP\_CODE | FORBIDDEN\_PROCS |
| A0EZR (Fish meat and products thereof) | A07JV (Smoking) |
| A0EZR (Fish meat and products thereof) | A0BYP (Canning/Jarring) |

## Forbid combinations of processes for a food group

In the previous Section it has been shown how to forbid the use of single processes for a specific food group. In this Section, it is explained with a practical example how to forbid the use of processes **together** (still allowing their use individually). For example, consider the “Globe tomato” (A00HR) RPC. In this case it is allowed to use the “Pulping / mashing” (A07LN) process, since there is no derivative which can be used instead. The same logic applies for the “Concentration / evaporation” (A07KF) process. What cannot be done is actually using the “Pulping / mashing” and “Concentration / evaporation” processes **together**, because a derivative named “Tomato paste” (A00ZF), which has both processes in the implicit facets, already exists in the hierarchy, and it should be used in this case as a starting point to describe the FoodEx2 code. For this reason, it is needed to forbid only the combination of processes, but not their individual use. This can be achieved by specifying a list of processes in the same FORBIDDEN\_PROCS cell ($ separated):

|  |  |
| --- | --- |
| ROOT\_GROUP\_CODE | FORBIDDEN\_PROCS |
| A00HR | A07LN$A07KF |

This configuration will raise an error only if the A07LN and A07KF processes are **both** added to the A00HR base term or to a descendent of it, but it will not block the use of the single process itself.