EUROMOD TECHNICAL NOTE SERIES

EMTN 1.1

EUROMOD Modelling Conventions

EUROMOD

January 2018

EUROMOD Technical Note Series

This series includes documents and papers about EUROMOD or specific uses of it that are suitable for sharing with model users or others interested in technical background and for citing in papers using EUROMOD. It complements the EUROMOD Working Paper Series which mainly contains substantive papers making use of EUROMOD for research or policy analysis, prior to final publication elsewhere.

If you would like to submit a Technical Note to the series please send to euromod@essex.ac.uk.

The Technical Note series includes documents that are regularly updated and this is reflected in the numbering and, where relevant, revision information in the following table:

Title	**How revised?**	**EMTN No.**	**Date of revision**
EUROMOD Modelling Conventions	First version for Technical Note series	1.0	February 2017
	Updated for start of EUROMOD update Year 9	1.1	January 2018

Citation

Please cite this document as EUROMOD, 2018, EUROMOD Modelling Conventions, EUROMOD Technical Note EMTN 1.1, Colchester: Institute for Social and Economic Research, University of Essex.

Editorial Note

The EUROMOD Modelling Conventions are documented in order to provide protocols and guidance to the large team of people involved in the update of EUROMOD. They are also useful information for EUROMOD users who want to know the detailed cross-country methods and assumptions adopted in the construction of each EUROMOD update. They are usually updated once a year. Comments are welcome and should be sent to euromod@essex.ac.uk.

EUROMOD Modelling Conventions

Last updated: January 31, 2018

This document is a collection of EUROMOD modelling conventions.[1] All guidelines are meant as compulsory except recommendations which are denoted with [REC].

Contents

1 General ... 1
2 Input datasets ... 2
3 Variable naming convention ... 4
4 Uprating (non-simulated) monetary variables ... 6
5 System and database configuration ... 9
6 Scope of policies ... 11
7 Order and structure of policies ... 11
8 Policy switches ... 13
9 Standard assumptions for simulating specific policies ... 15
  9.1 Minimum wage ... 15
  9.2 Unemployment (insurance) benefit ... 15
10 Tax units ... 17
11 Income lists ... 18
  11.1 Definitions of standard income lists ... 18
12 Output ... 21
13 Validation ... 21

1 General

- The term "**base-year simulation**" refers to the case, in which the year of policy rules matches input data income reference period (e.g. 2009 policy rules and SILC 2010 with 2009 income). The purpose of base-year simulation is to provide information as accurate as possible about the actual income distribution. Thus, aggregate figures of simulated components should match as closely as possible the official statistics. However, this should be done without any calibrations of the sort that would distort the effect of simulated changes. The aim is to create the best starting point for simulating changes, not to reproduce base year statistics as an end in itself.
- The term "**target-year simulation**" refers to the case, in which the year of policy rules does not match input data income reference period (e.g. 2010-2013 policy rules and SILC 2010). The purpose of target-year simulation is to provide accurate policy simulations based on certain assumptions about growth in (market) incomes rather than

---

[1] This is a list of *agreed* rules and as such partly forward-looking with some rules yet to be implemented in EUROMOD.

deriving the actual income distribution. Thus, aggregate figures of simulated and non-simulated components will not necessarily match official statistics.[2]

- The term "**baseline simulations**" refers to the best-possible combination (`best match') between policy rules year and input data (e.g. 2009-2013 policy rules and SILC 2010, in case SILC 2010 is the most recent data available). See also Section 5.
- The EU official country acronyms are to be used (e.g. in tax unit and input/output file names etc). The list can be found at http://publications.europa.eu/code/en/en-370100.htm.

2 Input datasets

- Input dataset is kept in (tabulated) **text format** and the file named ***CC_year_x#*** where *cc* is country acronym, *year* refers to the data collection year, *x* is a letter identifying database source (a, b, …) and # is the version number. For example, *ES_2005_b1.txt*.
- All variables in input data must be **documented** in a Data Requirement Document (DRD): most importantly, how they have been derived from the original source and what they contain. See also information requirements in the DRD template.
- Every dataset must include all compulsory variables (see the list in the DRD template). Where no required information is available, a variable with zero values needs to be created.
- Reference units:
    - Data is provided at the **individual level** (with people grouped into households).
    - All income variables must be provided with **gross values**, i.e. before deduction of employee and self-employed social insurance contributions and any taxes but excluding employer social insurance contributions. Where these are not available they must be imputed.[3]
    - Income and expenditure data must be expressed in **monthly terms**, i.e. divided by 12 if originally recorded in annual terms regardless of the actual number of months of receipt.[4]
    - Monetary variables need to be presented in the **national currency** (if other than the euro). Convert monetary variables using the same exchange rate as was used originally to convert the national currency into euros (e.g., PX010 in SILC). See Section 5 for rules concerning exchange rate for policy parameters and output.

---

[2] Nevertheless, by comparing against official figures one needs to check that the results from the simulation and uprating (see Section 4) move in a sensible direction and so, be able to explain any large variation in EUROMOD estimates. Furthermore, the Country Report discussion of the macrovalidation results should explain, where possible, the main reasons for diverging external and EUROMOD estimates (e.g. falling unemployment not captured in the target year EUROMOD estimates). See Section 13 for more on validation.

[3] If both gross and net values are recorded in the data, such values must be checked to see if these are reliable (e.g. gross > net; ratio between gross and net according to income source, personal characteristics and tax-benefit rules).

[4] Where the latter is known this information needs to be retained in the input dataset for improving the simulation accuracy of monthly-based policies (e.g., social insurance contributions and unemployment benefit).

- Sample adjustments:
    - Observations with zero (or negative) household weight need to be dropped from the sample. If observations with positive weights have been dropped (e.g. they had missing information on income) the weights must be recalibrated.
    - Children who are born after the income reference period must be dropped from the sample. (For example, those born in 2006 in EU-SILC 2006). Instead, an indicator variable is used to denote the number of such children in each household.
    - [REC] If population changes between the income reference period and policy year are such that compromise the validation of the model, re-weighting, in principle, could be considered as an option.[5]
- EUROMOD operates only with personal level variables:
    - Any monetary variable at the household level in the original dataset (typically, capital income, family allowances, social exclusion benefits, inter-household transfers, taxes and social contributions) must be assigned to one person in the household. Components of market income can be also divided between several persons where it makes sense. When the choice of the person(s) is not obvious, the value should be assigned to the person whose age is closest to 45 (either below or above). If there is more than one such person then the one who appears first in the household (by idperson).
    - Capital and property income must be shared equally between the oldest household member and his/her partner (the underlying assumption being that in the case of three generation households it would be probably the oldest couple who would retain this kind of income).
    - Any monetary variable related to housing at the household level in the original dataset (housing allowances, imputed rent, housing cost) must be assigned to the person(s) responsible for the accommodation. If there are two of such person then the amount is shared equally between them.
    - Any non-monetary variable at the household level in the original dataset should be assigned to all the persons in the household.
- Children with no parents ("loose children") are not to be assigned to other adults in the household if there are any.[6] However, one should check how common these cases are.
- Other imputations:
    - Missing values are not allowed. For non-applicable (N/A), -1 or 0 can be used.
    - Negative income values (e.g. self-employment) are to be included without any adjustments (to be consistent with EU-SILC income definition). However, negative income values should be recoded as zero within EUROMOD, in the policy neg_cc.
    - Age must be recorded as at the end of the income reference period.

---

[5] Notice though that this requires ideally information on how original weights were constructed.

[6] In order not to lose this information. Note that this might have implications for tax and benefit unit definitions.

  - Where there is no information about financial capital in the original data source (e.g. SILC), this needs to be imputed using average interest rates for each type of investment income recorded in the data.[7] By default, the interest rate should be calculated as the arithmetic average of the base interest rate and the Harmonised Consumer Price Index (see here, indicator prc_hicp_aind) if both are positive and if only one of them is positive then that should be used.

- Level of detail:
  - All cash incomes available in the original dataset need to be retained, including information on taxes and benefits to be simulated in the model (where this is reliable).[8] Capital gains and other lump-sum incomes (e.g. lottery winnings, severance pay) must be clearly separated from other incomes (e.g. dividends, interests etc) if possible.[9]
  - Income data should be as detailed as possible. Only the same type of income can be aggregated (e.g. earnings from the main and secondary job but not two unemployment benefits where one is means-tested and another not), on the condition that more detailed information is not needed for the model and unlikely to be used. Incomes can be aggregated up to class 2 (see Section 3), e.g. YEM.
  - The same monetary information cannot be entered in more than one variable of the same level of detail. If all different components (i.e. commonly rooted variables) are known then their sum must be equal to the root (with the exception of variables reporting duration).
  - Where both detailed and aggregated variables are included (e.g. YEM and YEM*) the exact relationship must be documented.
  - [REC] Housing expenditure should be as detailed as possible, at least with rent expenses separated from others (e.g. electricity, water etc).
- Original ID variables (renamed to origid*) are always retained so that the EUROMOD input dataset can be linked to the original data source (e.g., UDB or national SILC).

3 Variable naming convention

- Variables must be named following a naming convention that consists of a list of acronyms that joined together in a predetermined order build the variable's name. For a list of variables and acronyms currently used in EUROMOD see the model (Administration Tools > Variables).
- There are two classes of acronyms ordered hierarchically:
  - **Class 1**: one character that identifies the type of variable (**a**sset, **l**abour market, **d**emographic, **s**ystem, **y** for (cash) market income, e**x**penditure, **b**enefit, **p**ension,

---

[7] This is needed to facilitate policy swap exercises for tax-benefit systems which rely on these information.

[8] For validation purposes.

[9] In order to be able to exclude it from the standard concept of disposable income. However, for validation (and possibly other) purposes, it is useful to include them in the EUROMOD dataset rather than excluding altogether.

taxes and contributions, in-**k**ind income); the **id**-variables are exception as they have 2-letter class 1 acronym.

  - **Class 2**: two-character acronyms specific for each variable type (assets, demographic, etc). Each Class 2 acronym has a unique meaning within each variable type (Class 1). But it could be that the same acronym means different things across variable types (e.g., AG stands for age among demographic variables and agriculture in assets, taxes, market income and benefits). The Class 2 acronyms are listed in ordered groups.
  - For example, employment income is named yem: y for market income + em for employment.

- All variable names must always begin with a class 1 acronym followed by at least one class 2 acronym.
- The order of acronyms in the variable name must follow the order of the groups these acronyms are included. This prevents the same variable having different names because the acronyms are used in different orders. For example, a housing benefit complement for pensioners must be named bhopecm instead of bhocmpe (b - benefit, ho - housing, pe - pensioner, cm - complement).
- [REC] Avoid using acronyms from the same class 2 group. If more than one acronym is used from the same group then these should be ordered alphabetically.
- Intermediate Class 2 groups can be omitted, i.e. acronyms from a previous group are not compulsory.
- [REC] Avoid using more than five class 2 acronyms together (this would result in a name with more than 11 characters). Variable names include typically one to three class 2 acronyms.
- Each acronym should add a relevant or useful information. In case of monetary variables each additional acronym should represent an additional level of detail or specificity and it is assumed to be a component (part) of commonly rooted and less detailed variables. For example, bunctcm is a component of bunct which is a component of bun. (Note that main components can be distinguished from complements using the acronym 00. For example, poa00 is the main ("basic") old-age pension and poacm its complement.)
- Before creating a new variable, it must be checked whether that name (or something similar) is already defined in EUROMOD. It is strongly recommended to use the existing variables whenever possible. For example, if there is already a variable that measures the time worked in months, avoid creating a new one that measures that in weeks or years.
- When new categorical variables are created, the full list of categories, types or status must be documented.
- Each (input or output) variable used for a given country must have a country-specific description defined in the model (Administration Tools > Variables).

4 Uprating (non-simulated) monetary variables

- Where income reference period and policy year do not match, monetary variables need to be updated to the policy year. This is done by using relevant uprating factors in the model.
- Table 1 provides the guidelines for deriving uprating factors for market incomes as well as pensions, non-simulated benefits, and other sources of income, following the purpose of target-year simulations (see Section 1). These guidelines set out the **minimum standard** that country models need to fulfil and it is strongly encouraged to use for the derivation of uprating factors more accurate and comprehensive information, if available, and document the departures from the guidelines in the Country Report.
- Uprating of earnings:
  - In case national statistics don't provide (or provide with a delay) information suitable for uprating of earnings the following Eurostat data can be used: Gross wages and salaries, National accounts (time series can be found here, see Eurostat tables nama_10_gdp and nama_10_pe). This includes information on aggregate income from employment based on national accounts and number of employees. The data could be used to derive growth in average earnings if no better information is available.
  - For the most recent year (in case national forecasts are not available) the DG ECFIN's economic forecasts can be used (see description here and dataset here). The forecasts on nominal compensation per employee are available.
- Uprating of pensions and other benefits:
  - Detailed description of the policy changes regarding pensions needs to be included in the Country Report. Although pensions are largely not simulated in EUROMOD, they are a very important income source and we ought to know if there have been changes to the system and the extent to which they are captured in our simulations/uprating, if at all.
  - Uprating factors for public pensions and other (non-simulated) benefits should reflect **statutory and discretionary indexation rules (if such exists) as well as changes in policies** and exclude changes in the composition of recipients which could affect the dynamics of average amounts. Uprating factors should follow the rules which were applied in practice and should be comprehensive, capturing both regular annual indexation (according to the statutory rules or common practice) as well as additional ad hoc policy changes (cuts, revisions/recalculations, ad hoc increases etc). As such, calculations could be complex and involve differentiated % rates or even lump-sum amounts.
  - In addition to the main uprating factor for public pensions (i.e. the default option in the model), provide information on the **growth in average public pensions**.

Growth in average amounts would capture, in addition to the policy changes, changes in the composition of recipients.[10]

- If for the same pension/benefit two uprating factors are available calculated using information on 1) the indexation rule and 2) the growth in the average amount, the *default* uprating factor should be based on 1). However, a switch policy called UAA_cc (Uprating by Average Adjustment) should also enable the use of the uprating factor based on 2) instead of 1) (that may be useful for certain research purposes such as nowcasting poverty figures) – see also Section 8.

Table 1: Guidelines for deriving uprating factors (minimum standard)

Income component	**Uprating factors (by priority)**	**Comments**
	Market incomes	
Income from employment	1) **Growth in average earnings** (if possible in full-time equivalents) Projections (if data for above not available yet): 2) **Forecasts** (national source or Eurostat)	If available, please provide disaggregated information on the growth in average earnings (e.g. by sector, income deciles etc.).
Income from self-employment	Same as for income from employment	It is advised not to use information on aggregated self-employment income from national accounts because of likely large compositional changes and income fluctuations. However, if information on average self-employment income is available it can be used for the uprating factor and could be preferred over the minimum standard guidelines.
Private transfers	Same as for income from employment	
Private pensions	1) **Annual indexation**[11] 2) **Growth in average private pensions** if 1) unknown or difficult to calculate Projections: 3) **CPI** if 1) and 2) not available	In case private pensions are a minor source of income and average amounts or indexation rules are hard to obtain, discuss other options with the Essex developer.
Income from	**1) Growth in average amount**	Ideally, different sources of income

---

[10] Such information is of high importance and is extensively used for some research purposes (e.g. nowcasting poverty figures) and can be used to approximate the actual rules in case they are so complicated that it is not possible to follow them directly.

[11] Annual indexation rules can imply an uprating factor of 1 if no indexation took place.

capital[12]	2) **Growth in aggregate amount** (e.g. total dividend income, total interest income) 3) **Change in rate of return** (e.g. stock market index, average interest rate for household deposits) **4) GDP growth**	from capital (dividends, interest income, profit from unincorporated business) should be distinguished and updated separately using corresponding sources of information. If this is not feasible, then the second best approach is to apply uprating factors of the dominant income component to all income from capital. If the dominant income component is not possible to establish either (e.g. using national SILC) then use GDP growth.
Expenditures		
Expenditures	**CPI**	Note that for example, some tax allowances/deductions might be linked to earnings, in which case uprating by growth in average full-time equivalent earnings would be a more appropriate uprating factor. Also, if the model includes data on income <u>and</u> expenditure for any component (e.g. rent on property paid and received) then the uprating indexes should be the same for each.
Public pensions and benefits		
Public pensions	1) **Annual indexation and policy changes** 2) **Growth in average amount** if 1) unknown or difficult to calculate (annual indexation) or to model (policy changes) Projections: 3) **CPI** if 1) and 2) not available	If options (1) or (3) are used in the model, provide also **growth in average pension.**
Contributory benefits	1) **Annual indexation and policy changes** 2) **Growth in lagged average earnings** if 1) unknown or difficult to calculate 3) **Growth in average amount**	
Non-contributory	1) **Annual indexation and policy changes** 2) **Growth in average amount** if 1) unknown or difficult to	

---

[12] (Imputed) capital is not updated.

benefits	calculate Projections: 3) **CPI as a proxy of official indexation** if 1) unknown or difficult to calculate 4) **Uprating factor of 1** if 1) does not exist	
	Other	
Components that are used only for micro/macro-validation in the base-year	**Uprating factor of 1**	These are the components that we simulate in the model; and so they do not enter the simulations or any income list, e.g. tin (income tax), tscee (employee SIC), tscer (employer SIC), yds (disposable income).

- The model constructs uprating factors for each input dataset on the basis of the raw data, taking relevant year as the base period.[13] Table 2 sets out an example of the raw data series for prices and average wage.

Table 2: Raw data used for calculating uprating factors (Example)

Index	**2007**	**2008**	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**Original source and comments**
Harmonised CPI (index 2005=100)	117.3	135.2	139.6	137.9	143.7	147.0	149.7	152.9	Eurostat (link); 2014 - IMF forecasts (link)
Average monthly salary, gross, EUR	398.0	479.0	461.0	445.0	464.0	481.0	489.7	500.7	National Statistical Office (link), 2014 – national forecasts (link)

- An uprating factor equal to the Harmonised Consumer Price Index must be included and named $HICP (time series can be found on Eurostat website here (indicator prc_hicp_aind); projections of HICP published by DG ECFIN in their economic forecasts can be found here (see indicator ZCPIH) . All other uprating factors names should start with the suffix "$f_".

5 System and database configuration

- All policy system and input database combinations should be allowed by default (i.e. set to "x" in Country Tools > Databases). In exceptional cases where this is not feasible for certain system-database combinations, these should be set to "n/a" and the reasons explained in the model release documents.

---

[13] This will be introduced in version G3.0 and applied to the 2011 and later policy years (and, if possible, to earlier years as well). It represents an important change from the previous practice where instead constructed uprating factors (specific to the input dataset) were supplied.

- For each policy year, the (single) best combination/match with available input datasets – **baseline simulation** – is denoted by 'best' (in Country Tools > Databases). For a given policy year, the best dataset is (by priority):
    - (i) the input dataset which income reference period matches the policy year, e.g. EU-SILC 2012 (with incomes of 2011) for 2011 policy rules;
    - (ii) the latest input dataset which income reference period precedes the policy year, e.g. if EU-SILC 2008 and 2010 are both available then EU-SILC 2008 should be used for 2008 policy rules;
    - (iii) the earliest input dataset which income reference period follows the policy year, e.g. if only EU-SILC 2010 is available then this should be used for 2008 policy rules.
- All (allowed) policy-dataset combinations must run without producing errors and warnings, however, the checks of validity of the results (see Section 13) are generally limited to the baseline simulations only.
- Any dataset or policy system (or part of it) which are not publicly available (e.g. any system under construction or systems constructed for a specific project) must be defined as private.
- Exchange rate from euro to national currency should be as of June 30 in the corresponding policy year.
- Monetary policy parameters should be in the national currency. The choice is flexible for countries having policy years implemented before and after the euro adoption.
- Output data must be in the national currency (by default). In the case of euro-zone countries use the euro also for the systems preceding the euro adoption.
- All policy systems must have also training data defined.
- Each policy system and individual policy/function/parameter has a unique ID, which are critical for automated merging of different versions of the model. To ensure that unique ID-s remain constant (for baseline policy systems), only use add system/policy/function/parameter feature for genuine new elements and not for revising existing elements.[14]
- Hypothetical Household Data: Hypothetical Household datasets generated using the Hypothetical Household Tool (HHoT) are named ***CC_year*_hhot** where *cc* is country acronym, *year* refers to the data year. For example, *ES_2015_hhot.txt*. The hypothetical data is configured in a separate tab from other (micro) datasets. The following standardised specifications apply:
    - The currency needs to be defined in EUR as monetary variables are presented in EUR in HHoT.
    - Income and data year are the same as the year of the dataset. For example 2015 for the dataset ES_2015_hhot.txt.

---

[14] For example, if correcting 2015 policy parameters is done by first copy-pasting existing 2015 policies into a new policy system then all elements of the new system will be assigned new ID-s and hence no longer recognised as a revised version of the original 2015 system.

  - The field “Use common default” needs to be ticked on.
  - The default combination of HHoT data and policy system is the policy system with the same year as the dataset and should be set to “x”. Other combinations should be set to “n/a”. For example: the systems es_2007 to es_2014 are set to “n/a” for ES_2015_hhot.txt, while the system es_2015 is set to “x”. Preceding systems are again set to “n/a”.

6 Scope of policies

- Policies are simulated as of June 30 in the corresponding policy year.
- Policies to be simulated (at least partly) are
  - social insurance contributions: employer, employee, self-employed and credited contributions;
  - personal income taxes (excluding taxation of capital gains);
  - property, wealth and other personal direct taxes *if possible*;
  - cash benefits (including unemployment benefits);
  - minimum wage.
- Policies are to be simulated as detailed as possible given the underlying dataset. For example, all components of SICs are to be simulated and retained in separate variables (e.g. health insurance contributions by employers).
- Regional differences are to be simulated as far as possible.
- In general, policies must be implemented assuming full benefit take-up and no tax evasion. However, whenever the macro validation exercise shows that results produced by the model substantially deviate from comparable external sources, non take-up and tax evasion should be modelled (if possible). The baseline, in such cases, will be based on the parameterisation accounting for non-take-up and evasion (see also Section 8). Both versions of the model (with and without the adjustment for non take-up and tax evasion) must be validated and (substantial differences) documented in the Country Report. Where non take-up is modelled, random assignment etc is preferred over data eligibility approach.

7 Order and structure of policies

- Policy simulations should begin with the following sequence:
  - default values (SetDefault_cc),
  - Uprating by Average Adjustment (UAA_cc) *(if available)*,
  - uprating factors (Uprate_cc),
  - constant definitions (ConstDef_cc),
  - standard income list definitions (ILSDef_cc),
  - country-specific income list definitions (ILDef_cc),
  - tax unit definitions (TUDef_cc),
  - other policy switches (see Section 8) (to keep them outside the METR loop),

  - minimum wage (yem_cc), and
  - recoding negative values of income (neg_cc).[15]

- If pensions are uprated in bands (even only between two years and uprated as usual throughout the rest of the period), this should be implemented in a separate policy called Uprate_bands_cc, which should be placed right after (or as close as possible to) the policy Uprate_cc.

- [REC] If market incomes or non-simulated benefits (apart from pensions) are uprated in bands, this could be implemented either in Uprate_cc or in a separate policy (similar to Uprate_bands_cc).

- Simulation of a policy system (i.e. policy spine) should end with the following two policies: individual-level standard output (output_std_cc) and household-level standard output (output_std_hh_cc). In contrast to the former, the latter policy output_std_hh_cc should be always switched off.

- [REC] Each independent instrument should be modelled in a separate policy, e.g. every family benefit independent of others.

- Policy name should use the main output variable name of the policy (without _s) followed by _cc (e.g. tscee_be). In the comments column, instrument's title must start with either DEF, SWITCH, TAX, BEN or SIC. The title must specify the name of the instrument both in English and in native language in parenthesis.

- [REC] It is strongly recommended to store as many policy parameters as **constants** as possible and especially those repeatedly used throughout the model (e.g. minimum wage). It is also recommended to harmonise constant names in the following way e.g. $[policy name]_[r/thr/amt/inclt] where r = rate, thr = threshold, amt = amount, inclt = income limit. For example, the constant for the tax threshold in the income tax policy will be called $tin_thr.

- Constants used in more than one policy must be defined in a separate constant definition policy (ConstDef_cc).

- All monetary values (including those on monthly basis) must have the **period** defined.

- To store intermediate results in the baseline systems use either ready-defined variables (sinXX_s but not stmXX_s) or define temporary variables for intermediate results (preferably, using a prefix *i_** in the variable name).

- Whenever joint taxation is applied, the simulated tax must be allocated proportionally to the taxable income/tax base between the members of the assessment unit in the relevant policy (where the income tax is simulated).

- Each simulated benefit must be assigned to the most likely recipient within the assessment unit (often the head).[16]

---

[15] There are three important things to remember. First, uprating factors must be applied before any monetary variables are used. Second, income lists are often used for tax unit definitions, in which case they must be defined before. Third, it is preferential to define tax units before simulating minimum wage (note that individual tax unit must be defined anyway to simulate minimum wage).

[16] Assigning all benefits to one particular person allows to count the number of recipients more easily.

- Where (short-term) benefits are adjusted with the number of months in receipt, this should be modelled as the last step in the relevant policy (if possible). Where social insurance contributions contain fixed amount elements, these should be adjusted with the number of months in work.
- Where adjustments such as for benefit take-up or tax evasion rely on random numbers, these should be generated in the policy Random_cc for the whole population at the household level, assigning the same number for everyone in the household to ensure consistency with reform scenarios (see for example UK and BE). The assumption is that decisions about benefit take-up or tax evasion are taken at the household, rather than the individual level.
- If a policy for initialising variables exists, it should use the harmonised name InitVars_cc.
- Make sure that the Marginal Effective Tax Rate (METR) add-on works with the policy system and training data, if necessary make adjustment to account for country specifics. For the calculations of METRs, benefit take-up and tax compliance adjustments should be switched off.

8 Policy switches

- Adjustments for e.g. benefit take-up and/or tax evasion must be done by adding switchable policies such that the policy system runs with these either switched on or off without any further modifications (rather than having two separate systems and/or policies).[17]
- The following standardised policy switches are defined:
    - Benefit Take-up Adjustments (BTA_cc),
    - Tax Compliance Adjustments (TCA_cc),
    - Uprating by Average Adjustment (UAA_cc),
    - Full Year Adjustments (FYA_cc),
    - Pension Age Adjustment (PAA_cc),
    - Extended Policy Simulations (EPS_cc).

Table 3: The description and baseline settings of switch policies

Switch policy	**Set to OFF**	**Set to ON**	**Baseline value (if exists)**
BTA_cc	Full benefit take-up	Non-take up correction applied	Country-specific
TCA_cc	Full tax compliance	Incomplete tax compliance	Country-specific
UAA_cc	Respective benefits or pensions are uprated by the indexation rule	Respective benefits or pensions are uprated by the growth in the average nominal amount	Off
FYA_cc	Policies as of 30th June	Annual policies (reflecting policy changes over the year)	Off
PAA_cc	Labour market status and	Labour market status and incomes	Off

---

[17] See Administration Tools > Administrate Switchable Policies in the model.

	market incomes taken from the input data	(earnings, pension) adjusted for those affected by retirement age changes	
EPS_cc	Respective variables taken from the input data	Respective simulated variable used	Country-specific

- [REC] BTA_cc and TCA_cc policy switches should be activated in combination with training data where this is sensible, e.g. if the adjustment is implemented using a random adjustment (say 15% random non-take up).
- The policy switch Uprating by Average Adjustment (UAA_cc) should be implemented in the following way:
    - In the policy SetDefault_cc one should define a switch constant $uaa equal to 0.
    - After the policy SetDefault_cc, a switch policy called UAA_cc (Uprating by Average Adjustment) – kept off in the baseline – should be introduced where the switch constant $uaa is set to 1. When the policy is switched on, the value of $uaa will be overwritten with 1.

      In the function Uprate in the policy Uprate_cc, one should use first, the parameter factor_condition = {$uaa=1} for which the uprating factor is based on the growth in the average benefit/pension; second, the parameter factor_condition = {$uaa=0} for which the uprating factor equals the indexation rule.

      Thus, in the baseline, when the switch policy UAA_cc is off (i.e. $uaa=0), the respective benefit/pension will be uprated by the indexation rule. When the switch policy is on (i.e. $uaa=1), then the respective benefit/pension will be uprated by the growth in the average amount.
    - If pensions are uprated in bands in the policy Uprate_bands_cc, in the switch policy UAA_cc a ChangeParam function should be added which turns off Uprate_bands_cc. Furthermore, in the function Uprate in the policy Uprate_cc, for the condition factor_condition = {$uaa=0} the uprating factor should be set to 1.

      Thus, in the baseline, when the switch policy UAA_cc is off (i.e. $uaa=0), the respective pension will be uprated by a factor of 1 in Uprate_cc and then uprated in bands in Uprate_bands_cc. When the switch policy is on (i.e. $uaa=1), then Uprate_bands_cc will be turned off and the respective pension will be uprated by the growth in the average amount.
- Hypothetical Household Data: Switches for hypothetical household datasets need to be defined making sure that take-up adjustments and tax compliance adjustments are not taken into account.

9 Standard assumptions for simulating specific policies

9.1 Minimum wage

- Minimum wage must be simulated for all countries (even for those where it does not exist).
- The simulation would replace observed employment income with minimum wage which is adjusted with the number of months receiving employment income and the number of working hours (ignoring actual hours above the standard hours, i.e. overtime), whenever the former is lower than the later. The policy is kept switched off in the baseline.

9.2 Unemployment (insurance) benefit

- Unemployment benefits (UB) must be simulated for all countries.[18] The policy should be switched ON if the results of simulations are satisfactory (that is the simulated results in the base year correspond to the recorded amount and/or administrative record well in terms of aggregate and average amounts). If not, the policy should be set to TOGGLE. In either case, the implementation of UB should be updated annually and (if possible) the UB replacement rates should be checked.
- Unemployment benefit policy should be clearly marked as 'PART SIMULATED' (since eligibility is taken from the data). This should be marked in the comments fields next to the policy itself and next to the function restricting the simulation. If any other elements (duration, amounts, ceiling etc.) are taken directly from the data when simulating unemployment benefits, this should also be recorded as part simulation in the comments section.
- Three groups of 'potential' recipients of UB are distinguished in simulations and treated differently:
    - Individuals currently in receipt of UB in the data **({bun > 0})** should be eligible for receiving the simulated UB.
    - Individuals with unemployment spells and not in receipt **({lunmy_s > 0} & {bun = 0})** should be included into simulations, but their eligibility to UB should be restricted by setting the number of months worked in the qualifying period to 0 (liwmy_s = 0 if bun = 0).
    - Employed individuals with no unemployment spells are potential new unemployed and might be turned into such for specific applications (e.g. modelling employment transitions for labour market adjustments or calculating replacement rates). Previously these individuals were identified as ({ils_earns != 0} & {bunct = 0})**,** now they are included in simulations through the condition **{lnu > 0}**. Variable lnu (default is 0) is created within the Labour Market Adjustments (LMA)

---

[18] This will be used for the calculation of replacement rates and when adjusting data for changes in the labour market.

  or Net Replacement Rates (NRR) Add-Ons and serves as an identifier for those who undergo transition into unemployment.[19]

- Given that the data available in most countries is insufficient to fully or accurately simulate UB, the following imputations and assumptions will be used.
- **(a) Eligibility**
  - **Qualifying period**: most countries require a minimum number of contributions (or months in work) over a period of time. Usually the requirement is about 12 months of contributions in the last 18 to 36 months preceding unemployment. The minimum required number of months in work and the total number of months to be checked should be recorded as constants: e.g. $UB_QperMin and $UB_QperTot. Note, that the information available in the EU-SILC is number of months spent at full-time/part-time work in *income reference period*.
    - unemployed and in receipt: use the observed value for the income reference period and assume it is representative to the whole qualifying period (i.e. **liwmy*$UB_QperTot/12**), but not exceeding the total no of months worked (**liwwh**). As individuals in this group are eligible assume that the number of contributions is at least equal to the minimum legal requirement. In cases where self-employed are not eligible for UB receipt altogether, use yemmy instead of liwmy.
    - unemployed and not in receipt: set to 0.
    - new unemployed: equal to **liwmy_a** (default is 0). The variable is calculated in the LMA or NRR add-on in the same way as for group (1), but it is not restricted to be at least equal to the legal requirement.
- **(b) Benefit amount**
  - Previous contribution: in many countries the benefit amount depends on the amount of previous contributions (or on previous net/gross earnings). Since previous contributions/earnings are not available in the EU-SILC, assume previous earnings are equal to those predicted by wage equation (yivwg derived in the Stata do-files) and, if needed, contributions calculated over such earnings (can be calculated in EUROMOD applying SIC rules).
    - unemployed and in receipt
      - (a): impute previous wage based on the estimation of a wage equation (yivwg) assuming full-time employment as defined in the country (i.e. **yivwg*40*52/12** for per month amount)
      - (b): previous wage obtained by reverting UB rules (yempv)[20]
    - unemployed and not in receipt: set to 0

---

[19] Once the NRR Add-on is implemented and tested for all countries other functions and elements used in the past within the country files for calculation of replacement rates should be dropped from the model.

[20] Regarding options a) and b), our intention is to use yivwg. However, we are still in the process of testing the estimation of yivwg. Until this is resolved, a constant (e.g. named $ImputedWage) can be used to switch between

    - new unemployed: set to be equal to **yempv_a** (default is 0). The variable is calculated in the LMA or NRR add-on based on the actual observed earnings (e.g. yem).
  - Training: in some countries UB is complemented when attending vocational training courses. This should be implemented as an option that may be switched on/off (by default set to TOGGLE).
- **(c) Benefit duration**
  - Number of previous contributions: in some countries the maximum duration of the benefit depends on the total number of contributions. In the EU-SILC there is a proxy variable: number of years spent in paid work (PL200).
    - all groups: assume total number of months making contributions = PL200 * 12 (i.e. liwwh).
    - unemployed and in receipt: the duration of UB must be limited to the number of months in receipt of unemployment benefit (EUROMOD variable bunmy).
- **(d) Compatibility with the add-ons (LMA and NRR)**
  - If not ON in the baseline the UB policy should be set to TOGGLE (rather than off). All additional functions within the UB policy that are OFF in the baseline (if any) should also be set to TOGGLE instead.
  - Constants required for calculations of the qualifying period ($UB_QperMin, $UB_QperTot) should be defined in ConstDef_=cc=
  - Variables used in the add-on's should be set to 0 as default in SetDefault_=cc=. These include:
    - lnu – a dummy variable for new unemployed
    - liwmy_a – number of months in work in the year preceding unemployment (for new unemployed)
    - yempv_a – monthly gross wage in the year preceding unemployment (for new unemployed)
    - yem_a (previously yem00) – monthly gross wage for new employed
    - lhw_a (previously lhw00) – average weekly working hours for new employed
  - There might be other country specific variables, e.g. liwmy01_a and liwmy02_a in case of PT. They should be treated in the same way.

10 Tax units

- The name of each tax unit must start with a prefix "tu_".

---

yempv and yivwg. This might be used as a temporary solution while yivwg is being tested or as a permanent switch (in case relevant for a country).

- Tax units used in more than one policy must be defined in the tax unit sheet(s).
- For the purpose of METR add-on, individual and household unit must be defined (i.e. tu_individual_xx and tu_household_xx, respectively).

11 Income lists

- The name of each standard income list must start with a prefix "ils_" and all country-specific income lists must start with a prefix "il_".
- All income lists must have a description (next to where they are defined).
- All standard income lists are compulsory and must be defined in the standard income list policy (ILSDef_cc).
- All country-specific income lists used in more than one policy must be defined in the income list policy (ILDef_cc), others which are relevant only for a single policy can be defined in the corresponding policies.
- Make sure that no variable is double counted in income lists, e.g. a benefit component separately and in aggregated variable, both data and simulated variable.
- [REC] Income lists should include detailed incomes rather than their aggregates (here referring to variables not other income lists). For example, assuming there are two unemployment benefits and both need to be included, it is better to have each component separately rather their aggregate.

11.1 Definitions of standard income lists

- Earnings (ils_earns) – labour earnings (e.g. employment and self-employment income)
- Original income (ils_origy) – i.e. market income with the following components[21]:
    - Earnings (+)
    - Income from capital, e.g. dividends and interests (+)
    - Income from occupational and private pensions (+)
    - Income from property (+)
    - Income received by children (+)
    - Regular inter-household cash transfer *received* (+)
    - Regular inter-household cash transfer *paid* (-)
- Original income and replacement incomes (ils_origrepy) – the latter referring to contributory benefits intended to provide incomes for specific life cycle (e.g. old age, maternity, unemployment, disability, invalidity[22])
- Employer social insurance contributions (ils_sicer) – including (employer) payroll taxes
- Credited social insurance contributions (ils_sicct) – contributions paid by government or social security institution on benefits (if these are simulated)

---

[21] This should exclude capital gains and other lump-sum incomes (e.g. lottery winnings). Severance pay should be included in non-means tested benefits.

[22] Disability and invalidity benefits are not strictly replacement incomes. However (UDB) SILC does not allow separating such benefits from old-age pensions so they are included in the replacement incomes.

- Employee social insurance contributions (ils_sicee) – also including compulsory private pension contributions and *ils_sicot* (see below)
- Self-employed social insurance contributions (ils_sicse)
- Other social insurance contributions (ils_sicot) – contributions paid by individuals but not directly linked to employment or self-employment (e.g. SIC due on benefits and paid by the benefit recipients, health contributions paid by general population)
- Public pensions (ils_pen) - contributory benefits related to old age, survivors', disability and early retirement as well as integral non-contributory elements (e.g. minimum pensions)
- Means-tested benefits (ils_benmt)
- Non means-tested benefits (ils_bennt)
- **Total benefits (ils_ben) constructed as ils_ben = ils_pen + ils_benmt + ils_bennt**[23]
- Simulated benefits (ils_bensim)
- Taxes (ils_tax)
- Simulated taxes (ils_taxsim)
- **Disposable income (ils_dispy) constructed as ils_dispy = ils_origy + ils_ben – ils_sicee – ils_sicse – ils_tax**
- Benefits by function following Eurostat definitions (ESSPROS & SILC):
    - Family benefits (ils_b1_bfa) – benefits related to the cost of pregnancy, childbirth and adoption, bringing up children and caring for other family members (classification corresponding to SILC variable HY050G);
    - Education benefits (ils_b1_bed) - grants, scholarships and other education help received by students[24] (classification corresponding to SILC variable PY140G);
    - Old-age benefits (ils_b1_boa) – income maintenance and support in connection with old age (classification corresponding to SILC variable PY100G);
    - Survivor benefits (ils_b1_bsu) – income maintenance and support in connection with the death of a family member (classification corresponding to SILC variable PY110G);
    - Disability benefits (ils_b1_bdi) – income maintenance and support (except health care) in connection with the inability of physically or mentally disabled people to engage in economic and social activities (classification corresponding to SILC variable PY130G);
    - Unemployment benefits (ils_b1_bun) – income maintenance and support in cash or kind in connection with unemployment (classification corresponding to SILC variable PY090G);

---

[23] Public pensions, means-tested benefits and non means-tested benefits are here meant as mutually exclusive categories.

[24] Not counted as social benefits in ESSPROS.

  - Health/sickness benefits (ils_b1_bhl) – income maintenance and support in connection with physical or mental illness, excluding disability; health care intended to maintain, restore or improve the health of the people protected irrespective of the origin of the disorder (classification corresponding to SILC variable PY120G);
  - Housing benefits (ils_b1_bho) – help towards the cost of housing (classification corresponding to SILC variable HY070G);
  - Social assistance/exclusion benefits (ils_b1_bsa) – benefits (except health care) specifically intended to combat social exclusion where they are not covered by one of the other functions (classification corresponding to SILC variable HY060G).
  - **The sum of all ils_b1* income lists must equal total benefits (ils_ben).**
- Benefit by function (semi-aggregated categories):
  - Family and educations benefits (ils_b2_bfaed) constructed as ils_b2_bfaed = ils_b1_bfa + ils_b1_bed
  - Pensions, disability and health benefits (ils_b2_penhl) constructed as ils_b2_penhl = ils_b1_boa + ils_b1_bsu + ils_b1_bhl + ils_b1_bdi
  - Social assistance and housing benefits (ils_b2_bsaho) constructed as ils_b2_bsaho = ils_b1_bsa + ils_b1_bho
  - **Together with unemployment benefits (ils_b1_bun), the sum of ils_b2* income lists must equal total benefits (ils_ben): ils_ben = ils_b1_bun + ils_b2_bfaed + ils_b2_penhl + ils_b2_bsaho.**
- Income lists corresponding to UDB EU-SILC definitions of market income, tax and benefit variables – NB! May include variables not part of EUROMOD standard concept of disposable income (ils_dispy) and vice versa:
  - Employment income (ils_udb_yem)
  - Self-employment income (ils_udb_yse)
  - Investment income (ils_udb_yiy)
  - Pensions from individual private plans (ils_udb_ypp)
  - Income from rental of a property or land (ils_udb_ypr)
  - Income received by people aged under 16 (ils_udb_yot)
  - Regular inter-household cash transfers received (ils_udb_ypt)
  - Regular inter-household cash transfers paid (ils_udb_xmp)
  - Company car (ils_udb_kfbcc)
  - Regular taxes on wealth (ils_udb_tpr)
  - Tax on income and social contributions (ils_udb_tis)
  - Family benefits (ils_udb_bfa)
  - Education benefits (ils_udb_bed)
  - Old-age benefits (ils_udb_boa)
  - Survivor benefits (ils_udb_bsu)
  - Disability benefits (ils_udb_bdi)
  - Unemployment benefits (ils_udb_bun)
  - Health/sickness benefits (ils_udb_bhl)
  - Housing benefits (ils_udb_bho)
  - Social assistance/exclusion benefits (ils_udb_bsa)

    - Disposable income (ils_udb_yds) – the sum of all ils_udb* income lists, where ils_udb_xmp, ils_udb_tpr and ils_udb_tis are included with a negative sign.

12 Output

- Results must be outputted at the **individual level** by default.
- Output (at the individual level) needs to include:
    - all income variables from data which are not simulated in the model (including fringe benefits),
    - all simulated income variables,
    - all socio-demographic variables,
    - all income lists,
    - yds: household disposable income as reported in original data (e.g. HY020) entirely attributed to one member of the household,
    - [REC] identifiers for all tax units.
- Make sure that output policies does not have any variable (or income list) twice or any temporary/intermediate variables.

13 Validation

- Micro validation:
    - Check eligibilities and the amounts of taxes and benefits simulated by the model (case-by-case validation for a selection of particular households)
    - Compare simulated values against data recorded values in the same survey on case-by-case basis
    - Check descriptive statistics for outcome variables (min, max, mean)
    - Check that results for some basic indicators (e.g., average tax rate) make sense for all observations in the sample
    - [REC] Check budget constraint graphs for different types of hypothetical households
- Macro validation:
    - Compare the sum of each income component and the number of recipients (both original incomes and tax-benefit instruments) with external statistics; simulated values can be also compared against data recorded values in the same survey. For simulated instruments, this should answer the question how <u>close</u> EUROMOD estimates are to the external figures given the following factors:
        a) quality of external statistics
        b) survey quality, i.e. how representative market/non-simulated incomes and population structure are, measurement errors etc

c) quality of data imputations (e.g. net-to-gross imputations, income splitting, replacing missing values)

d) simulation quality, i.e. accuracy of tax-benefit rules

e) key modelling choices and assumptions (e.g. adjustments for benefit non-take up and tax evasion, the 30th of June rule etc)

- In general, the focus should be on relative differences, e.g. how one instrument compares to another, whether the bias has an expected sign and whether trends over years are in the expected direction.
- Calculate and compare inequality measures (Gini, S80/S20 ratio) and poverty measures with Eurostat statistics based on SILC or external statistics (if relevant)[25]. The latter should address the question how different EUROMOD estimates are due to differences in the underlying methodology. The following sources of bias are justified (but need to be acknowledged of course):
  - sample adjustments (e.g. removing children born after income reference period from the sample)
  - differences in the concept of disposable income
  - data imputations, i.e. (c) above
  - differences between data and simulated values, that is (d) and (e) from above combined.
- [REC] Produce budget constraint graphs for different household types with the hypothetical data.

---

[25] For such comparisons, indicators need to be calculated in the same way for both EUROMOD output and SILC original data. This means the same (i) equivalence scale, (ii) treatment of zero and negative incomes, and (iii) approach to bottom- and top-coding.