# Section 3 – Using CAPITA

**Date last modified: 28 September 2021**

## Introduction

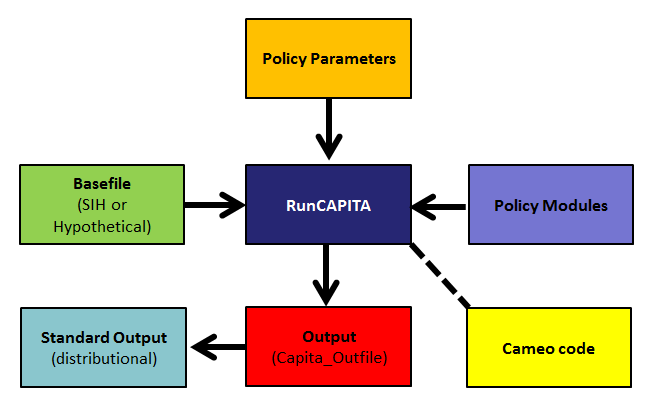
This section provides information on how to use CAPITA, including:

* [Using RunCAPITA](#_Section_2.1_–) (Section 3.1).
* [Conducting Analysis](#_Section_2.2_–) (Section 3.2).
* [Modelling Policy Changes](#_Section_2.3_–) (Section 3.3).

# Section 3.1 – Using RunCAPITA

## Introduction

*RunCAPITA.sas* is the central program for running the CAPITA policy code. *RunCAPITA* accepts user inputs to define the period of analysis and reads in the corresponding basefile and policy parameters (such as payment rates and thresholds). It then runs each of the policy modules sequentially to calculate the tax and transfer outcomes for each of the income units on the basefile. The outcomes are then added to the basefiles and output to the dataset **Capita\_Outfile.**



**Figure 3.1: The structure of *RunCAPITA*.**

Table 3.1 presents a summary of the policy modules in CAPITA and the purpose of each module. The Variable Register provides the definitions of the variables used in the model.

**Table 3.1: Summary of the policy modules in CAPITA and the functions of each module.**

|  |  |  |
| --- | --- | --- |
| Initialisation | Initialise and label all variables, both character and numeric. |  |
| Income1 | Calculate all incomes and assets used in means testing of DVA, Pensions, Allowances and Supplements. |  |
| Dependants1 | Calculate number of dependants according to the various definitions used for social security payments. |  |
| DVA | Calculate eligibility and entitlement for DVA service pension. | DVA Service Pension |
| Pensions | Calculate eligibility and entitlement for all pensions. | Age Pension  Disability Support Pension  Carer Payment  Parenting Payment Single  Wife Pension[[1]](#footnote-1) |
| Allowances | Calculate eligibility and entitlement for all allowances. | Austudy  Parenting Payment Partnered  JobSeeker Payment[[2]](#footnote-2)  Widow Allowance[[3]](#footnote-3)  Youth Allowance |
| Income 2 | Calculate all incomes used in means testing of family tax benefit and income tax calculations | Taxable transfer income  Taxable income  Rebate income  Net income from working  Rebatable benefit  Income for surcharge purposes  Adjusted taxable income |
| Dependants2 | Calculate number of dependants according to Medicare levy definitions | Dependants for Medicare Levy  Dependants for Medicare Levy surcharge |
| Family Tax Benefit | Calculate eligibility and entitlement for all family payments. | Family Tax Benefit Part A  Family Tax Benefit Part B  Newborn Supplement |
| Supplements | Calculate eligibility and entitlement for supplements. | Carer Allowance  Carer Supplement  Pensioner Education Supplement  Telephone allowance  Utilities allowance |
| Tax | Calculate income tax including the Medicare levy and eligibility and entitlement for all tax offsets. | Income tax  Senior Australian and Pensioner Tax Offset  Beneficiary tax offset  Low income tax offset  Low and middle income tax offset  Dependent (Invalid and Carer) Tax Offset  Dependent Spouse Tax Offset and Mature Age Workers Tax Offset (before 1 July 2014)  Medicare levy  Medicare levy surcharge |
| Childcare (Cameo model only) | Calculate Child Care Benefit and Child Care Rebate (current system), or Child Care Subsidy (new system) | Child Care Benefit (before 30 June 2018)  Child Care Rebate (before 30 June 2018)  Child Care Subsidy (from 1 July 2018) |
| Finalisation | Calculate disposable income and income unit level totals plus ad hoc tidying. |  |

## How *RunCAPITA* is used

There are three main contexts in which the *RunCAPITA* program is used:

1. **Generating distributional model outcomes:** Once the distributional basefiles have been created (see ‘Section 4 – Guide to CAPITA Basefiles’), they are used as inputs to *RunCAPITA,* which is run for the desired period of analysis to produce the distributional model outcomes for the *current* tax and transfer system in a SAS dataset called **Capita\_Outfile***.* More commonly, to examine the impact of policy changes, the current system is compared with a simulated policy world, which models proposed policy changes. In this case, *RunCAPITACompare.sas* is used to run first the current (or base world) policy modules and secondly the simulation world policy modules. These are then combined to compare the base and simulation world policy outcomes in the SAS dataset **CAPITA\_Compare**. The base world policy is reflected in the ‘Policy Modules’ folder, and where the simulated world policy modules differ from those in the base world, they are reflected in the ‘Policy Modules (Sim)’ folder. Variables are distinguished with ‘\_Base’ and ‘\_Sim’ suffixes. Base world policy and simulated world policy modules should be run on base world basefiles and simulated world policy basefiles respectively as there are some interactions between the creation of basefiles and policies, i.e. basefiles for different world could differ due to the different polices it based on.
2. **Generating cameo model outcomes:** Alternatively, *RunCAPITA* can be run on a hypothetical basefile containing the family attributes and income levels pertaining to a cameo scenario of interest. The *Cameo Code.sas* program will generate a hypothetical basefile and call *RunCAPITA* to run on this hypothetical basefile. Current policy or simulated policy outcomes can be generated by altering the policy code as required.
3. **Benchmarking to administrative data as part of the basefile creation process:** *RunCAPITA* is also called in the final stage of the basefile creation process, to create flags to indicate which payments individuals receive. These are then used as inputs into the reweighting process to benchmark to administrative data on customer numbers for the transfer payments.

## Running *RunCAPITA / RunCAPITACompare*

The following steps can be used to run *RunCAPITA* / *RunCAPITACompare* (these modules are located in the parent folder of the Current Version of the model).

**Step 1: Specify the Capita Directory**

Specify the CapitaDirectory. This is contained in the module *DefineCAPITADirectory.sas*. This needs to be updated to reflect the location of the CAPITA folders on your network drive, and then run, to define the required directory. Note also that the file path needs to end with a /. Once run, this file can be closed.

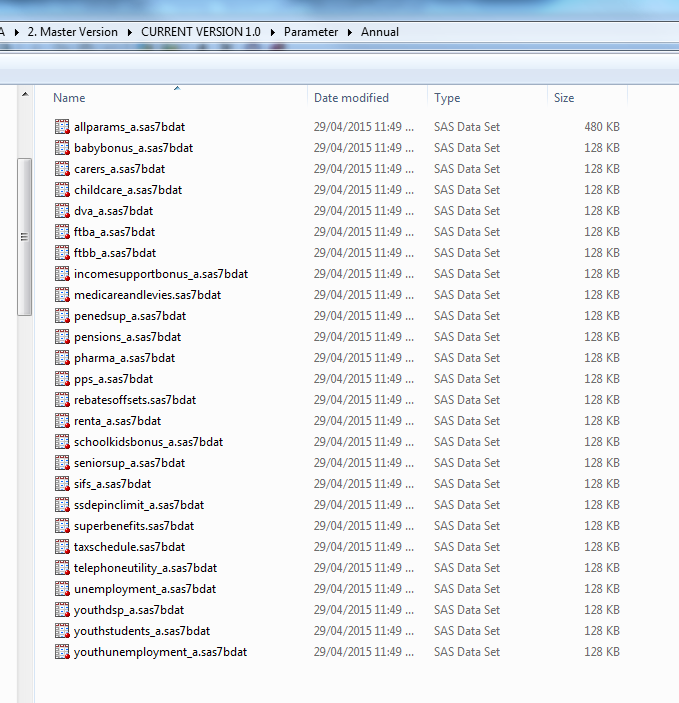
*DefineCAPITADirectory.sas* needs to be run each time you begin a new SAS session.

The CapitaDirectory and other paths in the model are based on the Treasury’s network structure. Non-Treasury users would need to change the paths and directories in the model based on their own network drive structure.

**Step 2: Generate the parameters (if not yet generated, or if changed)**

The CAPITA parameters are stored in a spreadsheet called the Common Parameter Spreadsheet (CPS). If you have made any changes to the parameters in the CPS, or if you are generating the parameter datasets for the first time, you will need to run[[4]](#footnote-4) *RunParameters.sas*. This converts the parameters contained in the CPS into SAS datasets containing the parameters ready for CAPITA to use, and stores them in the ‘Annual’ and ‘Quarter’ folder of the ‘Parameter’ folder (the ‘Annual’ folder is used for annual runs of the model, and contains weighted averages of the quarterly parameters applying across the year).

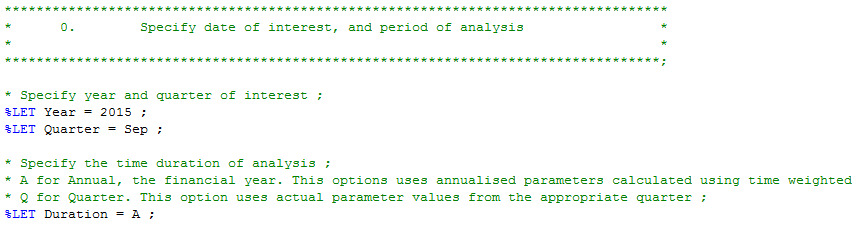
If the parameters for base world and simulated world are different, you need to run *RunParameters.sas* separately and save to the respective sub-folder.



**Figure 3.2: Example screenshot of the Parameter folder.**

*RunCAPITA* will then read in these datasets and attach them to the basefile, before running the policy modules. In addition to the parameter datasets created here, *RunCAPITA* also attaches parameter datasets containing grandfathering proportions by type of transfer payment (ProbGrndfthr\_Q and ProbGrndfthr\_A). These do not need to be created in *RunParameters*, but will need to be updated in the Parameter folder as the forward estimates of grandfathering proportions change. More detail on the modelling of the grandfathered removal of the Energy Supplement is provided in Section 5.2.

**Step 3: Specify the period of analysis**

In *RunCAPITA/RunCAPITACompare, s*pecify the Year and Quarter of analysis. If performing an annual run, the Quarter is not used, and the Duration is set to A. If performing a quarterly run, the Quarter should be set equal to the last month of the desired quarter (i.e. Sep, Dec, Mar or Jun), and the Duration set to Q. Note that the years are specified as 2017 for 2017-18, 2017 for 2017-18, and so on.****

**Figure 3.3: Screenshot of the Year, Quarter and Duration specifications in *RunCAPITA***

**Step 4: Run *RunCAPITA*/*RunCAPITACompare***

Now run *RunCAPITA / RunCAPITACompare*. The code calls each policy module sequentially using %INCLUDE statements and creates the SAS datasets **CAPITA\_Outfile** / **CAPITA\_Compare** that can be used for analysis (these are produced in the SAS Work directory). If you need to retain your outfile for use later, you will need to save a copy of the outfile dataset onto your network drive before closing.

Note that the *Childcare.sas* policy module is only run if *RunCAPITA / RunCAPITACompare* is called via the *Cameo Code*.

# Section 3.2 – Conducting Analysis

## Introduction

This section describes suggestions for conducting analysis using CAPITA. The main types of analysis undertaken using CAPITA are distributional analysis and cameo analysis (including analysis of Effective Marginal Tax Rates (EMTRs)).

## Distributional Analysis

### *Capita\_Outfile*

As discussed in Section 3.2 above, *RunCAPITA* produces the tax and transfer outcomes (as variables which get added to the basefiles) in a SAS dataset called **Capita\_Outfile** in the Work directory of your current SAS session*.* If a base world and simulation world comparison is being run using *RunCAPITACompare*, the output dataset is called **Capita\_Compare**, and the variables have \_Base and \_Sim suffixes attached.

#### Summary data

SAS DATA steps can be used on the **CAPITA\_Outfile** / **CAPITA\_Compare** to examine the distributional outcomes of the tax and transfer system, and to interrogate the outcomes in various ways depending on the purpose of the analysis. For example, a subset of the outfile, where the pension type is ‘AGE’, can be extracted to examine outcomes such as average payment rates for age pensioners. If a simulation world comparison has been run, a subset WHERE the change in disposable income is nonzero can be used to examine the records impacted by the policy changes. These datasets can be subsetted further, to partition outcomes into particular payments or payment structures (for example, all full rate FTBA recipients or all part rate FTBA recipients). A number of flags have been created in the model to facilitate such analysis[[5]](#footnote-5). The CAPITA variable register is useful for identifying variables on the outfile, and can be found in the parent folder of the current version of the model.

### *Standard Output*

The standard output code presents information in tables and charts to assist the user with routine analysis, and for testing the model. The files required are:

* *StandardOutput.sas*: Produces results as SAS datasets and outputs them to Excel.
* *Capita SO.xlsx:* Excel document containing the output data presented in tables and charts. The SAS code makes a copy of the SO template and writes the output to a time stamped copy of this workbook.
* *RunCapitaCompare.sas*: Used to call the *StandardOutput.sas* program, and create the required input datasets.

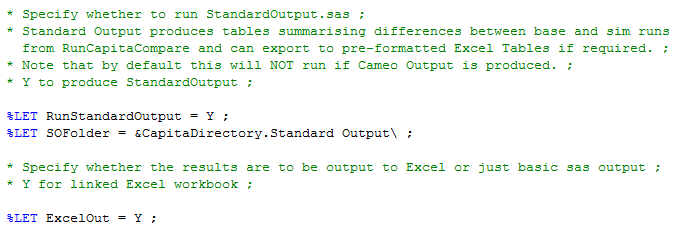
The standard output program is used to summarise and compare certain information from two sets of outfiles, usually between a base world and sim world outfile. The inputs required by *Standard Output* are a base world outfile and a sim world outfile (or basefiles) that are created by *RunCapitaCompare*, and the output is written to and presented in Capita SO.xlsx. Below is a guide to running standard output.

**Step 1: Set up the Standard Output Folder**

The Standard Output folder contains the files *StandardOutput* and Capita SO template.xlsx and is set up ready for running analysis. It is advisable, especially when multiple users are running analyses concurrently, to create a new folder which is a copy of the Standard Output folder, so that the *StandardOutput*, Capita SO.xlsx, and *RunCapitaCompare* files can be updated for the particular run being undertaken. This avoids conflicts arising from co-authoring.

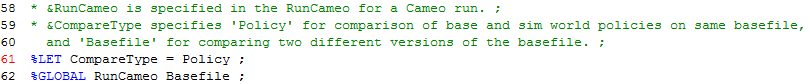
**Step 2: Adjust settings in *RunCapitaCompare* and run *Standard Output***

To produce a standard output the user should first open their copy of *RunCapitaCompare* and set both RunStandardOutput and ExcelOut to ‘Y’. These switches are to instruct *RunCapitaCompare* to call the *StandardOutput* program, and to write the output to the Excel file for presentation.



**Figure 3.4: Screenshot of the specifications in RunCAPITACompare related to Standard Output.**

Next select the type of comparison required, that is between a base and sim world outfile, or between two basefiles. Set CompareType to ‘Policy’ or ‘Basefile’ accordingly.



**Figure 3.5: Screenshot of the CompareType specification in RunCAPITACompare.**

Next run *RunCapitaCompare* to produce the output, with the desired set of base world and sim world policy code and parameters. *Standard Output* is called as a subroutine in *RunCapitaCompare*, and in most cases the code does not need to be opened or changed. The execution also produces a large number of datasets that are the source of the output, and can be examined for debugging purposes.

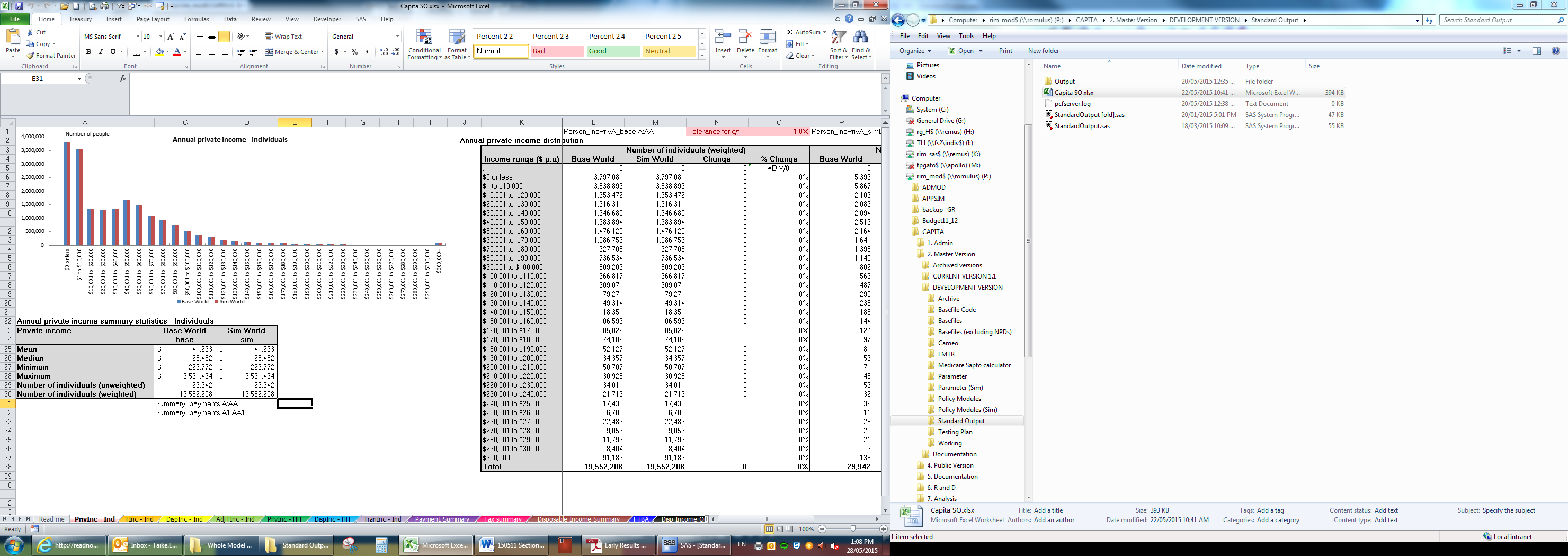
**Step 3: Examining the output in CAPITA SO.xlsx**

It is advisable that after every standard output run to keep a copy of the results, the user renames CAPITA SO.xlsx to reflect the scenario that was run. This way the user avoids having to go into *Standard Output* code to change the file name.

CAPITA SO.xlsx consists of two broad parts. The first part is a series of worksheets, each with tables and charts that are a permanent fixture of the file. These are ‘templates’ for the standard output. The second part consists of sheets that contain the raw information output directly from SAS and are updated after each run. They are the ‘source’ data for the standard output. The template sheets present the raw data in tables and charts.

Information presented includes the population distribution by a number of income measures such as private income, disposable income and taxable income. Also key statistics such as mean, median, maximum and minimum dollar amounts for transfer payments and tax elements. The user can produce additional results for their analysis if required.

It is important to note that the output is standardised, generic, and should be kept simple and manageable.



**Figure 3.6: The Standard Output spreadsheet.**

### *Unit record analysis*

‘Unit-record analysis’ (looking at outcomes for individual records) is important for testing changes to the model. This can be done using DATA steps as described above, by filtering using the ‘hide’ and ‘where’ functions in the SAS dataset viewer, or by using FSBrowse[[6]](#footnote-6).

## Cameo Analysis

Cameos in CAPITA are run using:

* A Cameo Input.xlsx spreadsheet, which specifies the hypothetical families for which cameos are to be generated;
* The *Cameo Code.sas* and *Cameo Initialisation.sas* modules, which create the hypothetical basefile based on the specified family characteristics and call RunCAPITA to generate the model outcomes for the hypothetical families;
* A Cameo Output.xlsx spreadsheet, in which the cameo results are produced (one tab for each family type).

The process for setting up and running cameos is provided below.

**Step 1: Specify the Capita Directory**

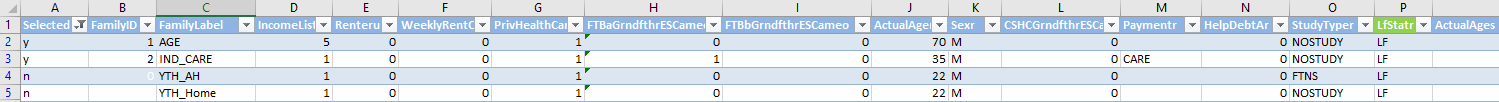
As discussed in Step 1 of the RunCAPITA steps above, first specify the CapitaDirectory.

**Step 2: Set up the Cameo Folder**

Create or assign a folder for storing the Cameo Input.xlsx, *Cameo Initialisation* and *Cameo Code*. Copy these files from the CAPITA cameo folder.

**Step 3: Specify the hypothetical families in the Cameo Input.xlsx spreadsheet**

Open the Cameo input.xlsx spreadsheet. On the ‘Family’ sheet, specify the details of the families you wish to generate (or choose from the default cameos already in the list).



**Figure 3.7: The Cameo Input spreadsheet.**

Details to specify are as follows:

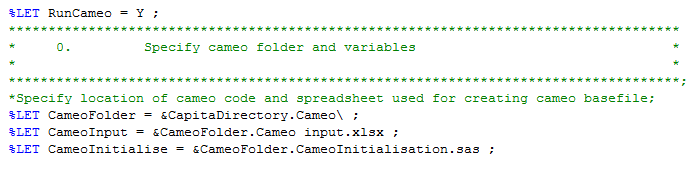
|  |  |
| --- | --- |
| **Variable** | **Description** |
| FamilyID | Family identification number |
| FamilyLabel | Family description |
| IncomeList | Name of tab defining income list to be used for this cameo. Cameo results will be generated for each reference and spouse income combination specified on the income list. |
| Renteru [0,1] | Rent status  0 = not renting  1 = renting privately |
| WeeklyRentCameo | Amount of private rent paid per week |
| PrivHealthCameo | Private Health Insurance status  1 = Has Private health insurance  0 = No Private health insurance |
| ActualAge[r,s] | Age of person |
| Sex[ r,s] | Sex of person  M = Male  F = Female |
| Payment[r,s] | Pension type of person  CARE = Carer payment recipient  DSP = Disability support pensioner  Blank if not applicable. Note both parents are assumed to be eligible for the Age Pension, Parenting Payment Single, Parenting Payment Partnered, JobSeeker Payment, and Youth Allowance if they satisfy age and parenting criteria. If required, more payments can be added in the code following the DSP example. This is explained in the *Guide to the Cameo Code.* |
| HelpDebt[r, s] | HELP debt status. If no HELP Debt, then set equal to zero. If HELP Debt is present then set to any positive value, size of debt is not important. We assume no voluntary payments, nor partial payments if level of debt is lower than calculated payment amount.  0 = no HELP Debt  >0 has outstanding HELP Debt |
| StudyType[r,s,1-4] | Study type of person  SS = Secondary school student  FTNS = Full time non-secondary student  PTNS = Part time non-secondary student  NOSTUDY = Not studying |
| LfStat[r,s] | Labour force status of person  LF = In labour force  NILF = Not in the labour force |
| ActivHr[r,s] | Number of hours worked per week. Used to determine how many hours of subsidised childcare the family is eligible for. |
| AgeOfKid[1-4] | Age of child |
| CcbType[1-4] | Childcare type for childcare benefit  LDC = Long day care  FDC = Full day care  FDC/INC – SHr = In-home care (standard hours)  FDC/INC – NSHr = In-home care (non-standard hours)  OSHC = Outside schools hours care  NONE = not using childcare |
| CcsType[1-4] | Childcare type for childcare subsidy  LDC = Long day care  FDC = Full day care  OSHC = Outside schools hours care  NONE = not using childcare |
| CcHrW[1-4] | The actual hourly cost for child care for each child paid by parents per week. |
| CcHrCost[1-4] | The actual hourly cost for child care for each child paid by parents. |
| CcwperYr[1-4] | The number of weeks per year that each child is in a particular type of child care. |

Note: r = reference; s = spouse; 1-4 = kid1 to kid4

Finally, select the cameos to run by putting Y in the ‘Selected’ column. Then save and close the spreadsheet.

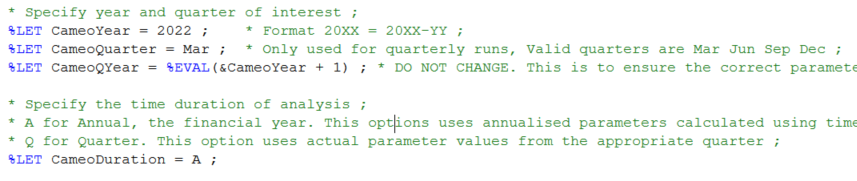
**Step 4: Setup and Run the *Cameo Code***

Open *Cameo code*. In the Cameo Folder definition, specify the location of the cameo folder in which you are working.



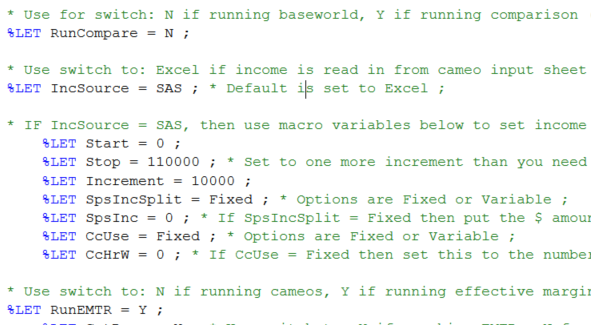
**Figure 3.8: Specification of the directories used in the Cameo Code.**

Also, specify the year, quarter of interest (if quarterly cameos are being run), and cameo duration.



**Figure 3.9: Specification of the parameters used in the Cameo Code.**

Specify whether just base world cameos or both base and simulation world cameos are being run, by setting RunCompare to N or Y respectively. Note that if simulation world cameos are being run, the variables with \_Base suffixes will reflect the base world (the modules in the Policy folder), the variables with \_Sim suffixes will reflect the simulation world (the modules in the Policy (Sim) folder), and the variables with \_Change suffixes are the change in the variable from the base world to the simulation world. For more information on running cameos for a package of new policies (for example, a package of Budget measures), see Section 3.4 (Modelling Policy Changes) below. Make sure that the RunEMTR switch is set to N when running cameos.

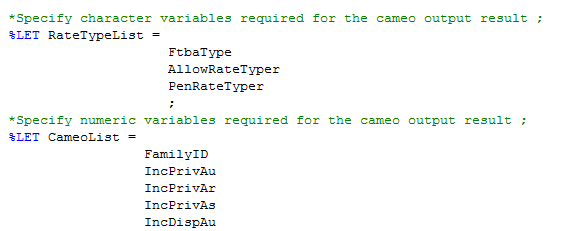


The IncSource toggle is used to specify the where the income for each of the cameos in the Cameo Input.xlsx will be sourced from. The toggle can either be set to “SAS” or “Excel”. If set to “Excel”, the incomes for each of the cameos will be imported from the Cameo Input.xlsx and should be set in the income sheets. If set to “SAS” the incomes will be calculated using the following variables.

|  |  |
| --- | --- |
| CcUse | Method of determining weekly childcare hours  Fixed = Childcare hours will be a fixed number specified by CcHrW.  Variable = Childcare hours will be a function of the end value of for the reference person’s income. Childcare usage is taken to be relative to the hours worked by the reference person. The default assumed weekly childcare hours for each child in childcare are:   * 10 hours per day worked for LDC and FDC, and * 5 hours per day worked for OSHC.   These assumptions can be updated in the SASIncome macro in the cameo code. |
| CcHrW | Weekly childcare hours  This will have no effect when CcUs is set to variable. If CcUse is set to Fixed, this will allocate each child with the same weekly care hours. |
| SpsIncSplit | Method of determining spouse’s income.  Fixed = Spouse’s income will be a fixed number specified by SpsInc.  Variable = Spouse’s income will be a multiple of the reference person’s income, specified by SpsInc |
| SpsInc | Spouse’s income  If SpsIncSplit = Fixed, the spouse’s income in dollars. If SpsIncSplit = Variable, the spouse’s income relative to the reference person’s income (e.g. 0.3 would result in the spouse have 30 per cent of the reference person’s income, at each income level simulated). |
| Start | Start value for reference person’s income. |
| Stop | End value for reference person’s income. Ensure that this is set to the value you want plus the increment i.e if you want to set the end value at $100,000 and your desired increment is $10, set this amount to $100,010. |
| Increment | Step value between each income point for calculating EMTRs. Note that having small increments can result in very large output files. |

**Figure 3.10: Specification of RunCompare, IncSource and RunEMTR in the Cameo Code.**

The RateTypeList and CameoList definitions specify the variables required for the cameo output (see the Variable Register for a complete list of available variables).



**Figure 3.11: Specification of the RateTypeList and CameoList in the Cameo Code.**

Finally, run the *Cameo Code*.

**Step 5: View and check Cameo Output**

Check the timestamp on the CameoOutput.xls spreadsheet to make sure new data has been written to it. Then open it to see the results of the cameos for the specified families. If you have any issues with the tabs displaying incorrectly, you may need to delete the CameoOutput.xls spreadsheet and then re-run the *Cameo Code*.

If additional diagnostic information is required, the **CAPITA\_Outfile** / **CAPITA\_Compare** can be consulted, or additional variables can be added to the cameo list in Step 3 and the cameos re-run.

### *Important Notes on CAPITA Cameos*

Note the following cautions regarding the use of CAPITA cameos:

**Note 1: Underlying Assumptions**

The following assumptions should be noted:

* Both the reference person and spouse are assumed to be eligible for JobSeeker Payment, the Age Pension, Parenting Payment Single, Parenting Payment Partnered and Youth Allowance if they satisfy age and parenting criteria. Therefore, only their level of private income will determine whether the individual is entitled to a positive amount of the payment.
* Children aged between 5 and 11 inclusive are assumed to be in primary school, and children aged between 12 and 19 are assumed to be in secondary school (provided they are dependent students). For more information, see the Guides to the *Dependants1* and *Dependants2* modules in Section 5.2.
* The minimum age for Age Pension eligibility is assumed to be 65.5 years old in 2017-18 and 2018-19. This is hardcoded in the *Cameo code*, because the *Cameo code* does not have access to the parameters datasets before it calls RunCAPITA. The age pension parameter is needed to enable assignment of private income of people of age pension age to superannuation income. Since the value is hardcoded, the age assumption will need to be updated if there is any change to the age eligibility for age pension.

**Note 2: Treatment of individuals eligible for both Parenting Payment Partnered and Youth Allowance**

From February 2015, all couples with dependent children under six years of age are eligible for Parenting Payment Partnered (PPP). The payment is made to one member of the couple, and the cameo code assigns the payment to the spouse. However, if the spouse is also a full-time student aged between 16 and 24, or a job seeker aged under 24, they will also be eligible for Youth Allowance (YA).

This policy change is treated differently in the cameo and distributional versions of CAPITA. In the distributional model the payments are assigned based on benefit receipt in the SIH, meaning that a spouse in this situation will be assigned to YA rather than PPP, as they should be receiving YA on the SIH.

In the cameo code, the policy modules determine eligibility for YA after they determine eligibility for all the other allowances. Therefore, allowing these individuals to be eligible for PPP means they are unable to qualify for YA (as the YA eligibility checks that they are not receiving any other DSS allowance).

Additional conditions have been added to the cameo code to ensure that PPP is only given to individuals who are not eligible for YA. While this makes them very slightly worse off at low income levels, it gives them the higher entitlement (YA) at all other income levels.

Since the cameo code does not read-in the policy parameters at the cameo basefile creation stage, the age parameters associated with Youth Allowance (Student) and Youth Allowance (Other) are hardcoded in the cameo code. Therefore, as was also the case for the age pension age parameter discussed above, if there are any policy changes to the age parameters associated with Youth Allowance they will need to be changed in the cameo code.

**Note 3: Removal of blank rows in IncomeList tabs**

When values are deleted from cells in Excel, SAS still recognises the cell (as blank) and so occasionally there are blank rows produced in the Cameo Output spreadsheet. To rectify this, select a large number of the rows in the IncomeList tab below the final income level desired, right click, and select Delete.

**Note 4: Recommended use of IncSource = SAS**

The IncSource can be set to the user’s preference however it is recommended to use IncSource = SAS for EMTR runs with a large range of income at small increments. This is because SAS has difficulty in importing the Cameo Input.xlsx and will run into an error if there are too many rows in the income sheets.

## Effective Marginal Tax Rates (EMTRs)

Effective Marginal Tax Rates refer to the proportion of each additional dollar earned that is removed due to either tax or the withdrawal of transfer payments. EMTRs are useful from a policy perspective in terms of assessing potential incentive and workforce participation effects arising from the design and structure of the tax and transfer system. CAPITA uses the cameo code to produce EMTRs for hypothetical families/individuals. The EMTR code runs on the base world *RunCAPITA*, however, if a new policy world needs to be analysed, this can be reflected in a different *RunCAPITA*, with the EMTR code redirected to this.

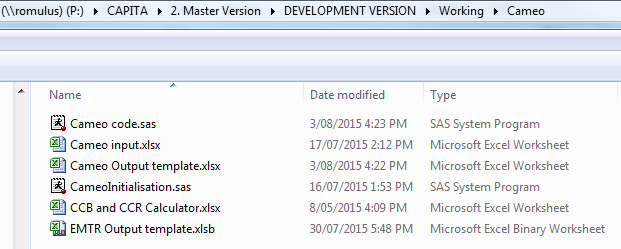
There are two types of EMTR charts that CAPITA can produce, the traditional EMTR chart (EMTR chart) and the Workforce Disincentive Rate chart (WDR chart). The EMTR chart uses a forward looking EMTR and can be set at any increment, whereas the WDR chart is set at daily increments and is backwards looking i.e. looks at how much additional private income is lost to taxation and net childcare costs from working an extra day.

Use the following steps to produce EMTR charts in CAPITA.

**Step 1: Specify the Capita directory, set up the Cameo Folder**

As discussed in Step 1 of the RunCAPITA steps above, first specify the CapitaDirectory.

As described above for the cameos, first make a copy of the entire Cameo folder and perform the work in this folder.



**Figure 3.12: Example of EMTR Working folder.**

**Step 2: Cameo input**

In *Cameo Input.xlsx* select the cameo/s you want to produce an EMTR chart for by putting a “Y” in the “selected” column.

EMTR runs use the same variables as Cameo runs (see step 3 of the Cameo analysis section above). The user can decide how to define the income variables. As discussed above, the IncSource toggle can be changed to ‘SAS’ to use SAS code to set the income. SAS is recommended as the income source for runs with large ranges of income at smaller increments.

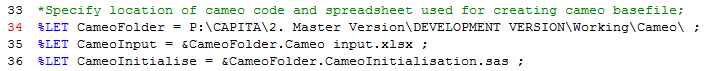
**Step 3: Define CAPITA directory**

Open and run DefineCAPITADirectory.

**Step 4: Setup and Run the Cameo code**

Open the *Cameo Code.sas* file in the Cameo folder.

You need to change the locations defined in the cameo code so that it reflects your new (copied) folder.

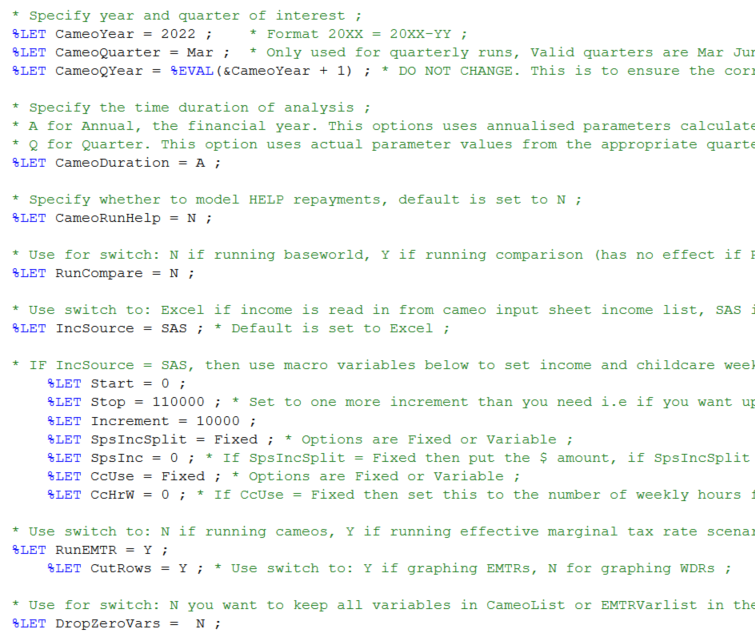


**Figure 3.14: Specification of the directories used in the EMTR code.**

Select the year or quarter you want to run as well as the time duration, annual or quarter.

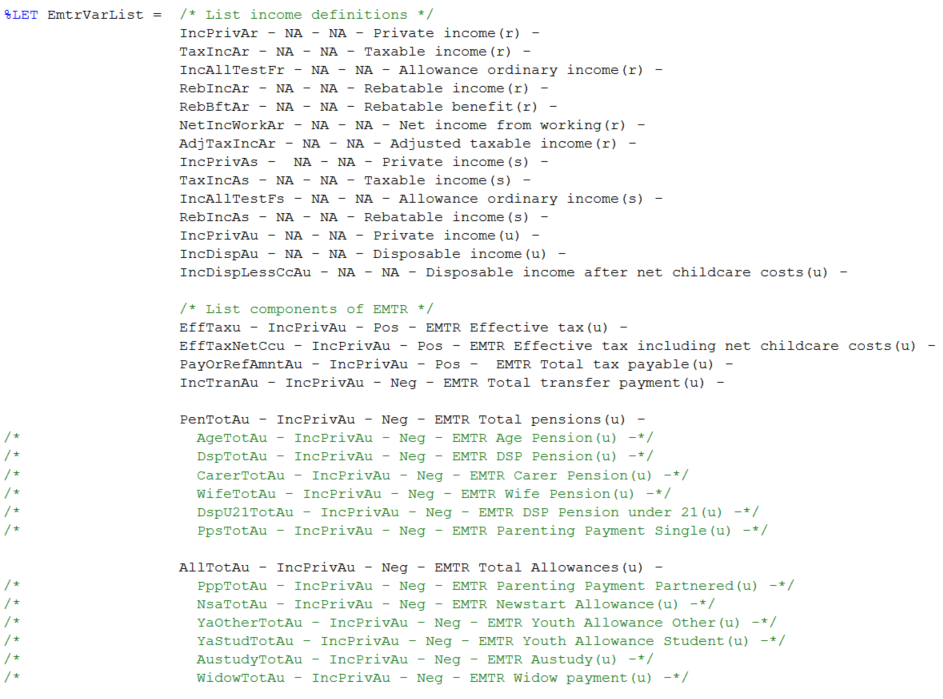
Set the RunEMTR switch to Y. Setting the DropZeroVars switch to Y will suppress output of any variables that are zero at all income levels for a given family.

Setting CutRows = Y will output a dataset with only the private income levels where there are changes in the EMTR. Setting CutRows = N will keep all private income levels in the final dataset, which will be exported to Excel in the final step.



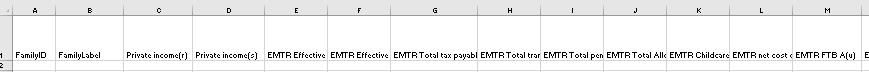
**Figure 3.15: Specification of the options used in the EMTR code.**

Like cameo runs, EMTR runs have a list of specified variables. These variables are specified in EmtrVarList. EmtrVarList is split into three groups: a list of income definitions, components of the EMTR and the variables used to calculate the EMTRs. Further information about the EmtrVarList is provided in Section 5 – Guide to CAPITA Policy Code.

****

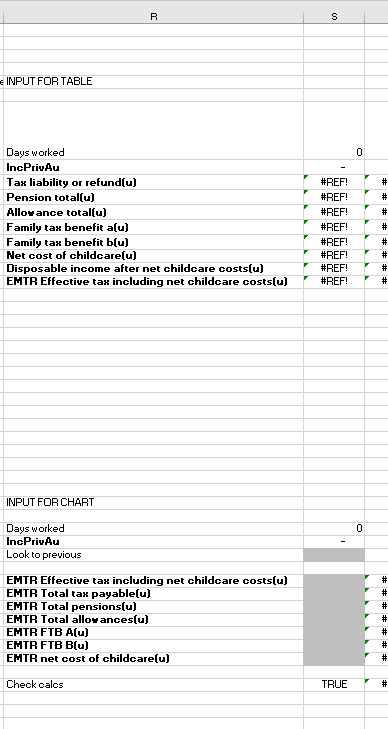
**Figure 3.16: Specification of variables used in the EMTR code.**

For EMTR charts, set CutRows = Y, DropZeroVars = N and only keep the components of the EMTR that are necessary for charting. The labels for the variables for charting are available in the *EMTR Output Template.xlsb*, in the EMTRchart tab in cells A1:AB1. The variables should also be kept in the same order in EmtrVarList for easy charting.



**Figure 3.17: Example variables in EMTRchart tab in EMTR output template**

For WDR charts, set CutRows = N, DropZeroVars can be set according to user preference. Any number of variables can be kept in EmtrVarList, however the variables in bold in the WDRchart tab in the *EMTR Output Template.xlsb* should be kept. The WDR chart, unlike the EMTR chart, is not sensitive to the ordering of variables.



**Figure 3.18: Labels for required variables in the WDRchart tab in EMTR output template**

The EMTR code must be outputted to a pre-formatted spreadsheet. The *EMTR Output Template.xlsb* spreadsheet in the EMTR folder contains these pre-formats. The EMTR code will output to a time stamped copy of this spreadsheet.

Finally, run the EMTR code.

**Step 5: Constructing charts**

***EMTR charts***

Open the EMTR Output spreadsheet (in the CameoFolder defined in Step 4), and go to any tab other than the tabs with SAS output (this is to ungroup the SAS output tabs).

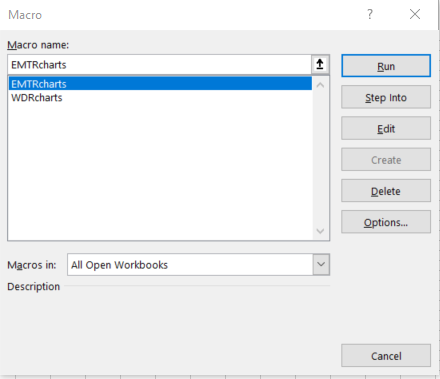


**Figure 3.16: Example of group tabs in EMTR output template.**

The EMTRchart tab is a template for creating the three EMTR charts. The three EMTR charts show:

* the overall EMTR as a result of aggregating all transfer payment taper rates and tax rates;
* the breakdown by tax and transfers ; and
* the breakdown by elements of the tax and transfers system.

To create these charts, click on the “View” tab > Macros > EMTRcharts> Run.

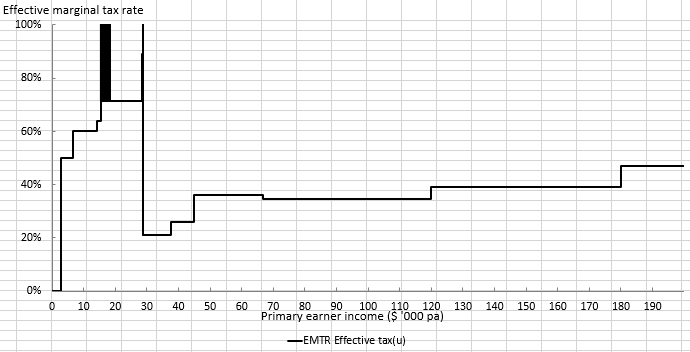


**Figure 3.17: How to run the macros for creating EMTR charts in Excel.**

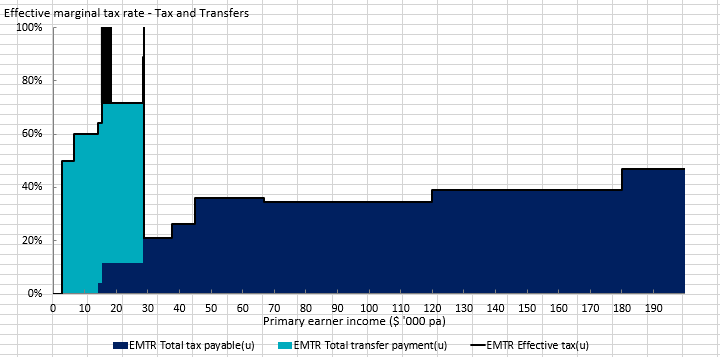
Running the macro will create a number of new sheets, equivalent to the number of sheets you have after the “SAS output>>” tab. Each new tab should have the names of the cameos plus EMTR and will contain the three EMTR charts. The data for each chart is hidden and can be changed using the “Select Data” option in the “Chart Design” tab.

****

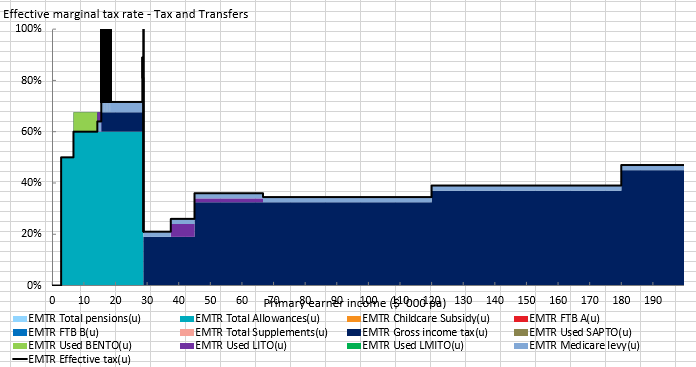
**Figure 3.18 Example of the tabs created after running EMTR macros**

****

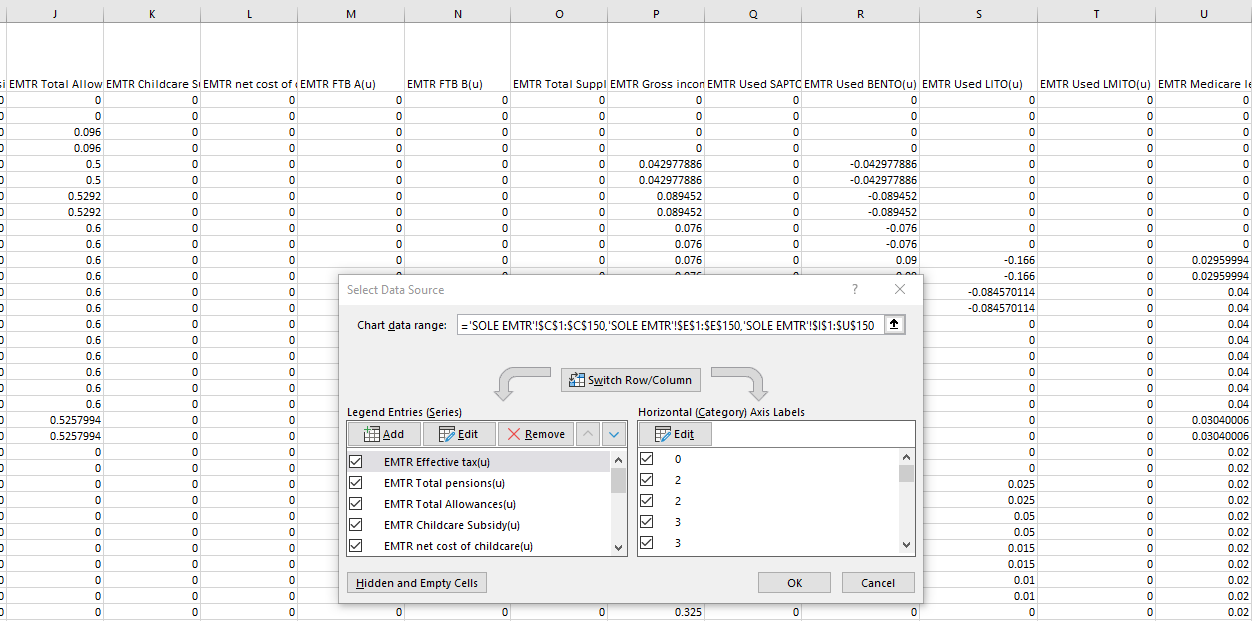
**Figure 3.19: Example of overall EMTR chart.**



**Figure 3.20: Example of EMTR chart broken down by tax and transfers.**



**Figure 3.21: Example of EMTR chart broken down by elements of tax and transfers.**



**Figure 3.22: How to ‘Select Data’ in Excel.**

**Important note about EMTR charts**

Some elements, namely **EMTR** **Used LITO** and **EMTR** **Used BENTO** are in effect ‘negative EMTRs.’ This is because these offsets give more back than they take away at certain income levels.

As the chart type is a **Stacked Area chart**, the negative elements will be stacked onto of the positive elements and will overlap.

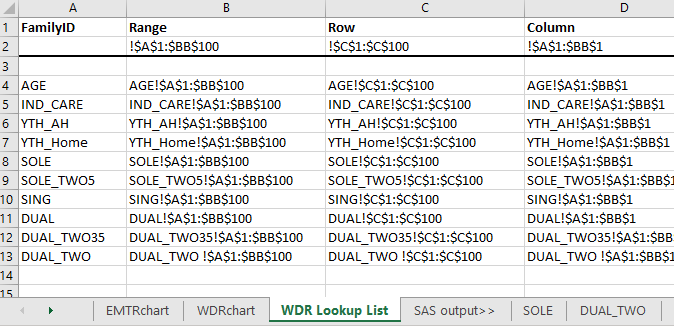
The choice of presentation of the negative EMTR elements is up to the person creating the chart and may depend on the purposes of the chart.

Finally, also note that the standard charts contained in the EMTR spreadsheet will only display EMTRs between 0 per cent and 100 per cent. There are particular income levels at which EMTR ‘spikes’ can occur, for example, due to sudden cut-offs in payment amounts or when transfer entitlements tapering to zero cause changes in the application of other offsets or thresholds. To see these spikes above 100 per cent, the vertical axis ranges of the standard charts can be adjusted accordingly.

***WDR charts***

Creating WDR charts is similar to EMTR charts. Once you have the output in Excel, the first you will need to update is the “WDR lookup list” tab. Make sure that the name of your cameo is on the list and that each of Range, Row and Column is updated. This sheet can hold up to 20 cameos.

Once you have done this, run the “WDRcharts” macro to create the new tabs for each cameo containing the WDR chart and table.

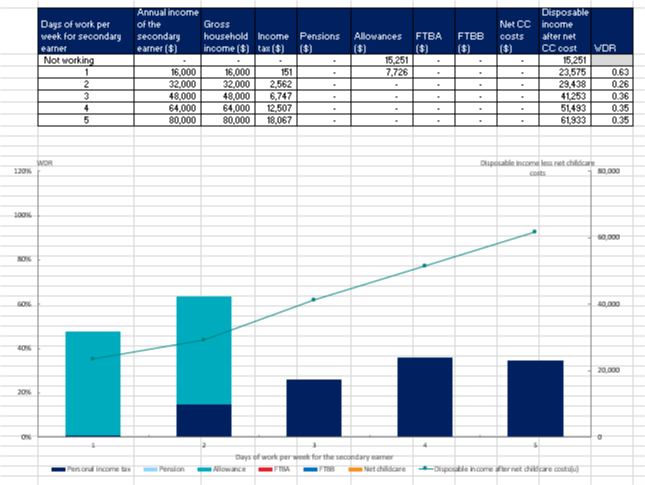


**Figure 3.24:Example of the values that need to be populated for the WDR charts.**

The “WDRcharts” macro will copy the “WDRchart” tab and update the Family label i.e. cell D1, to the name of the respective cameo. You will need to update the Primary earner full-time equivalent and Secondary earner full-time equivalent incomes to that of the cameo to populate the table and chart correctly.



**Figure 3.24:Example of the values that need to be populated for the WDR charts.**



**Figure 3.25: Example WDR chart and table.**

# Section 3.3 – Modelling Policy Changes

## Introduction

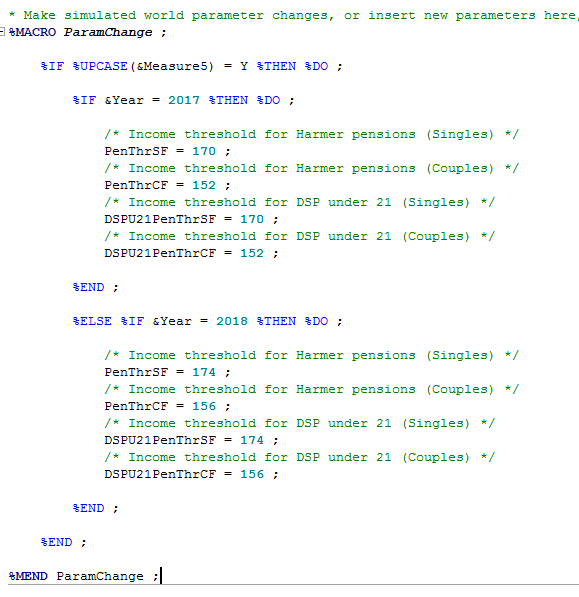
This section describes how to model policy changes in CAPITA. It also suggests a method for modelling a large suite of policy changes such as a Budget or MYEFO package of measures.

## Making parameter changes

The simplest policy changes to model are changes to payment rates and thresholds (or parameter changes). There are two different options for making parameter changes to the CAPITA simulation world. However, in situations where a change is made to tax rates 1 to 4 or the Medicare Levy for the base year separate steps will need to be taken. This is explained further below.

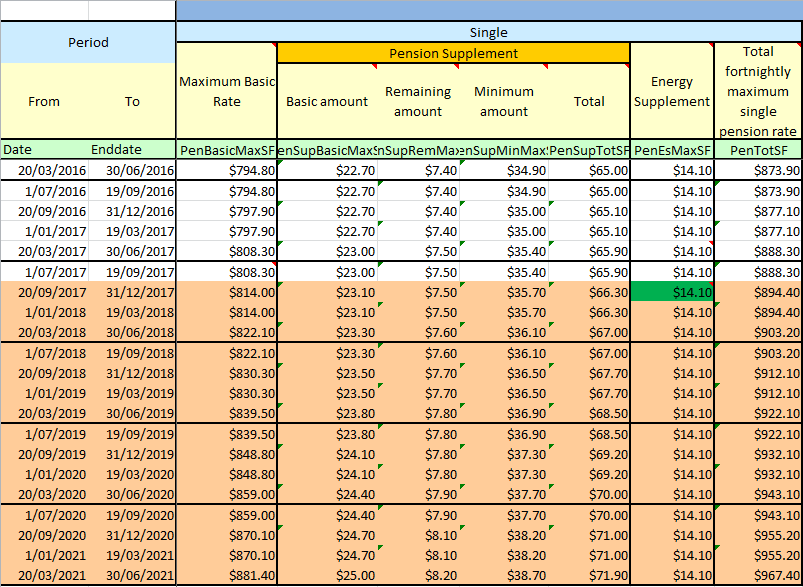
**Option 1: Code the parameter change in RunCAPITACompare**

The first option is to change the parameters change in the code, by overwriting individual parameter values. This can be performed in the **ParamChange** macro in *RunCAPITACompare*. The benefit of this approach is that the change is very transparent (since it is easy to see exactly which parameter values have been changed), however, the downside is that the parameter change will need to be coded for each possible year that the model could be run. An example parameter change performed in the **ParamChange** macro, which affects the 2017-18 and 2018-19 years only, is coded as:



**Figure 3.25: Using the ParamChange macro to code policy changes as changes to parameters.**

To incorporate the parameter changes into a simulation world version of the CPS. you need to make a copy of the CPS and place it in the Parameter (Sim) folder. Then incorporate the parameter change into this version of CPS. Next, use the *RunParameters.sas* program to generate the parameter datasets for the sim world in the Parameter (Sim) folder, noting that the ParamWkBk and AllParmDrive macro variables will first need to be updated to the location of the sim world CPS and the Parameter (Sim) folder respectively. Then, you need to create sim world basefiles with these sim world parameters and run *RunCAPITACompare* .

**Figure 3.26: The Common Parameter Spreadsheet (CPS).**

**Parameter changes to tax rates or Medicare Levy rate in the base year**

For changes to the tax rates and Medicare Levy rate in the base year, a new basefile will need to be created for the sim world with changes to these parameters. This is because tax rates 1 to 4 and the Medicare Levy exist as separate variables on the basefile for the base year and any change to these parameters using the above options will not be made. Option 1 and 2 can be used for any subsequent year and also for any other parameter change. If changes are to be made for multiple parameters, including the tax rates and Medicare levy for the base year, a combination of creating a new basefile and Option 1 or 2 will need to be applied.

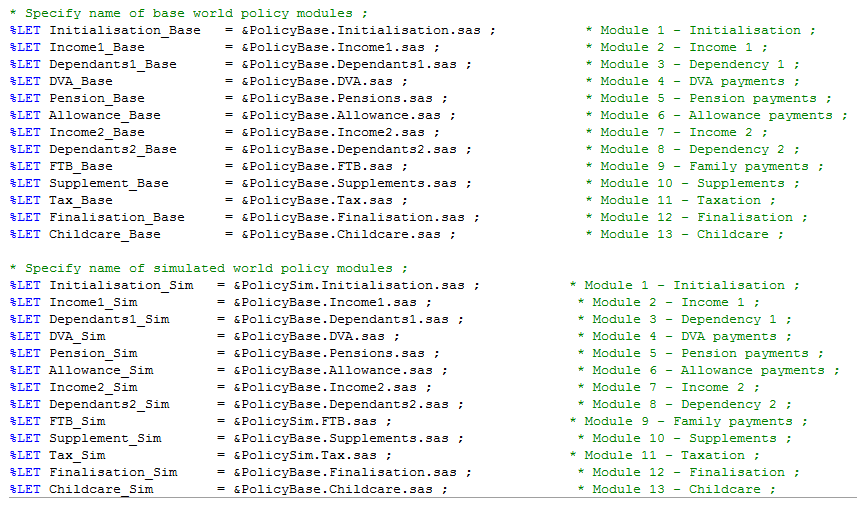
## Making policy changes

Structural policy changes, such as the introduction of new payments or redesign of existing payments, are more complex and need to be incorporated in the policy code. **Proposed policy changes should be modelled in the simulation world of CAPITA.** This is because the base world of CAPITA contains all current policy to date, and serves as the benchmark for comparisons with the simulated world. The base world should only be changed when new policies become official government policy (for example, at the conclusion of a Budget or MYEFO).

Note that the convention for the timing of structural policy changes[[7]](#footnote-7) in CAPITA which begin part way through a financial year is to code them to begin at the **beginning of the closest financial year**. For example, a redesign to a payment which takes effect from 1 January 2017 would be coded to begin in the 2017-18 financial year (i.e. as if the change was being made on 1 July 2017), whilst a structural change taking effect from 1 September 2016 would be coded to begin in the 2016-17 financial year (i.e. as if the change was being made on 1 July 2016).

A related issue is the appropriate way in which to model the slippages to the implementation dates of policy measures, i.e. policy measures which are still current Government policy but which cannot possibly be legislated in time for the intended start date to eventuate. CAPITA currently models these slippages to implementation dates when they are reflected in the forward estimates.[[8]](#footnote-8) This approach involves both advantages and disadvantages. Whilst it causes a deviation between official Government policy and the CAPITA policy code, it ensures that the CAPITA policy code is consistent with DSS customer number benchmarks, which do reflect the latest slippage dates in policy implementation. In addition, it ensures that the payment parameters in CAPITA remain consistent with the actual payment rates that exist in reality.

Structural policy changes are reflected as new code in the modules contained in the Policy (Sim) folder. For example, if there was a measure to introduce a new tax offset in the personal tax system, that offset would be incorporated into the *Tax.sas* module in the Policy (Sim) folder. If there was also a measure to restructure Family Tax Benefit from 1 July 2017, this would be incorporated in the *FTB.sas* module in the Policy (Sim) folder, with a date flag so that the change only takes effect from 1 July 2017 onwards. It is good practice to only point to modules in the Policy (Sim) folder when there have been changes to the modules in the simulated world – i.e. if there are no changes to particular modules between the base and simulated worlds, then the sim world directory for that module should be the base world directory. For example, the two changes made above would be likely to only involve changes to the *Tax.sas, FTB.sas* and *Initialisation.sas* modules:



**Figure 3.27: Specification of locations of the simulated world modules. The Sim directories are only specified for modules which are affected by the simulation world policy changes.**

## Using flags to activate and de-activate policy changes

During policy development, it may be useful to use flags at the top of the RunCAPITACompare code to enable particular policy changes to be switched on and off as desired. For example, these flags could appear as follows:

\* Measure 1 change description ;

%LET Measure1 = Y ;

\* Measure 2 change description ;

%LET Measure2 = Y ;

**Figure 3.28: Using macro variables as flags for easily activating or de-activating policy changes.**

Then, in the modules in the Policy (Sim) folder, the measures can be coded as follows:

%IF Measure1 = Y %THEN %DO ;

    \* Updated code to implement new measure ;

%END ;

%ELSE %DO ;

    \* Code from base world (i.e. existing code) ;

%END ;

**Figure 3.29: Coding the measures using DO loops.**

This means that the measures can be activated by setting them to Y or N at the top of *RunCAPITACompare* as required.

1. Wife pension ceased from 20 March 2020. Recipients moved onto Carer Payment or JobSeeker Payment after this date. [↑](#footnote-ref-1)
2. Known as NewStart prior to 20 March 2020. [↑](#footnote-ref-2)
3. Widow Allowance ceases from 1 January 2022. Recipients moved onto Age Pension after this date. [↑](#footnote-ref-3)
4. In most cases, you will not need to change any of the specifications in the *RunParameters* code. However, if you are using a separate copy of the CPS to make simulated policy changes which are to be reflected in the simulation world of CAPITA, the ParamWkBk should be directed to this separate copy of the CPS, and the AllParmDrive should be directed to the Parameter (Sim) folder, rather than the Parameter folder. This ensures that *RunParameters* generates the simulated parameter datasets into the Parameter (Sim) folder. [↑](#footnote-ref-4)
5. See the Variable Register, search for ‘flag’ or ‘type’. [↑](#footnote-ref-5)
6. The FSBrowse tool in SAS provides an alternate means of conducting unit-record analysis. The main benefit of FSBrowse is that shortcut keys can be used to quickly go through each observation record by record, with the variables being placed in the same location for each screenshot. From the FSBrowse window, using the Tools -> Options -> Keys menu, it is useful to change the F7 definition to left and the F8 definition to right, for use in scrolling through the observations. Also, Alt+S+W brings up a window for adding ‘where’ conditions, and Alt+S+A adds ‘where also’ conditions. [↑](#footnote-ref-6)
7. Structural policy changes include the introduction of new policies or structural changes to existing policies which are more complex than simple parameter changes. Any policy change which can be reflected through simple parameter changes (such as indexation changes or changes to payment rates) should be reflected in the CPS as occurring at the actual date that they take effect (not the start of the closest financial year). [↑](#footnote-ref-7)
8. For simple parameter changes, CAPITA models the slippages of their implementation as they are changes to the CPS which are picked up in year average payment rates and thresholds. For structural policy changes however, annual runs of the model maintain the default to model the start date of policies that have slipped to the nearest financial year. Quarterly runs of CAPITA reflect the slippage of measures to the nearest quarter by including a date condition in the code. [↑](#footnote-ref-8)