A Retirement Planner Model User’s Guide

Introduction:

In this document we will describe the objectives for using ARetirementPlanner, some simple cases and how tos. For this there is a section on the models input specification as well as a section on the models output.

ARetirementPlanner is aimed at helping retirees to get the most out of what they have. This does not, for the most part, include how to optimize your action in getting to the point of retirement. That said, there are at least two legitimate uses that occur prior to your actual retirement:

1. Using the tool to assess whether you can retire and planning near term retirement
2. Using the tool for a married couple where they do not retire the same year

In these cases there is a period of time where not everyone (or no one) is retired.

To use this application you must first prepare an input file of type toml. The package include some example toml files including one to be used as a starting point. The input configuration defined in the toml file is used to set up the constraint model to represent your specific situation for the solver to produce a proposed plan of action. The plan of action consists of a list of withdrawals (and deposits) that optimize your use of your retirement funds. The model accounts for all your federal income and capital gains taxes, rules for handling tax deferred retirement accounts, social security and other incomes as well as the sale of major assets like a home.

One issue that is not fully handled by the model is company base retirement plans for non-retirees. Take for example a retiree with a spouse that is working and has a 401(k) plan with their employer. Both the employee and the employer may be adding to this account and as with IRA and Roth IRA accounts there are maximum contribution levels. For the most part, at least in the current version of this program, we model only thee IRA / Roth IRA and not the 401(k) maximum contributions. It is up to the user to convert this information into that which the model will use. This can most easily be done by simply using the current balance in the model Tax Deferred Retirement Account (TDRA). The more sophisticated user may choose to add their expected contributions up to the retirement date to the balance which may improve the results.

The current implementation, which will change as we understand how it should work best, applies the rate of return for each account to the given balance between the current year and the start year for the retirement plan (year of the first person to retire). Also, contributions for each account that occur during this pre-plan period will be added each year and will then also receive growth at the account’s rate of return. However, no checking is being applied in the pre-plan period for meeting IRS rules. This, of course is not ideal. It does however provide an implementation dependent work-a-round for the models lack of support for 401(k) contribution levels which are much higher than those of IRA and Roth IRA. So, for the time being, in the IRA section you can define a pre-plan period that has a contribution related to the 401(k). If such a contribution is larger than allowed for IRA/Roth IRA it must stop before the first person retires or the optimization model will complain.

Accounts are modeled as having a yearly balance, deposits and withdrawals. Each retiree can have a TDRA account and / or a Roth Retirement Account (RothRA). Additionally one After Tax Retirement Savings / Investment account (ATRSI) can be included. So a single retiree could have up to three accounts being modeled (one of each type) while a married filing jointly could have up to five accounts (one per retiree for the retirement accounts and one non-RA).

To model your accounts you would sum your balances for all your accounts that act as a traditional IRA into the TDRA account starting balance. Then do the same for all your Roth account types. And finally, sum all the accounts with no special tax treatment for retirement for the after tax account.

For the most part 401(k) should be included in the TDRA account but sometimes they include portions that were contributed with after tax money. In this case the after tax contributions should be included in the RothRA account. IRA account balances go to the TDRA account and Roth IRA balances go to the RothRA account.

Once you have a handle on your account balances for each of the account types, you can choose an inflation rate (defaults to 0) and an overall rate of return for investments (defaults to 6%), set your retirement type, joint, mseparate or single (defaults to joint) and enter your account information. The final information you must specify is your current age, age you plan to retire and through what age to create the plan. With this there is enough for your first run.

Optimize for Spending or PlusEstate: You can choose to optimize for a maximum spending amount during retirement or for the final estate (PlusEstate). This setting defaults to optimize for spending and will create a plan that starts at a value of X dollars a year and increasing this amount for inflation each subsequent year. This gives a good starting point to understand where you are in terms of your ability to retire. If the first year amount is too low then you’ll have to do some serious thinking before you retire. If on the other hand it’s more than you believe you need or want, you can set a maximum spending amount and let the remainder stay in your accounts to increase your estate.

Another approach when you want to maximize your estate and you know what you will need each year, is to optimize for the estate (PlusEstate) and set a minimum income amount. In this case the solver will attempt to provide the minimum amount you specify for the first year of the plan and that amount increased for inflation each subsequent year. The plan will maximize your estate subject to this income.

The iterative process of using the model: Experiment with different input to see how to get the best results. Changing the date of retirement or when to start social security will give different plans and you can choose the one you like best. Higher rate of inflation and lower rate of return will give a more conservative plan. And while the solver creates a plan for the entire retirement planning period it is not intended that you would actually use this one plan for the entirety of your retirement. Over the course of retirement inflation, rates of return, spending needs, tax laws… all change as do your needs. So, each year you should update your information in the model and rerun it. Try some new ‘what if’ runs and see what seems best. Once you settle on a plan with your given specification, follow that plan as much as it make sense to.

Get your feet wet: In this section we will describe a very basic application of ARetirementPlanner. The first thing that you will need to do is to choose an example input file and modify it to include your specific data. Married and want to do a joint plan, choose ARetirementPlannerJointStartHere.toml to start. Single then choose ARetirementPlannerSingleStartHere.toml to begin. Copy your choice to whatever you want to call your plan input file, say try.toml. This won’t give a very interesting result when run because almost everything is commented out. Give it a try with:

PS C:\plan> ARetirementPlanner.exe try.toml -vi -va -vt -vtb

The following table is always printed on success of the optimization. It gives an overall summary of the proposed, optimized, plan. Here we can see our retirees, their age and how much should be withdrawn from the TDRA account each year. In this particular case the amount withdrawn by itself is not enough to incur federal taxes. This means, as can be seen that the spendable amount for the year matches the withdrawal amount. Starting at age 70 the Require Minimum Distribution (RMD) kicks in but is less than the amount to be withdrawn so it has no effect on the outcome. TDRA is the only account with activity, no other income, social security, expense or taxes so these all remain zero.

It helps to understand how these values are related within a year. Spendable equals the sum of the amount withdrawn from the accounts minus the amounts deposited, plus other income and social security and finally minus expenses and total federal tax. That is spendable is your after tax money minus expenses. Now in this light, expenses are meant to be major or unusual expenses rather than your normal expenses that you will have over the course of retirement. They are a way of telling the model that an extra amount is needed in certain years.

One way to think about these numbers or to compare them with your situation before retirement is to compare your pre-retirement annual income to the sum of the plan total Federal tax and spendable amounts for the first few years. Beyond this, inflation will overly distort the numbers for comparison. Setting inflation to zero could be used to compare farther out but that is a longer discussion than I want to take on here. Note that this ignores expenses and this is ok if they are not every year events.

The main reason to give a spendable amount rather than an annual income amount is that the annual income product by summing total federal tax and spendable will likely jump around a lot as compared to the spendable amount. This is because the optimization is selecting the best ways for you to consume your funds and this by itself results in very different tax burdens as the source of funds for the each year changes.

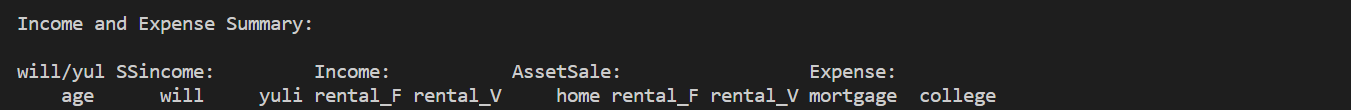
All the numbers are thousands (000) unless the -1k switch is given on the command line.



The –vi switch given on the command line causes the income and expense summary to be printed as below. Not much to see from try.toml sense it does not define [SocialSecurity], [income], [asset] nor [expense] sections. If these were defined additional columns would be displayed in sections as in the snippet based on a different toml file after our try.toml snippet.



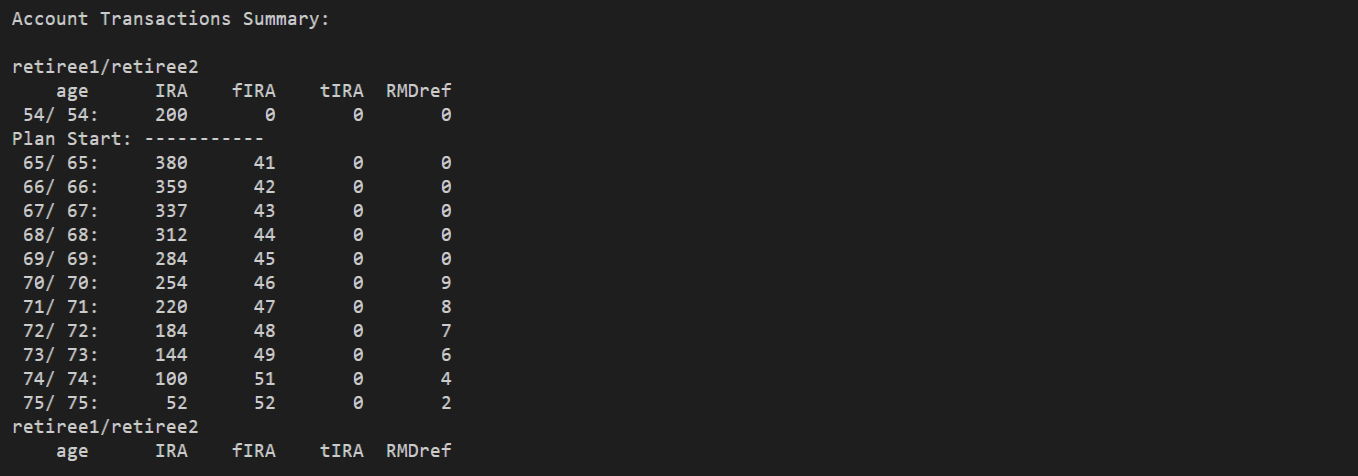
The SSincome section would come first with each column labeled with the retiree id key if defined or ‘SS.’ Next would come the Income section with each column labeled with the income id key. Then the AssetSales section with columns labeled with the asset id key and finally the Expense section with columns labeled with the expense id keys. Note that, as we will see below, the id key value is the text string after the ‘.’ in the toml file for each of these sections. An example for [income] is [income.rental1] where ‘rental1’ is the id key for one particular [income] section.



The –va switch given on the command line causes the account transaction summary to be printed as here below. Only one account is documented because only one account is specified in try.toml. In addition to the withdrawal (fIRA), deposit (tIRA) and RMD reference (RDMref) this summary includes the balance for the account (IRA). All modeled accounts (those defined in the input file) will be summarized here.

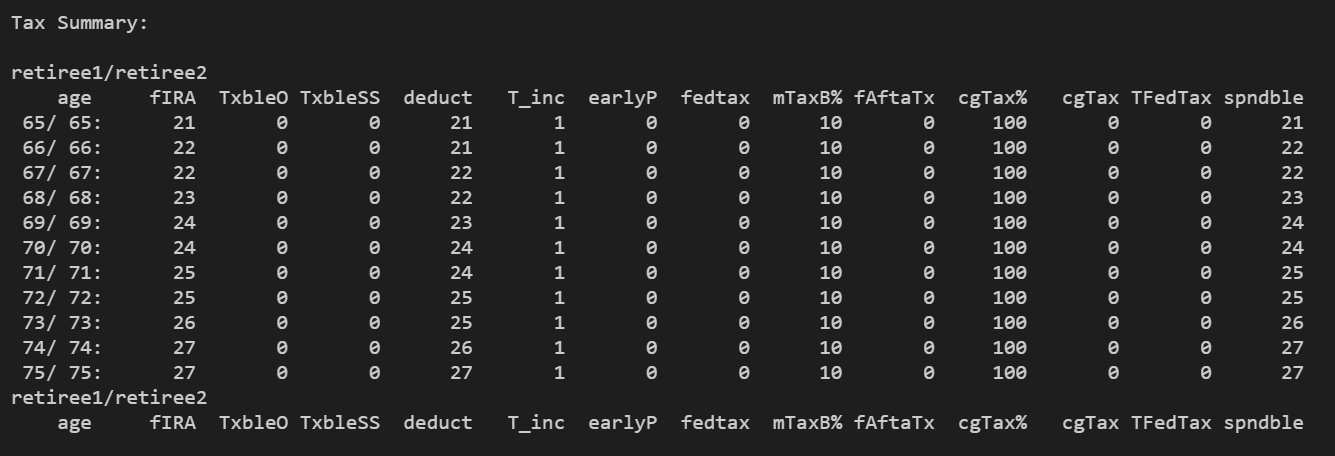
The first data line is for your current year which may be prior to the plan start. It lists the original balance for each account and any contributions specified. The other fields remain zero.

As you can see in the IRA column, the balance jumps from 200K at age 54 to 380K at age 65, plan start. This jump is 200K times the rate of return (6%) to the 11th power. That is the increase is based only on the rate of return for the account over the 11 year period. There are no contributions. They would be specified in the tIRA column.



The tax summary is displayed if the –vt switch is given on the command line. Here we see the withdrawals from the TDRA account as they are taxed as ordinary income. Also the taxable portions of other income (TxbleO) and 85% of social security (TxbleSS). This is followed by the allowed deduction and exemptions (deduct) and the Total taxable income (T\_inc). Early withdrawal penalties (earlyP) comes next followed by the federal tax amount (fedtax) and the marginal tax rate (mTaxB%). Next comes the withdrawals from ATRSI (fAftaTx) the fraction of these withdrawals that are not from basis, that is the gains on the investment, (cgTax%) and the capital gains tax itself (cgTax). Finally the total federal tax, fedtax plus cgTax, is given (TFedTax) and the amount of spendable funds for the year (spndble).

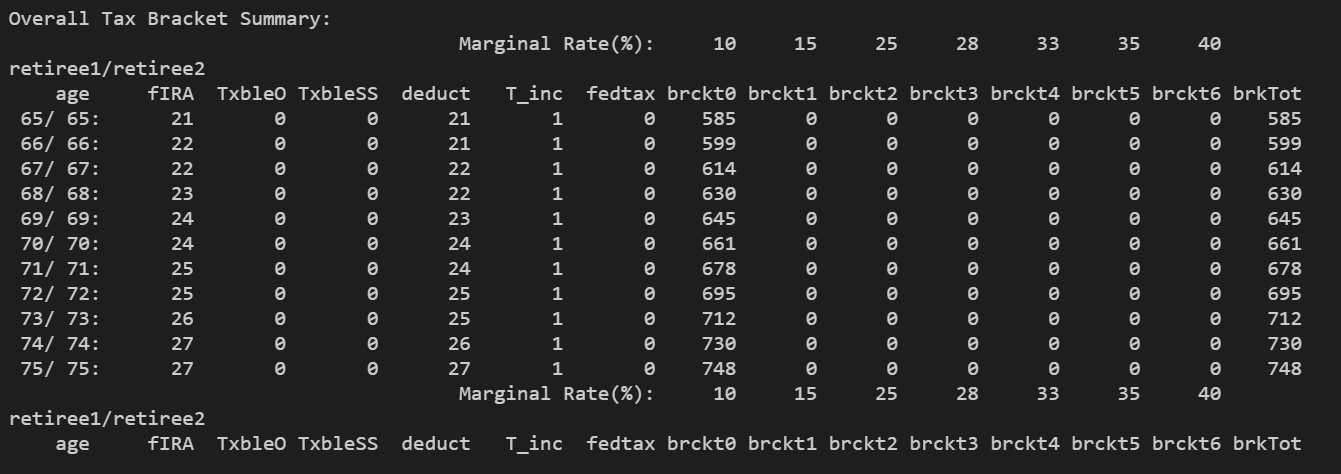
If the early withdrawal penalty includes a penalty for early withdrawal from a Roth account the value will have a ‘\*’ attached. In the current implementation Roth early withdrawal penalties are assigned more in the model than in life. It should rarely happen but when it does this indication will allow you to check to see if it is significant to your plan.

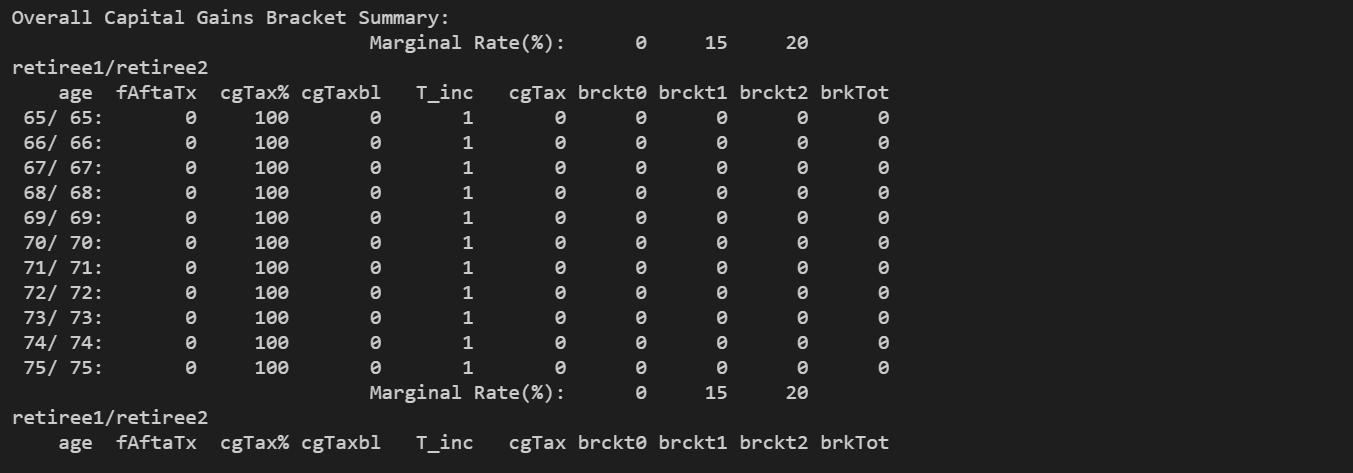


As we saw in the account activity summary it helps to understand how these columns are related. There are three basic subsections: income tax, capital gains tax and and summary. For income tax withdrawal and deposit (fIRA, tIRA), the taxable portion of other income and social security (TxbleO, TxbleSS) and deduction and exemptions (deduct) are all combined to calculate the total taxable income (T\_inc). It is this amount that is used to determine the marginal tax bracket, federal income tax amount (mTaxB%, fedtax). The second section, capital gains, uses withdrawals and deposits from the after tax investment account times the non-basis fraction of withdrawals to determine the portion (profits) to be taxed ((fAftaTx – tAftaTx)\*cgTax%) to determine the capital gains tax (cgTax). Finally, in the final section these results are brought together to define the total federal tax (TFedTax = earlyP+fedtax+cgTax) with spendable given for reference.

The next two tables are more technical. They are displayed whenever –tb is given on the command line. This represents the tax bracket details for ordinary income (first table) and capital gains (second table). In these tables the new columns are the brackets. Above and below each of these marginal rate columns in the marginal rate itself, the heading (brcktN) where N is a tax bracket from 0 on up to 6 for ordinary income taxes and 0 to 2 for capital gains brackets. The final column is to total of all the funds in the previous column’s brackets for the year.

An exception to the rule for all numbers to be in thousands is that the bracket data is for every dollar, that is, not in the thousands.

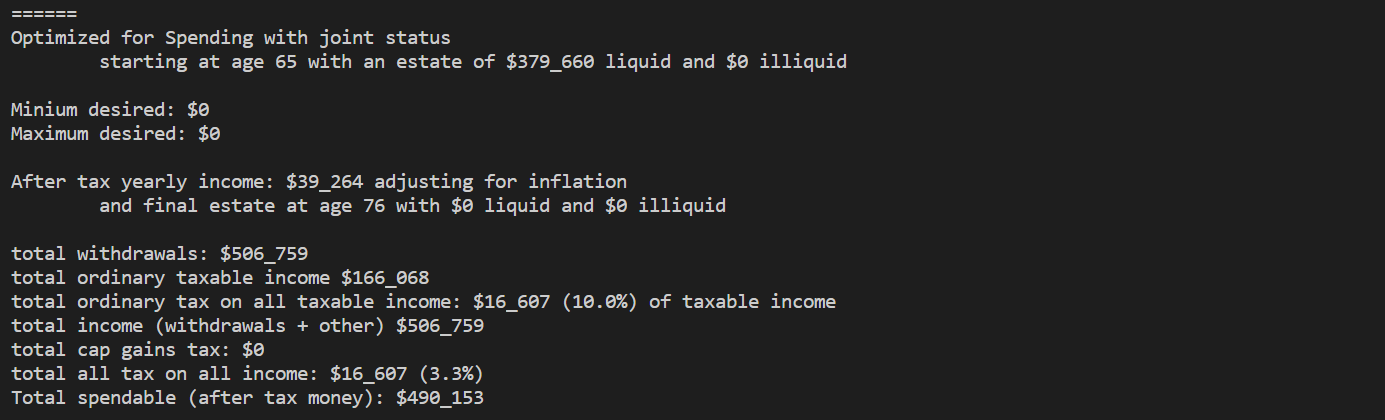




The tax for each year is calculated by summing the amount in each bracket times the bracket marginal rate.

The final output is always displayed with the following data. The object function being optimized: Spending or PlusEstate. The plan start estate liquid and illiquid value. Any minimum or maximum spendable amounts desired. The after tax yearly income derived from the optimization plan. This is the first year spendable amount and later years will be adjusted for inflation. This is followed by the plan end estate liquid and illiquid value.

After this some information on the overall plan totals are given to provide a more complete idea of the outcome. This includes total withdrawals from all accounts, total taxable income, total income tax and its percentage of all ordinary taxable income, total income, capital gains tax and then all taxes on all income and its percentage. Finally the total amount of after tax spendable money over the plan.



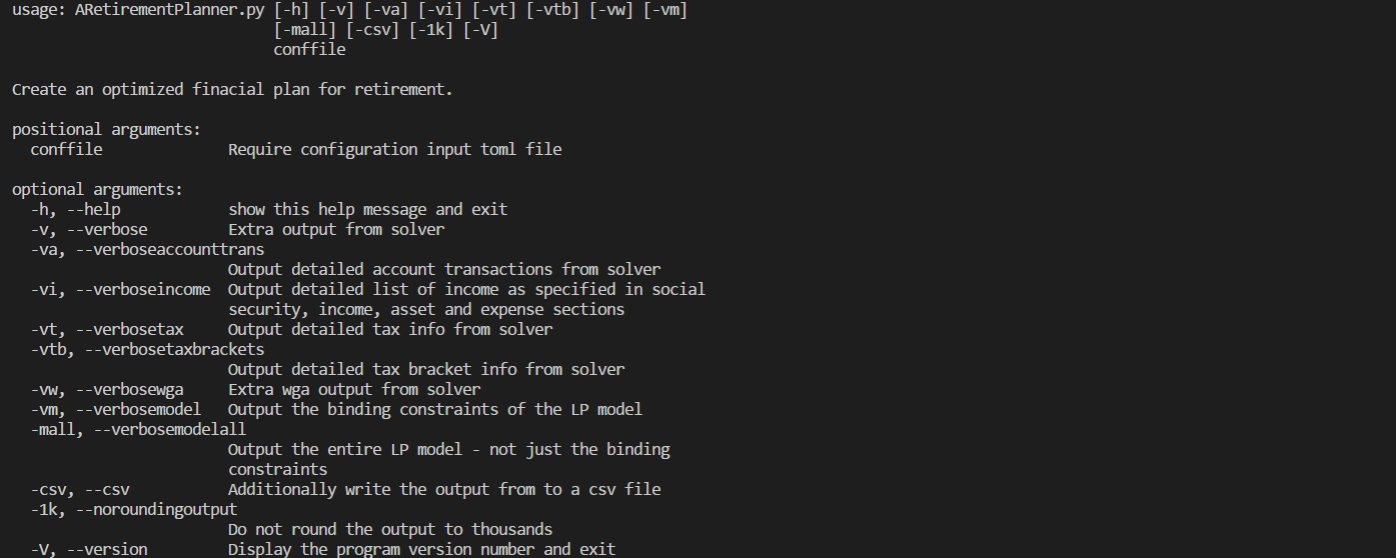
Review the toml file to get a feeling for what is in it. Most options are commented out to simplify your modifications. This way you can change and uncomment only those fields you choose and ignore all the other lines. Once ready modify the file to include your own information. Doing this in small steps and running the optimization at each step will help you understand the impact of your changes and ensure the input file syntax remains correct.

Example input files can be reviewed and run to get a better understanding of the inputs and there results. A few example files are included:

* ARetirementPlannerJointExample.toml
* ARetirementPlannerSingleExample.toml

Run the Example file versions to see how they can impact the results.

Invoking the model: All the switches for ARetirementPlanner.exe are display by invoking it with the help (–h) switch. The optimizer will print out the following information:



Model input specification:

The basis for the model input specification is the toml specification[[1]](#footnote-1) with its library. The basic format is sections of information ‘[‘ section name ‘]’ where section name can be a category followed by a ‘.’ and a descriptive name. ‘#’ to the end of the line represents a comment. The global section has no section name header. The input information is represented by an assignment.

Let us step through the ‘StartHere’ toml file to understand the input file components.

General model information (global section):

# This input file is meant as a starter for a joint plan

# most fields are options and have reasonable defaults

#retirement\_type = 'joint' # defaults to joint, could also be single or mseparate(married filing separately)

returns = 6 # return rate of investments, defaults to 6%

inflation = 2.5 # yearly inflation rate, defaults to 0%

# what to optimize for? 'Spending' or spending 'PlusEstate', defaults to ‘Spending’

#maximize = "PlusEstate"

Retiree Section Retiree is specified. If only one retiree is given the final portion of iam.final can be dropped. However it must match with the final portion of TDRA and RothRA accounts. That is if this final portion is dropped here it must also be dropped for the accounts. ([iam] mates with [IRA] and [iam.joe] mates with [IRA.joe]):

[iam.retiree1] # iam (for each) is required in some joint cases (".xxx" is used to match accounts IRA/roth)

primary = true # retiree to have age listed first in the output (must choose one)

age = 54 # your current age

retire = 65 # age you plan to retire

through = 75 # age you want to plan through

A second retiree means that at least one must have the final portion of iam.final specified. It is possible to only specify one retiree for a joint plan if all the accounts are owned by the specified retiree.

[iam.retiree2] # iam (for each) is required in some joint cases (".xxx" use to match accounts IRA/roth)

age = 54 # your current age

retire = 65 # age you plan to retire

through = 75 # age you want to plan through

The social security section is optional. If given its final portion as in SocialSecurity.final must match the final portions given or not in the retiree section.

#[SocialSecurity.retiree1]

#FRA = 67 # your full retirement age (FRA) according to the IRS

#amount = 20\_000 # estimated yearly amount at Full Retirement Age (FRA); Assumes inflation, 85% taxed

#age = "70-" # period you expect to receive SS ("68-" indicates start at 68 and continue)

#[SocialSecurity.retiree2]

#FRA = 67 # your full retirement age (FRA) according to the IRS

#amount = -1 # -1 for default spousal benefit amount, amount at Full Retirement Age (FRA); Assumes inflation, 85% taxed

#age = "67-" # period you expect to receive SS ("68-" indicates start at 68 and continue)

Annuities, reverse mortgages and other types of income can be specified in an income section. These sections are named as income.final where the final portion is some name meaningful to you for the type of income it is. It includes an annual amount, period you will receive it, whether it should be inflation adjusted and whether it is taxable.

#[income.taxfreeNoneInflationAdjustedAnuity]

#amount = 3000 # yearly amount

#age = "65-70" # period you expect to receive it

#inflation = false # not inflation adjusted

#tax = false # not federally taxable

#[income.InflationAdjustedAnuity]

#amount = 3000 # yearly amount

#age = "65-70" # period you expect to receive it

#inflation = true # inflation adjusted

#tax = true # federally taxable

Reverse mortgages take many forms with one of the simplest paying a constant amount each month for the rest of your life. This form and others can be represented in the income section. For example the simple form can be represented as follows:

[income.reversemortgage]

amount = 12000 # yearly amount, 1000/mo

age = '70-' # period to receive payments

inflation = false # payment is not inflation adjusted

tax = false # payment/loan is not taxable

#[income.rental]

#amount = 5\_000 # yearly amount

#age = "67-" # period you expect to receive it

#inflation = true # inflation adjusted

#tax = true # federally taxable

Major assets like your home or other real estate can have a significant impact on your retirement. If you plan to sell these assets during your retirement you can include there return in your plan with the [asset] section as below. This section provides the information needed for the optimizer to determine the amount to add to your after tax account and the amount that is taxable. If the asset is your primary residence the IRS exempts the first $250K ($500K joint) of capital gains from taxes.

#[asset.home]

#value = 550\_000 # current value of the asset

#costAndImprovements = 300\_000 # purchase price plus improvement cost

#ageToSell = 72 # age at which to sell the asset

#owedAtAgeToSell = 100\_000 # amount owed at time of sell (ageToSell)

#primaryResidence = true # Primary residence gets tax break

#rate = 4 # avg rate of return (defaults to global rate)

#[asset.rental]

#value = 250\_000 # current value of the asset

#costAndImprovements = 150\_000 # purchase price plus improvement cost

#ageToSell = 72 # age at which to sell the asset

#owedAtAgeToSell = 100\_000 # amount owed at time of sell (ageToSell)

#primaryResidence = false # Primary residence gets tax break

#rate = 4 # avg rate of return (defaults to global rate)

Desired income and max income should only be used with the appropriate ‘maximize’ case

#[desired.income] # used when maximize = "PlusEstate"

#amount = 45\_000 # retirement first year income

#[max.income] # used when maximize = "Spendable" (default)

#amount = 100\_000 # retirement first year income

Account sections for TDRA, RothRA and ATRSI all contain a bal and rate options for the account balance and account specific rate or return. If the rate is not given the global rate will be used. The final portion of the account.final section name must match with the final portion of the retiree section for the account owner. ATRSI accounts have an additional field ‘basis’ for the amount of money contributed to the account. This is after tax money and will not be taxed on withdrawal.

# pre-tax IRA accounts (TDRA)

[IRA.retiree1]

bal = 200\_000 # current balance

#rate = 7.25 # defaults to global rate set above

#contrib = 0 # Annual contribution you will make for period (below)

#inflation = false # Will the contribution rise with inflation?

#period = '56-60' # period you will be making the contributions

#[IRA.retiree2]

#bal = 100\_000 # current balance

#rate = 7.25 # defaults to global rate set above

#contrib = 0 # Annual contribution you will make for period (below)

#inflation = false # Will the contribution rise with inflation?

#period = '56-60' # period you will be making the contributions

# roth IRA accounts (RothRA)

#[roth.retiree1]

#bal = 20\_000 # current balance

#rate = 7.25 # defaults to global rate set above

#contrib = 0 # Annual contribution you will make for period (below)

#inflation = false # Will the contribution rise with inflation?

#period = '56-60' # period you will be making the contributions

#[roth.retiree2]

#bal = 20\_000 # current balance

#rate = 7.25 # defaults to global rate set above

#contrib = 0 # Anual contribution you will make for period (below)

#inflation = false # Will the contribution rise with inflation?

#period = '56-60' # period you will be making the contributions

# after tax savings accounts (ATRSI)

#[aftertax]

#bal = 100\_000 # current balance

#basis = 50\_000 # Contributions to total, for capital gains tax

#rate = 7.25 # defaults to global rate set above

#contrib = 0 # Anual contribution you will make for period (below)

#inflation = false # Will the contribution rise with inflation?

#period = '56-60' # period you will be making the contributions

1. The toml specification is archived at: <https://github.com/toml-lang/toml> [↑](#footnote-ref-1)