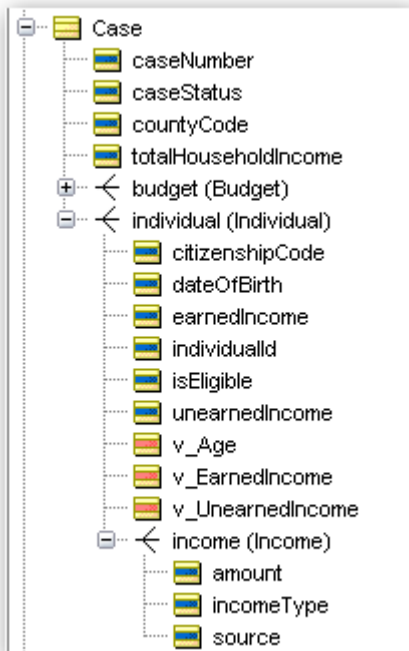# Automatic Generation of Test Data with Associations

The first rule model showed how you can automatically generate all combinations of test data for a single business object by making use of the power of the scope section.

Now let's consider how to do this when there are associations involved.

Suppose you want to generate instances of this vocabulary:
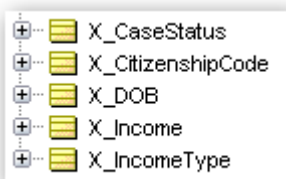

This can be done in three steps:

1. First generate all possible cases
2. Then add all possible individuals and budgets to each case
3. Then add all possible incomes to each individual

If you need to limit the individuals, or incomes being added then you can do it with additional rules.

Just as in the simple example we'll need ancillary business objects to hold the values:

And then the rule sheets will look like this:

Create Cases

PossibleValues	CreateCases	AddIndividuals	AddIncome

Scope

	Term	Alias
1	X_CaseStatus	case

Preconditions/Filters

Rules (non-conditional)

N.1	Case.new[caseStatus=case.value]
N.2	

Add Individuals

PossibleValues	CreateCases	AddIndividuals	AddIncome	Calc_Incon

Scope

	Term	Alias
1	X_CaseStatus	case
2	X_CitizenshipCode	citCode
3	X_DOB	dob
4	Case	cases
5	cases.individual	individuals
6		

Preconditions/Filters

Rules (non-conditional)

N.1	individuals+=Individual.new[citizenshipCode = citCode.value,dateOfBirth = dob.value]
N.2	

Notice the use of the += to associate the newly created instance with a case (every case in fact). If you need to limit the individuals, or incomes being added then you can do it by moving the non conditional rule down into the decision table and adding appropriate conditions (perhaps use a random number generator?)

Add Incomes

	PossibleValues	CreateCases	AddIndividuals	AddIncome	

Scope

	Term	Alias
1	X_Income	income
2	X_IncomeType	incType
3	Case	cases
4	cases.individual	individuals
5	individuals.income	incomes

Preconditions/Filters

Rules (non-conditional)

N.1	incomes+=Income.new[incomeType = incType.value,amount=income.value]
N.2	
N.3	

When you run this you will get a result like this:

```
iCIS
  Case [1]
    caseStatus [O]
    individual [7]
        citizenshipCode [01]
        dateOfBirth [01/01/1999]
        isEligible [true]
        v_Age [11]
        v_EarnedIncome [1500]
        v_UnearnedIncome [1500]
        income [73]
            amount [100]
            incomeType [E]
        income [79]
        income [85]
        income [91]
        income [97]
        income [103]
        income [109]
        income [115]
        income [121]
        income [127]
        income [133]
        income [139]
    individual [8]
    individual [9]
    individual [10]
    individual [11]
    individual [12]
  Case [2]
```