

# OPENRULES®

**Open Source Business  
Decision Management System**  
Release 6.4.0

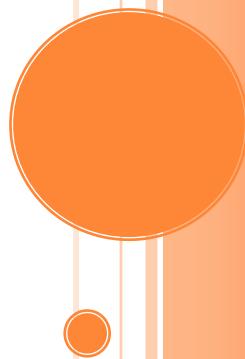
## **Preparing a Tax Return Using Executable Decisions (Decision1040EZ)**

# **Tutorial**

**OpenRules, Inc.**

[www.openrules.com](http://www.openrules.com)

May-2016



## Table of Content

Introduction .....	3
Business Case .....	3
Decision “Decision1040EZ”.....	4
Starting with Decision.....	4
Defining Decision Tables .....	7
Validation Rules – Simple Decision Tables .....	7
Using Formulas inside Decision Tables.....	9
When Conditions and Conclusions Share Decision Variables.....	10
More Complex Decision Tables .....	11
Defining a Business Glossary.....	14
Defining Test Data in Excel .....	17
Decision Execution.....	19
Appendix. Form 1040EZ .....	25

## INTRODUCTION

In this tutorial we explain how to implement a rules-based application that calculates tax returns using the notorious US tax form 1040EZ. This tutorial is written in the form of a dialog between two people:

- **User** – a person who wants to learn how to write OpenRules®-based applications
- **OR** – a representative of OpenRules® technical support who explains how to do it.

A user is not expected to be a software developer but rather a business analyst who is familiar with Excel and who has already looked at the OpenRules document “Getting Started” that includes several basic examples.

This tutorial explains how to use OpenRules® Decisioning approach to implement tax calculation logic for the form 1040EZ. It does not include any GUI and can be incorporated in any application as a pure decision service.

There is also another tutorial that explains how to use OpenRules® Dialog (ORD) to create a web-based dialog during which only necessary questions are being asked, all other answers are automatically calculated, and a ready to go tax return in the PDF format is being produced – see <http://openrules.com/pdf/Tutorial.Dialog1040EZ.pdf>.

## BUSINESS CASE

**OR.** We will prepare the notorious US tax form 1040EZ used by many US taxpayers to report on their revenue and expenses and calculate whether the taxpayer owes additional money or can expect a refund.

**User.** I did not use it myself but I know how frustrated some of my colleagues were when they tried to fill out this form themselves.

**OR.** There are plenty of web-based applications that help people to do it. While the form 1040EZ is supposed to be used only in simple cases, the business logic behind this form is not so simple, especially when a taxpayer was shown as a dependent by his/her parents or somebody else. The actual form has only 2 pages and in many cases only the first page is needed. The second page has to be filled only when a taxpayer is shown as a dependent. We also have to keep in mind the eligibility criteria that specify when to use this form.

**User.** Where can we get the form?

**OR.** Look at the [Appendix](#) below. We downloaded this form from the IRS website and used this form to demonstrate how to use OpenRules®. For your convenience, the form 1040EZ is presented on the next two pages. We use the tax form for 2003 as the example for our

application – see the Appendix below. You may download the latest 1040EZ form from <http://www.irs.gov/pub/irs-pdf/f1040ez.pdf>. The instructions can be found at <http://www.irs.gov/pub/irs-pdf/i1040ez.pdf>. As you can see, the form is self-explanatory and people should be able to fill it out without external help.

**User.** I noticed that the official 1040EZ instructions have more than 35 pages; probably it is not so easy to cover all possible situations. However, our real OpenRules®-based application is also not going to be easy, so I'd appreciate it if you would explain how to handle not just simple cases but also much more complex business scenarios.

**OR.** OK, we plan to build a simple decision service Decision1040EZ that implements only the business logic of the form 1040EZ.

## DECISION “DECISION1040EZ”

**OR.** This section will describe a methodological approach and consider different implementation options for the above tax form 1040EZ. A ready to go application is a part of the OpenRules® installation – you can find it in the sample project “Decision1040EZ”.

## STARTING WITH DECISION

**OR.** We will apply a top-down development approach. It means that we will start with the definition of a Decision and not with rules or data. Let's call our decision “DetermineTaxReturn”.

**User.** Sounds good to me.

**OR.** Let's look at the first page of the Form [1040EZ](#). Can you list all decisions that we should make?

**User.** I guess they are defined in the lines of the form. After filling out general information about a taxpayer we should do the following calculations:

- Calculate Adjusted Gross Income (Line 4 by adding the content of Lines 1, 2, and 3)
- Calculate Dependent Amount (Line 5 with potentially complex logic described on the second page)
- Calculate Taxable Income (Line 6)
- Calculate Total Payments (Line 9)
- Define Tax (Line 10 using the standard Tax Table)
- Calculate Refund or Amount You Owe

**OR.** Good. So, let's create an Excel file “Decision.xls” with the following table of the type “Decision”:

<b>Decision DecisionCalculateTaxReturn</b>	
<b>Decisions</b>	<b>Execute Decision Tables</b>
Calculate Adjusted Gross Income	CalculateAdjustedGrossIncome
Calculate Dependent Amount	CalculateDependentAmount
Calculate Taxable Income	CalculateTaxableIncome
Calculate Total Payments	CalculateTotalPayments
Define Tax using the standard Tax Table	DefineTax
Calculate Refund	CalculateRefundAndAmountYouOwe

The first column of this table defines the exact names for all of the above decisions. We will need to create decision tables for each of these decisions but at this point it is enough to just give these decision tables exact (unique) names – see the second column.

**User.** I noticed that names of decision tables in the second columns do not have spaces.

**OR.** This is necessary as they represent the names of executable Java functions contrary to the decision names in the first column that can use any English statements.

**User.** Now we can start creating Excel tables for the Excel-based decision tables.

**OR.** Not so fast. If you look at the second page of the 1040EZ from you will notice some limitations when this form can and cannot be used. It means that before we start any calculations we should validate if the taxpayer is eligible to use this form. I'd recommend defining two different decisions, one that validates a taxpayer eligibility to use this form and another that actually does the necessary 1040EZ calculations.

**User.** OK, we can add another decision in a separate decision table:

<b>Decision DecisionValidateTaxReturn</b>	
<b>Decisions</b>	<b>Execute Decision Tables</b>
Validate Eligibility	Validate Eligibility

But how can we connect these two decisions? We should execute the first decision only when the second one is successful.

**OR.** We may define a top-level decision with the following logic:

1. Execute a sub-decision that validates a taxpayer's 1040EZ eligibility
2. If a taxpayer is 1040EZ eligible  
Then execute sub-decision that calculates the tax return.

Let's call this top-level decision "Apply1040EZ".

**User.** OK, but I did now know that decisions can have sub-decisions.

**OR.** Actually, you can use multiple tables of the type "Decision" to represent any decision trees just in a tabular format (that frequently is more manageable than a graphical tree). You even can use conditions inside decision tables. Let me show you a possible solution:

Decision DetermineTaxReturn			
Condition		ActionPrint	ActionExecute
1040EZ Eligible		Decisions	Execute
		Validate	DecisionValidateTaxReturn
Is	TRUE	Calculate	DecisionCalculateTaxReturn
Is	FALSE	Do Not Calculate	

Because we use an optional "Condition" we have to add a second row. The keywords "Condition", "ActionPrint", and "ActionExecute" are defined in the standard OpenRules template "DecisionTemplate" – see the configuration file "DecisionTemplates.xls" in the folder "openrules.config". Here I am using a decision variable "1040EZ Eligible" that we will add to our glossary later on (probably as an attribute to a business concept "TaxReturn"). I assume that decision "ValidateTaxReturn" should set this decision variable to TRUE or FALSE.

**User.** Interesting... And if our validation fails (1040EZ Eligible will be FALSE) you only want to print the words "Do Not Validate" and do nothing.

**OR.** Exactly right. Note that the decision "ValidateTaxReturn" will always be executed (no conditions are defined). Also note that contrary to the decision tables, decisions always use a predefined parameter (decision) as defined in the template.

Now, when we are done with the definition of our decision and sub-decisions, we may concentrate on the decision tables.

## DEFINING DECISION TABLES

**OR.** The form 1040EZ gives us an opportunity to demonstrate different types of rules starting from very simple validation decision tables, then rules that deal with different calculation formulas, and finally with quite complex rules for calculation of dependent amounts. Let's start with validation rules for the names we already specified in the decision "ValidateTaxReturn".

### Validation Rules – Simple Decision Tables

**User.** OK. Eligibility rules are defined at the beginning of the second page of Form 1040EZ:

Form 1040EZ (2003)		Page 2
<b>Use this form if</b>	<ul style="list-style-type: none"> <li>• Your filing status is single or married filing jointly.</li> <li>• You (and your spouse if married filing jointly) were under age 65 and not blind at the end of 2003. If you were born on January 1, 1939, you are considered to be age 65 at the end of 2003.</li> <li>• You do not claim any dependents.</li> <li>• Your taxable income (line 6) is less than \$50,000.</li> <li>• You do not claim a deduction for educator expenses, the student loan interest deduction, or the tuition and fees deduction.</li> <li>• You do not claim an education credit, the retirement savings contributions credit, or the health coverage tax credit.</li> <li>• You had only wages, salaries, tips, taxable scholarship or fellowship grants, unemployment compensation, or Alaska Permanent Fund dividends, and your taxable interest was not over \$1,500. But if you earned tips, including allocated tips, that are not included in box 5 and box 10 of your W-2, you may not be able to use Form 1040EZ (see page 13). If you are planning to use Form 1040EZ for a child who received Alaska Permanent Fund dividends, see page 14.</li> <li>• You did not receive any advance earned income credit payments.</li> </ul>	

I think we can automatically validate only two eligibility conditions:

1. Your taxable income (line 6) is less than \$50,000
2. Your taxable interest was not over \$1,500

So, I can create the proper decision table "ValidateEligibility" as follows:

DecisionTable ValidateTaxReturn						
Condition		Condition		Conclusion		Message
TaxableIncome		TaxableInterest		1040EZ Eligible		Message
>=	50000			Is	FALSE	SORRY, YOU CAN NOT USE 1040EZ FORM (your TaxableIncome should be less than \$50,000)
		>	1500	Is	FALSE	SORRY, YOU CAN NOT USE 1040EZ FORM (your taxable interest > \$1,500)

**OR.** Looks good but I can see two problems here. First of all, what will happen if both eligibility conditions are met? In other words, who will set the decision variable “1040EZ Eligible” to TRUE?

**User.** It is easy to fix. I will add one more row at the end of the table that will set this decision variable to TRUE and will even produce a message that 1040EZ can be used. Here it goes.

DecisionTable ValidateTaxReturn						
Condition		Condition		Conclusion		Message
TaxableIncome		TaxableInterest		1040EZ Eligible		Message
>=	50000			Is	FALSE	SORRY, YOU CAN NOT USE 1040EZ FORM (your TaxableIncome should be less than \$50,000)
		>	1500	Is	FALSE	SORRY, YOU CAN NOT USE 1040EZ FORM (your taxable interest > \$1,500)
				Is	TRUE	YOU CAN USE 1040EZ FORM

**OR.** The OpenRules® engine will actually execute this decision table in the way you expect. However, theoretically this decision table violates a requirement that the order of rules (rows) inside a decision table should not matter. In this case it is easy to fix the reliance on the rule ordering by simply avoiding empty conditions as shown below:

DecisionTable ValidateTaxReturn						
Condition		Condition		Conclusion		Message
TaxableIncome		TaxableInterest		1040EZ Eligible		Message
>=	50000			Is	FALSE	SORRY, YOU CAN NOT USE 1040EZ FORM (your TaxableIncome should be less than \$50,000)
<	50000	>	1500	Is	FALSE	SORRY, YOU CAN NOT USE 1040EZ FORM (your taxable interest > \$1,500)
<	50000	<=	1500	Is	TRUE	YOU CAN USE 1040EZ FORM

**User.** I still like my initial version better because it is simpler. With your approach we have to mention thresholds like “50000” and “1500” in several places. What if we want to change such thresholds? What if we have not 2 but 4 or 5 condition columns? It would be much more difficult to maintain such a decision table.

**OR.** Your arguments are well taken. In reality it is really up to a rule designer (or to the company's official policy) to decide how strictly to follow the methodological principles.

Let's keep going. As mentioned earlier, I noticed not one but two problems with your decision table. Are your decision variables "Taxable Income" and "Taxable Interest" already defined by the time when you try to validate them?

**User.** Oh no! They are parts of the calculation logic. Let me look again at the first page of 1040EZ... OK, Taxable Interest is defined in Line 2 and simply should be entered by a taxpayer. However, the Taxable Income is defined in Line 6 and it should be calculated. I am afraid we have to repeat the first 3 calculation steps before starting validation:

- Calculate Adjusted Gross Income
- Calculate Dependent Amount
- Calculate Taxable Income (Line 6)

**OR.** No problem, simply add these steps to your decision table "ValidateTaxReturn".

**User.** Here it goes:

Decision DecisionValidateTaxReturn	
Decisions	Execute Decision Tables
Calculate Adjusted Gross Income	CalculateAdjustedGrossIncome
Calculate Dependent Amount	CalculateDependentAmount
Calculate Taxable Income	CalculateTaxableIncome
Define 1040EZ Eligibility	ValidateTaxReturn

But why should we repeat these steps inside both decisions? Maybe it is better to add one more decision, say "Pre-Process", that should be executed before "ValidateTaxReturn".

**OR.** You certainly can do that, and in some cases it will be the right approach. However, I'd rather treat our two decisions "Validate" and "Calculate" as loosely coupled services and would rather minimize unnecessary dependencies. Besides, we will reuse the same calculation decision tables anyway. So, let's stick to what you've already done.

## Using Formulas inside Decision Tables

**User.** OK. Now we have to put the actual calculation logic inside decision tables that we already have specified within our table "ValidateTaxReturn". The first decision table is "CalculateAdjustedGrossIncome". We will need to calculate the decision variable "Adjusted Gross Income" that is specified in Line 4 as a sum of Line 1 (Wages), Line 2 (Taxable Interest), and Line 3 (Unemployment Compensation). I can use a decision table with only

one conclusion for the decision variable “Adjusted Gross Income”. But how can I define a sum of different type variables?

**OR.** OpenRules® allows you to put any formulas inside decision table cells. They are not Excel’s formulas that start with a special character like “=”. You may simply write your formulas as follows:

DecisionTable CalculateAdjustedGrossIncome	
Action	
AdjustedGrossIncome	
<code>Wages + TaxableInterest + UnemploymentCompensation</code>	

**User.** Very nice! A quick question: if such formulas use decision variables not yet included in the glossary, should we add them later on?

**OR.** Right, that is a good point otherwise we will receive a syntax error. Before we jump to more complex calculation rules, let’s do very similar decision table “CalculateTotalPayment”.

**User.** It defines “Total Payments” as the sum of “Tax Withheld” and “Earned Income Credit”. Here it goes:

DecisionTable CalculateTotalPayments	
Action	
TotalPayments	
<code>TaxWithheld + EarnedIncomeCredit</code>	

**OR.** Good. You actually use the Friendly Enough Expression Language (FEEL) specified by the DMN standard.

Now, let’s leave the most complex rule “CalculateDependentAmout” for desert, and next work on the decision table to “CalculateTaxableIncome”.

## When Conditions and Conclusions Share Decision Variables

**OR.** Decision table “CalculateTaxableIncome” does not simply use a formula, but has an interesting flavor, if the formula for Taxable Income produces a negative result we need to set Taxable Income to 0. How would you approach this problem?

**User.** We may use a formula only in a rule conclusion like we did earlier and in our conditions simply check if `AdjustedGrossIncome` is larger or smaller of `DependentAmount`.

**OR.** Good. OpenRules® “`DecisionTableExecuteTemplate`” actually provides special optional conditions to compare different integer or real variables. As our variables are real , we may create the following decision table:

DecisionTable CalculateTaxableIncome			Action
ConditionRealOperReal			
Var	<oper>	Var	TaxableIncome
AdjustedGrossIncome	>	DependentAmount	<code>AdjustedGrossIncome - DependentAmount</code>
AdjustedGrossIncome	<=	DependentAmount	0

**User.** Seems quite natural- probably we have a similar condition for integer variables. I do not like this implementation – it seems that it calculates the same expression 3 times.

**OR.** Yes, and it is called “`ConditionIntOperInt`”. Now let’s switch to more complex dependent rules from the back page of the form 1040EZ. What does the form tells us about the Dependent Amount calculation?

## More Complex Decision Tables

**User.** Line 5 asks a question, “Can your parents (or somebody else) claim you on their return?” If the answer is “No” then the Dependent Amount will be \$7,800 if a taxpayer is single or \$15,600 if married filing jointly. However, if the answer is “Yes” we should go to the second page...

**OR.** OK, to start with something, I suggest we create a decision table “`CalculateDependentAmout`” with the first condition “Claimed As Dependent” is FALSE. How would you address the “single” vs. “married filing jointly” situation?

**User.** I don’t think we need a separate condition for a marital status – we may simple check condition for the decision variable “Married Filing Jointly”. Here is the proper decision table:

DecisionTable CalculateDependentAmount					
Condition		Condition		Conclusion	
Claimed As Dependent		Married Filing Jointly		Dependent Amount	
Is	FALSE	Is	FALSE	Is	7800
Is	FALSE	Is	TRUE	Is	15600

**OR.** Look good. However, we need to consider all other situations described on the second page in the case when the first condition is TRUE.

**User.** Here is what the second page says:

- A . Amount, if any, from line 1 on front       +  Enter total ► A.
- B. Minimum standard deduction . . . . . B.
- C. Enter the **larger** of line A or line B here . . . . . C.
- D. Maximum standard deduction. If **single**, enter \$4,750; if **married filing jointly**, enter \$9,500 . . . . . D.
- E. Enter the **smaller** of line C or line D here. This is your standard deduction . . . . . E.
- F. Exemption amount.
  - If single, enter -0-.
  - If married filing jointly and—
    - both you and your spouse can be claimed as dependents, enter -0-.
    - only one of you can be claimed as a dependent, enter \$3,050.
- G. Add lines E and F. Enter the total here and on line 5 on the front . . G.

So, we need to calculate all Lines from A to F and then define the decision variable “Dependent Amount” (also Line G) as a sum of Lines E and F. I can add conclusions for Lines A, B, C, D, E, and F and then we consider additional conditions when we need them.

**OR.** Sounds like a good plan. Go for it.

**User.** OK, let me go line by line.

Line A can be calculated using the formula: Wages + 250.

Line B is always a constant: 750.

Line C requires a maximum of Line A and Line B. Should I add conditions like A>B?

**OR.** Actually you do not have to. In your formula you can simply write:

$\max(\text{LineA}, \text{LineB})$ 

Here we are using the standard FEEL method  $\max(x, y)$  that provides a maximum between x and y.

**User.** Nice!

Let's look at Line D. We already have a condition for the decision variable "Married Filing Jointly". So, if it is FALSE we will use the constant 4,750, and if it is TRUE we will use 9,500.

Line E is very similar to Line C but it deals with a minimum of two numbers. Probably I can use a similar formula:

 $\min(\text{LineC}, \text{LineD})$ 

**OR.** Perfect. Now let's evaluate Line F.

**User.** Here we have two sub-conditions when the decision variable "Married Filing Jointly" is TRUE:

- both you and your spouse can be claimed as dependents, or
- only one of you can be claimed as a dependent.

We already have the decision variable "ClaimedAsDependent". Now we can define a new decision variable "SpouseClaimedAsDependent" and consider different combinations of these two decision variables. Right?

Wait a minute! If I am already working with the worksheet for dependents it means the decision variable "ClaimedAsDependent" is already TRUE. It also means that the first sub-condition should simple check if "SpouseClaimedAsDependent" is TRUE, and the second one – if it is FALSE. Why does Form 1040EZ ask a question in such a complicated way when they already know the answer?

**OR.** You are certainly pointing to a problem in probably one of the most used (and supposedly carefully validated) IRS forms! In spite of all attempts by the IRS to keep it as simple as possible they still made a mistake in their form.

**User.** Maybe they should try to do what we are doing. Anyway, I think I am ready to extend my previous table. It's becoming very wide, it looks better in the Excel file, but here it is in a slightly compressed form:

DecisionTable CalculateDependentAmount											
Condition		Condition		Condition	Action	Action	Action	Action	Action	Action	
ClaimedAsDependent		MarriedFilingJointly		SpouseClaimed AsDependent	LineA	LineB	LineC	LineD	LineE	LineF	Dependent Amount
Is	FALSE	Is	FALSE								7800
Is	FALSE	Is	TRUE								15600
Is	TRUE	Is	FALSE					4750		0	
Is	TRUE	Is	TRUE	Is	TRUE	Wages + 500	750	max(LineA, LineB)	9500	min(LineC, LineD)	0
Is	TRUE	Is	TRUE	Is	FALSE						LineE + LineF
											3050

**OR.** Let's check how it will work. The rule 3 covers a situation when a taxpayer is claimed as a dependent but she/he is single (or at least did not file jointly with his/her spouse). Yes, your Line D will be 4750 and Line F will be 0. That's correct. The next rules states that when a taxpayer is claimed as a dependent, married filing jointly is TRUE, and the spouse is also claimed as a dependent, then Line D will be 9500 and Line F will be 0. And finally the last rule is similar to a previous rule but covers the case when only the main taxpayer can be claimed as a dependent. In this case while Line D should stay the same, Line F should show 3050. It seems that this is a good representation of a complex set of conditions. I also like the way how you merged cells for the same values to avoid repeating the same formulas and values.

## DEFINING A BUSINESS GLOSSARY

**OR.** So, let's briefly summarize what we've already accomplished. We started with decisions and they led us to specifications of different decision tables. Inside the decision tables we freely introduced different decision variables assuming that they are somehow defined. Now, when we have all of the decision tables in place, it is time to actually define our decision variables. To do this we simply need to put all the decision variables in a Glossary table that has the following structure:

Glossary glossary			
Variable	Business Concept	Attribute	Domain

The first column will simply list all decision variables using exactly the same name that we used inside the decision tables. Then we have to associate different decision variables with different business concepts. It seems in our case we need use only one business concept "Tax Return" because it includes all of the input decision variables, like Wages or Married Filing Jointly, all intermediate decision variables such as Dependent Amount, and two output decision variables Return and Amount You Owe.

**User.** OK, here is the glossary table with the first two columns filled out:

Glossary glossary	
Variable Name	Business Concept
1040EZ Eligible	TaxReturn
MarriedFilingJointly	
ClaimedAsDependent	
SpouseClaimedAsDependent	
Wages	
TaxableInterest	
UnemploymentCompensation	
AdjustedGrossIncome	
DependentAmount	
LineA	
LineB	
LineC	
LineD	
LineE	
LineF	
TaxableIncome	
TaxWithheld	
EarnedIncomeCredit	
TotalPayments	
TaxForSingle	
TaxForMarriedFilingJointly	
Tax	
Refund	
AmountYouOwe	

**OR.** Good, I hope you did not forget any previously mentioned decision variable. But even if you did, OpenRules® will point us to such errors. It is good that you merged all rows in the second column showing that all decision variables belong to the same concept TaxReturn.

Now we need to fill out the remaining two columns. The column “Attribute” contains “technical” names of decision variables – these names will be used to connect our decision variables with attributes of objects used by actual applications that will use our decision “Apply1040EZ”. The application objects could be defined in Java, in Excel tables (like we will do in the next section), in XML, etc. The decision does not have to know about it, the only requirement is that the attribute names should follow the usual naming convention for identification in languages like Java, it basically means no spaces allowed. The last column “Domain” is optional but it can be useful to specify which values are allowed to be used for different decision variables. In our case the first 4 decision variables are Boolean, so we may define their domain as “TRUE, FALSE”, all other decision variables are real numbers. Here is our final glossary:

Glossary glossary			
Variable Name	Business Concept	Attribute	Domain
1040EZ Eligible	TaxReturn	eligible	TRUE,FALSE
MarriedFilingJointly		marriedFilingJointly	TRUE,FALSE
ClaimedAsDependent		claimedAsDependent	TRUE,FALSE
SpouseClaimedAsDependent		spouseClaimedAsDependent	TRUE,FALSE
Wages		wages	
TaxableInterest		taxableInterest	
UnemploymentCompensation		unemploymentCompensation	
AdjustedGrossIncome		adjustedGrossIncome	
DependentAmount		dependentAmount	
LineA		dependentLineA	
LineB		dependentLineB	
LineC		dependentLineC	
LineD		dependentLineD	
LineE		dependentLineE	
LineF		dependentLineF	
TaxableIncome		taxableIncome	
TaxWithheld		taxWithheld	
EarnedIncomeCredit		earnedIncomeCredit	
TotalPayments		totalPayments	
TaxForSingle		taxForSingle	
TaxForMarriedFilingJointly		taxForMarriedFilingJointly	
Tax		tax	
Refund		refund	
AmountYouOwe		amountYouOwe	

Do not forget that all columns in the very first row “Glossary glossary” should be merged.

**User.** OK, but how will we associate our Tax Return with actual tax returns?

**OR.** We may assume that the actual Tax Return will be put in our decision under the name “TaxReturn”. Here is a table DecisionObject that associates our business concept Tax Return with a Java object that we may get from the decision:

DecisionObject decisionObjects	
Business Concept	Business Object
TaxReturn	<code>:= decision.get("TaxReturn")</code>

**User.** But can we test our decision before we integrate it with an actual Java application? I do not know (and more importantly I do not want to know) how to create a Java object “TaxReturn”.

**OR.** Yes, we can define our own test data directly in Excel using OpenRules® Datatype and Data tables.

## DEFINING TEST DATA IN EXCEL

**OR.** First, we define a Datatype “TaxReturn”:

Datatype TaxReturn	
boolean	eligible
boolean	marriedFilingJointly
boolean	claimedAsDependent
boolean	spouseClaimedAsDependent
double	wages
double	taxableInterest
double	unemploymentCompensation
double	adjustedGrossIncome
double	dependentAmount
double	dependentLineA
double	dependentLineB
double	dependentLineC
double	dependentLineD
double	dependentLineE
double	dependentLineF
double	taxableIncome
double	taxWithheld
double	earnedIncomeCredit
double	totalPayments
double	tax
double	refund
double	amountYouOwe

Please note that we use exactly the same attribute names as in our glossary. You can use the following types for your attributes:

- String – for text attributes
- int – for integer numbers
- Boolean – for Boolean attributes
- double – for real numbers
- Date – for dates

By the way they correspond to the standard Java types and are case-sensitive.

Now we will create several test instances of the type TaxReturn:

Data TaxReturn taxReturns				
eligible	Eligible	FALSE	FALSE	FALSE
marriedFilingJointly	MarriedFilingJointly	TRUE	TRUE	FALSE
claimedAsDependent	claimedAsDependent	TRUE	TRUE	FALSE
spouseClaimedAsDependent	spouseClaimedAsDependent	TRUE	TRUE	FALSE
wages	wages	32026	42026	38000
taxableInterest	taxableInterest	1450	1450	1250
unemploymentCompensation	unemploymentCompensation	0	0	0
adjustedGrossIncome	adjustedGrossIncome	0	0	0
dependentAmount	dependentAmount	0	0	0
dependentLineA	dependentLineA	0	0	0
dependentLineB	dependentLineB	0	0	0
dependentLineC	dependentLineC	0	0	0
dependentLineD	dependentLineD	0	0	0
dependentLineE	dependentLineE	0	0	0
dependentLineF	dependentLineF	0	0	0
taxableIncome	taxableIncome	0	0	0
taxWithheld	taxWithheld	4530	4530	1530
earnedIncomeCredit	earnedIncomeCredit	300	300	300
totalPayments	totalPayments	0	0	0
taxForSingle	taxForSingle	0	0	0
taxForMarriedFilingJointly	taxForMarriedFilingJointly	0	0	0
tax	tax	0	0	0
refund	refund	0	0	0
amountYouOwe	amountYouOwe	0	0	0

Here we defined 3 test cases in the 3 last columns. Of course, you may add more test cases. Now to use OpenRules Test Harness, we need to define a special table “testCases”:

DecisionTableTest testCases			
#	ActionUseObject	ActionExpect	ActionExpect
Test ID	TaxReturn	Refund	AmountYouOwe
Test 1	:= taxReturns[0]	1934	0
Test 2	:= taxReturns[1]	434	0
Test 3	:= taxReturns[2]	0	2836

The first column specifies the name of the test case.

The second column of the type “ActionUseObject” specifies which test case from the previous table “taxReturns” should be used, e.g. := taxReturns[0] points to the first test case. Then we defined two columns of the type “ActionExpect” in which we describe what results we expect to receive for what variables.

**User.** So, in the first case we expect the Refund = 1934 and 0 as the AmountYouOwe. I think I got it.

**OR.** And when we execute these test cases, OpenRules will point us to possible mismatches between expected and actually received results.

## DECISION EXECUTION

Now we are ready to execute the defined decision against the above test data. The project “Decision1040EZ” includes the following Java launcher for our decision.

```
public static void main(String[] args) {
    String fileName = "file:rules/main/Decision.xls";
    Decision decision = new Decision("DetermineTaxReturn", fileName);
    decision.put("report", "On");
    decision.put("FEEL", "On");
    decision.test("testCases");
}
```

This code is used when you simply double-click on “run.bat”.

It will run our decision against all 3 test cases and will produce results which we may see in the following execution protocol:

```
*** Decision DetermineTaxReturn ***
Decision has been initialized

RUN TEST: Test 1
Decision Run has been initialized
Decision DetermineTaxReturn: Print Input Data
INPUT: TaxReturn(id=0) {
adjustedGrossIncome=0.0
amountYouOwe=0.0
claimedAsDependent=true
dependentAmount=0.0
dependentLineA=0.0
dependentLineB=0.0
dependentLineC=0.0
dependentLineD=0.0
dependentLineE=0.0
dependentLineF=0.0
```

```
earnedIncomeCredit=300
eligible=false
marriedFilingJointly=true
refund=0.0
spouseClaimedAsDependent=true
tax=0.0
taxForMarriedFilingJointly=0.0
taxForSingle=0.0
taxWithheld=4,530
taxableIncome=0.0
taxableInterest=1,450
totalPayments=0.0
unemploymentCompensation=0.0
wages=32,026
}
Decision DetermineTaxReturn: Validate
Decision DecisionValidateTaxReturn: Calculate Adjusted Gross Income
Assign: AdjustedGrossIncome = Wages + TaxableInterest + UnemploymentCompensation [33476.0]
Decision DecisionValidateTaxReturn: Calculate Dependent Amount
Assign: LineA = Wages + 500 [32526.0]
Assign: LineB = 750 [750.0]
Assign: LineC = max(LineA,LineB) [32526.0]
Assign: LineD = 9500 [9500.0]
Assign: LineE = min(LineC,LineD) [9500.0]
Assign: LineF = 0 [0.0]
Assign: DependentAmount = LineE + LineF [9500.0]
Decision DecisionValidateTaxReturn: Calculate Taxable Income
Assign: TaxableIncome = AdjustedGrossIncome - DependentAmount [23976.0]
Decision DecisionValidateTaxReturn: Define 1040EZ Eligibility
Conclusion: 1040EZ Eligible Is true [true]
YOU CAN USE 1040EZ FORM [produced by ValidateTaxReturn]
Decision DetermineTaxReturn: Calculate
Decision DecisionCalculateTaxReturn: Calculate Adjusted Gross Income
Assign: AdjustedGrossIncome = Wages + TaxableInterest + UnemploymentCompensation [33476.0]
Decision DecisionCalculateTaxReturn: Calculate Dependent Amount
Assign: LineA = Wages + 500 [32526.0]
Assign: LineB = 750 [750.0]
Assign: LineC = max(LineA,LineB) [32526.0]
Assign: LineD = 9500 [9500.0]
Assign: LineE = min(LineC,LineD) [9500.0]
Assign: LineF = 0 [0.0]
Assign: DependentAmount = LineE + LineF [9500.0]
Decision DecisionCalculateTaxReturn: Calculate Taxable Income
Assign: TaxableIncome = AdjustedGrossIncome - DependentAmount [23976.0]
Decision DecisionCalculateTaxReturn: Calculate Total Payments
Assign: TotalPayments = TaxWithheld + EarnedIncomeCredit [4830.0]
Decision DecisionCalculateTaxReturn: Define Tax using the standard Tax Table
Decision DefineTax: Calculate Tax Table
Assign: TaxForSingle = 3246 [3246.0]
Assign: TaxForMarriedFilingJointly = 2896 [2896.0]
Decision DefineTax: Calculate Tax
Assign: Tax = $TaxForMarriedFilingJointly [2896.0]
Decision DecisionCalculateTaxReturn: Calculate Refund
```

Assign: Refund = TotalPayments - Tax [1934.0]

Assign: AmountYouOwe = 0 [0.0]

Decision DetermineTaxReturn: Print Results

RESULTS: TaxReturn(id=0) {

adjustedGrossIncome=33,476

amountYouOwe=0.0

claimedAsDependent=true

dependentAmount=9,500

dependentLineA=32,526

dependentLineB=750

dependentLineC=32,526

dependentLineD=9,500

dependentLineE=9,500

dependentLineF=0.0

earnedIncomeCredit=300

eligible=true

marriedFilingJointly=true

refund=1,934

spouseClaimedAsDependent=true

tax=2,896

taxForMarriedFilingJointly=2,896

taxForSingle=3,246

taxWithheld=4,530

taxableIncome=23,976

taxableInterest=1,450

totalPayments=4,830

unemploymentCompensation=0.0

wages=32,026

}

Decision has been finalized

Validating results for the test <Test 1>

Test 1 was successful

RUN TEST: Test 2

Decision Run has been initialized

Decision DetermineTaxReturn: Print Input Data

INPUT: TaxReturn(id=0) {

adjustedGrossIncome=0.0

amountYouOwe=0.0

claimedAsDependent=true

dependentAmount=0.0

dependentLineA=0.0

dependentLineB=0.0

dependentLineC=0.0

dependentLineD=0.0

dependentLineE=0.0

dependentLineF=0.0

earnedIncomeCredit=300

eligible=false

marriedFilingJointly=true

refund=0.0

spouseClaimedAsDependent=true

tax=0.0

```
taxForMarriedFilingJointly=0.0
taxForSingle=0.0
taxWithheld=4,530
taxableIncome=0.0
taxableInterest=1,450
totalPayments=0.0
unemploymentCompensation=0.0
wages=42,026
}
Decision DetermineTaxReturn: Validate
Decision DecisionValidateTaxReturn: Calculate Adjusted Gross Income
Assign: AdjustedGrossIncome = Wages + TaxableInterest + UnemploymentCompensation [43476.0]
Decision DecisionValidateTaxReturn: Calculate Dependent Amount
Assign: LineA = Wages + 500 [42526.0]
Assign: LineB = 750 [750.0]
Assign: LineC = max(LineA,LineB) [42526.0]
Assign: LineD = 9500 [9500.0]
Assign: LineE = min(LineC,LineD) [9500.0]
Assign: LineF = 0 [0.0]
Assign: DependentAmount = LineE + LineF [9500.0]
Decision DecisionValidateTaxReturn: Calculate Taxable Income
Assign: TaxableIncome = AdjustedGrossIncome - DependentAmount [33976.0]
Decision DecisionValidateTaxReturn: Define 1040EZ Eligibility
Conclusion: 1040EZ Eligible Is true [true]
YOU CAN USE 1040EZ FORM [produced by ValidateTaxReturn]
Decision DetermineTaxReturn: Calculate
Decision DecisionCalculateTaxReturn: Calculate Adjusted Gross Income
Assign: AdjustedGrossIncome = Wages + TaxableInterest + UnemploymentCompensation [43476.0]
Decision DecisionCalculateTaxReturn: Calculate Dependent Amount
Assign: LineA = Wages + 500 [42526.0]
Assign: LineB = 750 [750.0]
Assign: LineC = max(LineA,LineB) [42526.0]
Assign: LineD = 9500 [9500.0]
Assign: LineE = min(LineC,LineD) [9500.0]
Assign: LineF = 0 [0.0]
Assign: DependentAmount = LineE + LineF [9500.0]
Decision DecisionCalculateTaxReturn: Calculate Taxable Income
Assign: TaxableIncome = AdjustedGrossIncome - DependentAmount [33976.0]
Decision DecisionCalculateTaxReturn: Calculate Total Payments
Assign: TotalPayments = TaxWithheld + EarnedIncomeCredit [4830.0]
Decision DecisionCalculateTaxReturn: Define Tax using the standard Tax Table
Decision DefineTax: Calculate Tax Table
Assign: TaxForSingle = 5304 [5304.0]
Assign: TaxForMarriedFilingJointly = 4396 [4396.0]
Decision DefineTax: Calculate Tax
Assign: Tax = $TaxForMarriedFilingJointly [4396.0]
Decision DecisionCalculateTaxReturn: Calculate Refund
Assign: Refund = TotalPayments - Tax [434.0]
Assign: AmountYouOwe = 0 [0.0]
Decision DetermineTaxReturn: Print Results
RESULTS: TaxReturn(id=0) {
    adjustedGrossIncome=43,476
    amountYouOwe=0.0
```

```
claimedAsDependent=true
dependentAmount=9,500
dependentLineA=42,526
dependentLineB=750
dependentLineC=42,526
dependentLineD=9,500
dependentLineE=9,500
dependentLineF=0.0
earnedIncomeCredit=300
eligible=true
marriedFilingJointly=true
refund=434
spouseClaimedAsDependent=true
tax=4,396
taxForMarriedFilingJointly=4,396
taxForSingle=5,304
taxWithheld=4,530
taxableIncome=33,976
taxableInterest=1,450
totalPayments=4,830
unemploymentCompensation=0.0
wages=42,026
}
```

Decision has been finalized

Validating results for the test <Test 2>

Test 2 was successful

RUN TEST: Test 3

Decision Run has been initialized

Decision DetermineTaxReturn: Print Input Data

```
INPUT: TaxReturn(id=0) {
    adjustedGrossIncome=0.0
    amountYouOwe=0.0
    claimedAsDependent=false
    dependentAmount=0.0
    dependentLineA=0.0
    dependentLineB=0.0
    dependentLineC=0.0
    dependentLineD=0.0
    dependentLineE=0.0
    dependentLineF=0.0
    earnedIncomeCredit=300
    eligible=false
    marriedFilingJointly=false
    refund=0.0
    spouseClaimedAsDependent=false
    tax=0.0
    taxForMarriedFilingJointly=0.0
    taxForSingle=0.0
    taxWithheld=1,530
    taxableIncome=0.0
    taxableInterest=1,250
    totalPayments=0.0
}
```

```
unemploymentCompensation=0.0
wages=38,000
}
Decision DetermineTaxReturn: Validate
Decision DecisionValidateTaxReturn: Calculate Adjusted Gross Income
Assign: AdjustedGrossIncome = Wages + TaxableInterest + UnemploymentCompensation [39250.0]
Decision DecisionValidateTaxReturn: Calculate Dependent Amount
Assign: DependentAmount = 7800 [7800.0]
Decision DecisionValidateTaxReturn: Calculate Taxable Income
Assign: TaxableIncome = AdjustedGrossIncome - DependentAmount [31450.0]
Decision DecisionValidateTaxReturn: Define 1040EZ Eligibility
Conclusion: 1040EZ Eligible Is true [true]
YOU CAN USE 1040EZ FORM [produced by ValidateTaxReturn]
Decision DetermineTaxReturn: Calculate
Decision DecisionCalculateTaxReturn: Calculate Adjusted Gross Income
Assign: AdjustedGrossIncome = Wages + TaxableInterest + UnemploymentCompensation [39250.0]
Decision DecisionCalculateTaxReturn: Calculate Dependent Amount
Assign: DependentAmount = 7800 [7800.0]
Decision DecisionCalculateTaxReturn: Calculate Taxable Income
Assign: TaxableIncome = AdjustedGrossIncome - DependentAmount [31450.0]
Decision DecisionCalculateTaxReturn: Calculate Total Payments
Assign: TotalPayments = TaxWithheld + EarnedIncomeCredit [1830.0]
Decision DecisionCalculateTaxReturn: Define Tax using the standard Tax Table
Decision DefineTax: Calculate Tax Table
Assign: TaxForSingle = 4666 [4666.0]
Assign: TaxForMarriedFilingJointly = 4014 [4014.0]
Decision DefineTax: Calculate Tax
Assign: Tax = $TaxForSingle [4666.0]
Decision DecisionCalculateTaxReturn: Calculate Refund
Assign: Refund = 0 [0.0]
Assign: AmountYouOwe = Tax - TotalPayments [2836.0]
Decision DetermineTaxReturn: Print Results
RESULTS: TaxReturn(id=0) {
    adjustedGrossIncome=39,250
    amountYouOwe=2,836
    claimedAsDependent=false
    dependentAmount=7,800
    dependentLineA=0.0
    dependentLineB=0.0
    dependentLineC=0.0
    dependentLineD=0.0
    dependentLineE=0.0
    dependentLineF=0.0
    earnedIncomeCredit=300
    eligible=true
    marriedFilingJointly=false
    refund=0.0
    spouseClaimedAsDependent=false
    tax=4,666
    taxForMarriedFilingJointly=4,014
    taxForSingle=4,666
    taxWithheld=1,530
    taxableIncome=31,450
```

```
taxableInterest=1,250
totalPayments=1,830
unemploymentCompensation=0.0
wages=38,000
}
Decision has been finalized
Validating results for the test <Test 3>
Test 3 was successful
All 3 tests succeeded!
```

When you decide to pass your already well-tested decision to your IT people, they will simply replace the object “TaxReturn” with real Java objects. They may use their own class TaxReturn. The only integration requirement is that the names of attributes in their classes should be the same as in our Glossary. An important point is that your IT guys do not have to even look at your decision and decision tables.

This completes the decision part of our tutorial.

**User.** This decision was not as simple as the examples from “Getting Started” but it gave me many more choices that I can use in real-world decision. Thank you.

**OR.** You are welcome. You can find many more details in the [User Manual](#). You also may consider a completely opposite approach to the 1040EZ implementation using OpenRules® Dialog or ORD™ – see <http://openrules.com/pdf/Tutorial.Dialog1040EZ.pdf>.

## APPENDIX. FORM 1040EZ

**Department of the Treasury—Internal Revenue Service**  
**Income Tax Return for Single and**  
**Joint Filers With No Dependents (99) 2003**

OMB No. 1545-0675

<b>Label</b> (See page 12.) <b>Use the IRS label.</b> Otherwise, please print or type.  <b>Presidential            Election            Campaign            (page 12)</b>	Your first name and initial		Last name		Your social security number	
	If a joint return, spouse's first name and initial		Last name		Spouse's social security number	
	Home address (number and street). If you have a P.O. box, see page 12.			Apt. no.		
	City, town or post office, state, and ZIP code. If you have a foreign address, see page 12.					
<b>▲ Important! ▲</b> You must enter your SSN(s) above.						
<b>You      Spouse</b> <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Yes <input type="checkbox"/> No						
Note. Checking "Yes" will not change your tax or reduce your refund. Do you, or your spouse if a joint return, want \$3 to go to this fund? ►						
<b>Income</b> <b>Attach Form(s) W-2 here.</b> Enclose, but do not attach, any payment.						
<b>Note. You must check Yes or No.</b>						
<b>Payments and tax</b>						
<b>Refund</b> Have it directly deposited! See page 19 and fill in 11b, 11c, and 11d.						
<b>Amount you owe</b>						
<b>Third party designee</b>						
<b>Sign here</b> Joint return? See page 11. Keep a copy for your records.						
<b>Paid preparer's use only</b>						
For Disclosure, Privacy Act, and Paperwork Reduction Act Notice, see page 23.						
Cat. No. 11329W      Form 1040EZ (2003)						

**Use  
this  
form if**

- Your filing status is single or married filing jointly.
- You (and your spouse if married filing jointly) were under age 65 and not blind at the end of 2003. If you were born on January 1, 1939, you are considered to be age 65 at the end of 2003.
- You do not claim any dependents.
- Your taxable income (line 6) is less than \$50,000.
- You do not claim a deduction for educator expenses, the student loan interest deduction, or the tuition and fees deduction.
- You do not claim an education credit, the retirement savings contributions credit, or the health coverage tax credit.
- You had only wages, salaries, tips, taxable scholarship or fellowship grants, unemployment compensation, or Alaska Permanent Fund dividends, and your taxable interest was not over \$1,500. But if you earned tips, including allocated tips, that are not included in box 5 and box 7 of your W-2, you may not be able to use Form 1040EZ (see page 13). If you are planning to use Form 1040EZ for a child who received Alaska Permanent Fund dividends, see page 14.
- You did not receive any advance earned income credit payments.

If you are not sure about your filing status, see page 11. If you have questions about dependents, use TeleTax topic 354 (see page 6). If you cannot use this form, use TeleTax topic 352 (see page 6).

**Filling in  
your  
return**

For tips on how to avoid common mistakes, see page 21.

If you received a scholarship or fellowship grant or tax-exempt interest income, such as on municipal bonds, see the booklet before filling in the form. Also, see the booklet if you received a Form 1099-INT showing Federal income tax withheld or if Federal income tax was withheld from your unemployment compensation or Alaska Permanent Fund dividends.

Remember, you must report all wages, salaries, and tips even if you do not get a Form W-2 from your employer. You must also report all your taxable interest, including interest from banks, savings and loans, credit unions, etc., even if you do not get a Form 1099-INT.

**Worksheet  
for  
dependents  
who  
checked  
"Yes" on  
line 5**

(keep a copy for  
your records)

Use this worksheet to figure the amount to enter on line 5 if someone can claim you (or your spouse if married filing jointly) as a dependent, even if that person chooses not to do so. To find out if someone can claim you as a dependent, use TeleTax topic 354 (see page 6).

- |  |                         |                                |                         |
|--|-------------------------|--------------------------------|-------------------------|
| A. Amount, if any, from line 1 on front  | <input type="text"/>    | + 250.00 Enter total ►         | A. <input type="text"/> |
| B. Minimum standard deduction  | <input type="text"/>    | B. <input type="text"/> 750.00 |                         |
| C. Enter the larger of line A or line B here   | <input type="text"/>    | C. <input type="text"/>        |                         |
| D. Maximum standard deduction. If single, enter \$4,750; if married filing jointly, enter \$9,500  | <input type="text"/>    | D. <input type="text"/>        |                         |
| E. Enter the smaller of line C or line D here. This is your standard deduction   | <input type="text"/>    | E. <input type="text"/>        |                         |
| F. Exemption amount.   | F. <input type="text"/> |                                |                         |
| ● If single, enter -0-.  |                         |                                |                         |
| ● If married filing jointly and—<br>—both you and your spouse can be claimed as dependents, enter -0-.<br>—only one of you can be claimed as a dependent, enter \$3,050. |                         |                                |                         |

G. Add lines E and F. Enter the total here and on line 5 on the front . . . G.

If you checked "No" on line 5 because no one can claim you (or your spouse if married filing jointly) as a dependent, enter on line 5 the amount shown below that applies to you.

- Single, enter \$7,800. This is the total of your standard deduction (\$4,750) and your exemption (\$3,050).
- Married filing jointly, enter \$15,600. This is the total of your standard deduction (\$9,500), your exemption (\$3,050), and your spouse's exemption (\$3,050).

**Mailing  
return**

Mail your return by April 15, 2004. Use the envelope that came with your booklet. If you do not have that envelope or if you moved during the year, see the back cover for the address to use.