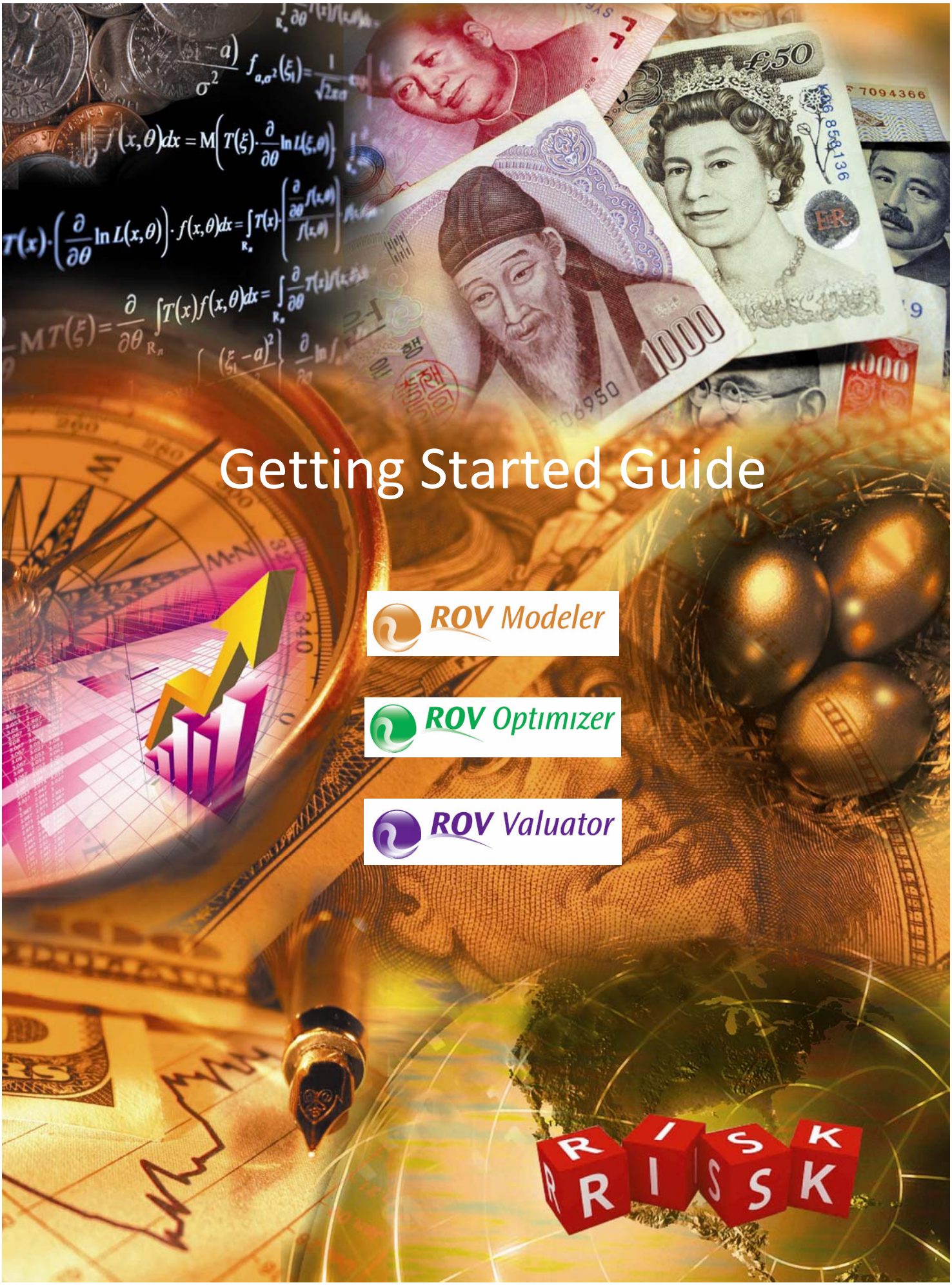

Getting Started Guide
ROV Modeler
ROV Optimizer
ROV Valuator
RISK
RISK

TABLE OF CONTENTS

Introduction

This help file introduces the structure of ROV Risk Modeler software suite, developed by Real Options Valuation, Inc. This software takes the modeling outside of Excel and into the database environment, allowing the end user the ability to directly link to databases and large data files, clean the data and run advanced analytics at very high speeds. This ROV Risk Modeler comprises several modules, including:

- ROV Modeler is a customizable advanced analytical modeling software module for solving multiple types of models, including computing advanced models in various industries, advanced forecasting and simulation models, historical back-fitting, time-series forecasts (ARIMA, Autoeconometrics, Regression, stochastic processes, and others), volatility computation (GARCH), and many other applications. Also included in this module (as well as the Basel Modeler and Risk Optimizer modules) are the ability to link and download from various database and data sources (e.g., Oracle OFDM, SQL Server, Excel, CSV, text, and other ODBC compliant databases), screen and clean the data prior to use (applying SQL commands and data cleansing routines), compute new variables based on existing data, run Monte Carlo Risk Simulations, apply data and distributional fitting, and other routines. This module is also customizable in that users can modify the functions list, descriptions, and what models or applications to show, allowing users to customize the tool to fit his or her needs.
- ROV Basel Modeler is an advanced analytical software module for solving multiple types of models, including computing advanced models in various industries (e.g., for banks, insurance and financial services companies, models such as probability of default, loss given default, exposure at default, Value at Risk, and other key metrics). It also functions like the ROV Modeler as described above.
- ROV Risk Optimizer has the ability to quickly run project selection and investment or project portfolios and nonlinear optimization with simulation and stochastic optimization, all the while using discrete integer, binary and continuous variables subject to multiple constraints.
- ROV Risk Valuator has over 600 models and functions to value everything from simple options and exotic options to commodities, futures, and risk-return profiles of asset portfolios, and so forth. Please see the Appendix for a more detailed listing of the models that are available.
- ROV Risk Charter runs different Modeler and Optimizer profiles and returns predefined XML files that can be used by ROV Dashboard to generate dynamic charts, tables, pivot tables and reports. ROV Dashboard is another software program developed by Real Options Valuation, Inc.
- ROV Scheduler runs different Modeler and Optimizer profiles and returns the results in flat text files that can be saved or easily uploaded into Excel or other databases.
- ROV Portfolio runs multiple Modeler and Optimizer profiles and multiple models immediately. This is similar to the Scheduler in that multiple models can be chosen to run at once from different profiles, but the difference is that the analyses are run immediately as opposed to being scheduled to run at a later time.

System Requirements

This software suite can be run in any Windows or MAC environment (MAC operating systems require Parallels or Virtual Machine to emulate a Windows environment), and is compatible with Microsoft Excel as well as other ODBC compliant databases and data files. The software suite requires 100MB of free disk space and recommended minimum 1GB of RAM for best performance. We recommend that the user has administrative rights (this is by default on most personal computers) but can also run on logins with limited user rights (simply install the software to a nonprotected file/folder location in order to properly run the software).

Copyright and Contact Information

The contact information for this software's developer is:
Real Options Valuation, Inc.
4101F Dublin Boulevard, Suite 425
Dublin, California 94568 USA
admin@realoptionsvaluation.com
Tel: +1.925.271.4438 Fax: +1.925.369.045
www.realoptionsvaluation.com

Risk Modeler

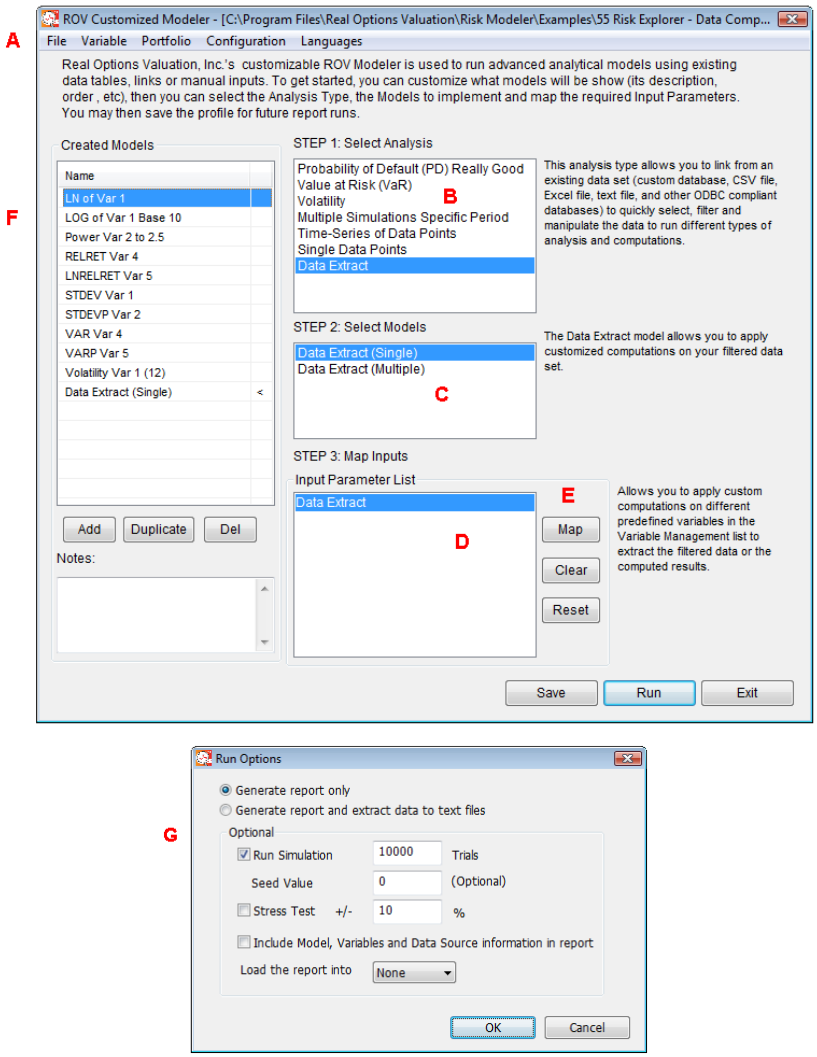
ROV Risk Modeler (also known as ROV Modeler for simplicity) is a simulation and advanced analytical tool that provides many advanced models to simulate, fit, forecast, value, and reports the results to the user. The software is compatible with different ODBC compliant data tables and databases (e.g., Excel, Oracle OFDM, SQL Server, and others) and allows the user to link to an existing database, manually inputting data, or setting simulation assumptions, in order to obtain the inputs to run the analyses. In addition, the software is customizable in the sense that the models, descriptions and analytics used can be customized (you can change the name of a method, the description, or delete the entire model altogether). This tool runs outside of Excel or any other databases and the computations occur at extremely fast speeds. For instance, if you have a few million data points saved in an Excel spreadsheet, opening this file in Excel might take 10 to 30 minutes depending on your system's performance, and making a change or performing a calculation inside Excel will take just as long. If you need to repeat the process for multiple periods (e.g., each Excel file is a month or week, and you need to replicate the analysis every week or month, or process multiple files at the same time, this task is intractable within Excel). In addition, most database software tools are only useful for storing data and not for data analysis or manipulation. Risk Modeler is the database intermediary in the sense that it is used for performing highly data intensive computations, and these computations can be repeated and scheduled to run as required.

When you start the ROV Modeler software, you will see a user interface as shown below. Briefly, the software interface has a menu bar **[A]**, a set of analysis types or family of models **[B]**, list of the models in the analysis type **[C]**, the specific inputs in the selected model **[D]**, the ability to map or link the variables in the model **[E]**, and the ability to create multiple models in a single profile **[F]**. To illustrate, an example of a type or family of models might be Volatility **[B]**. That is, there are several ways to compute volatility of some time-series data, including using a GARCH model, a logarithmic returns model, exponentially weighted moving average model, and so forth, where these models are listed in the models section **[C]**. Depending on the model chosen, the data required would be different, and would be listed in the inputs section **[D]**. For each input variable required in the selected model, you can map each of these to some data **[E]**. You may then decide to perform other model computations and these are listed on the created models section by clicking on the ADD button to add new models **[F]** to the list. Each of the models in the created models list **[F]** can be run by itself (select the model to run and click on the RUN button) and simulations, stress tests and reports can be generated as required **[G]**. The analysis will be run very quickly and the results will be shown **[H]**.

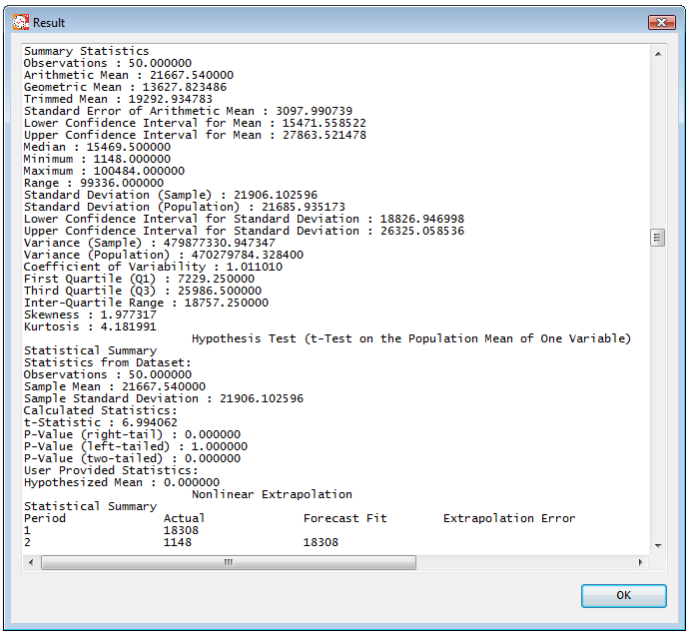
To summarize, these are the basic steps required to run ROV Risk Modeler:

1. Select the analysis type **[B]**
2. Select the model type **[c]**
3. Select one input variable at a time **[D]** and map these to existing data or enter in your data (E)
4. Select the model to run **[F]** and run the analysis **[G]**

Note that ROV Modeler can run one model at a time. If you wish to run multiple models from a single or multiple profiles at the same time, you can use the ROV Portfolio tool or the ROV Schedule if you wish these models to run at specific scheduled days and times.



H



Variable Mapping

When you click on the MAP button **[E]**, you will be presented with several choices **[I]** to map your data to the selected input variable. These include the ability to perform a data link to existing data files and databases, manual inputs by typing in or copy and pasting in some data, data compute to modify and run calculations of other variables prior to using the data, set assumptions to set simulation assumptions and probability distributions to run simulations on the model, or model fitting to take a large set of existing models to run a statistical best-fit on all 24 probability distributions to decide which distributions and parameters best fit the data such that the distribution will be used in the model. The following illustrates the details of each type of variable mapping and how to perform the required tasks.

Data Link

There are seven types of ODBC data connect for accessing existing data when you select the Data Link option **[J]**. To get started, provide the variable a name (the default name for data linking starts with the prefix "DL" so that you can identify if a variable is a linked variable or otherwise) and click on Open Database **[K]**. Here, you can see the seven types of databases and datafiles supported by ROV Modeler **[L]**, from Excel files to Oracle databases and SQL Servers, as well as regular CSV data files and other ODBC compliant databases (DSN format). For instance, if you select the Connect to Excel option **[L]**, click on the Browse button to locate the Excel file you want to link to and click OK. In the list of Available Fields section **[M]** you will see the Excel file name and when you click on it to expand, you will see the list of worksheets, and in each worksheet, the available variables will be listed. Typically, when using Excel as a data repository, make sure your variables are arranged in columns with the first row set as the name of the variable. You can then select the variable you wish to use and click on the >> button to select it, or >>> to add all the variables at once (alternatively, you can select the variables you wish to remove in the Selected Fields section and click on the << to remove the selected variable or <<< to remove all variables).

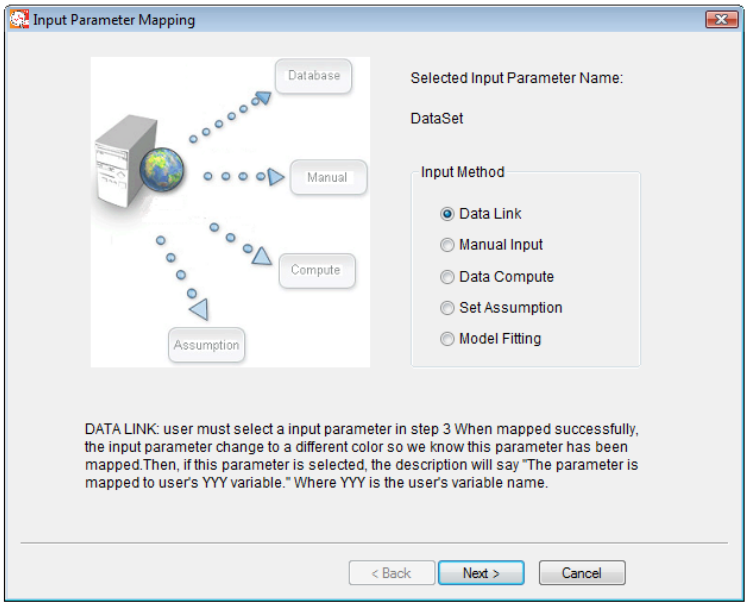
Other sources of data links are also available. As an example, for the Oracle data connect option, you will need to setup the required login inputs such as User Name and Password to access the database. When clicking on the OK button, the software will call the database connect method to connect the specified database. For different types of ODBC applications, the software codes are wrapped with popular calling methods such as CONNECT, QUERY, and so forth.

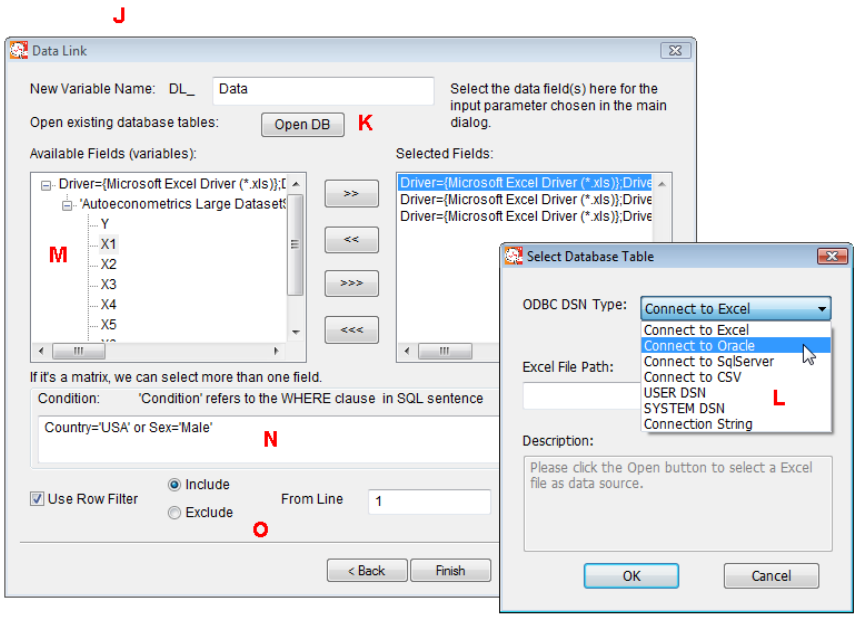
You can also perform some data cleansing routines using basic SQL commands **[N]**. For more details on using SQL commands, please refer to the appendix on some examples of using SQL filtering. Finally, there is a simple row filter tool **[O]** that can quickly filter in or out certain data (e.g., include or exclude data from rows *n* to *m*).

TIP: Saving into CSV File Format

As the CSV file format is the most commonly used format (this file format is compatible with most databases as a means to upload or download data), here are some getting started notes on creating the CSV data file.

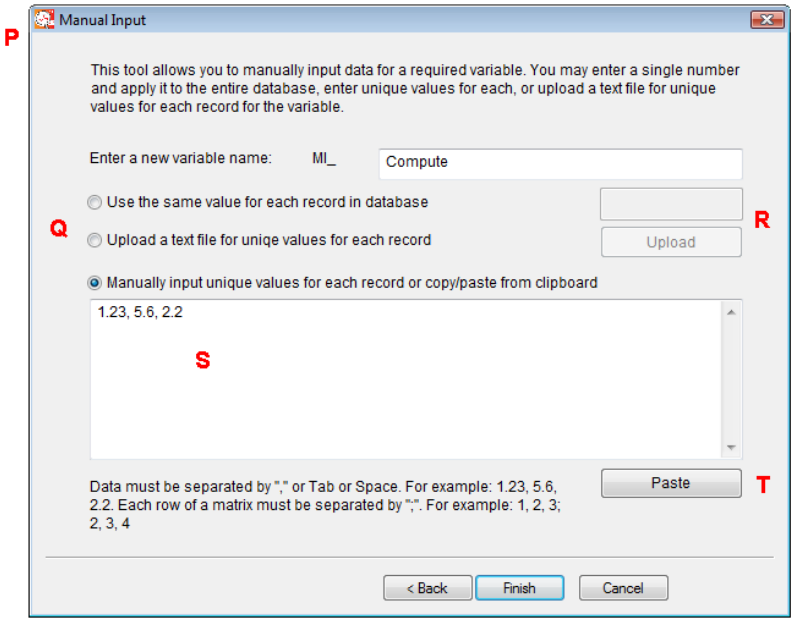
- It is always advisable to change a flat text data file into CSV as it has more features and the data can be viewed quickly and easily. To convert a text file to CSV, within Excel, do a File, Open and open the text file (go through the data file filter with space or tab delimited). Then Save As the file into CSV Comma Delimited.
- When manipulating CSV files, make sure that you do not add rows, values or type in data at the bottom (after the end of the data set). This is because whatever happens at the bottom of the CSV file is saved even if you have deleted the cell values. If you have done some computations at the bottom, please select the rows and perform a DELETE ROW(s) to eliminate all residual items that will be saved in the CSV (this is because deleted cells are assumed to contain empty values). Doing the delete row is critical otherwise the SQL upload will include empty elements and the computed values might be incorrect.
- It is also good practice that the first row of the data has the Variable name. Please note that Variable names can have spaces and special characters for Risk Modeler to work. Nonetheless, in some other databases, special characters and spaces might not be allowed and when creating your dataset be aware of this limitation. Therefore, it is always safer to not add spaces and special characters as variable names (e.g., do not use things like @, %, #, &, / and so forth).
- If the first row of data has an integer value (e.g., 0, 1, 2, etc) then make sure that it has decimals associated with the value. Sometimes certain databases using MySQL and SQL scripts may identify that as a string instead of a value. It is always a good idea to double check this. You can always change the number of decimals in CSV when editing in Excel. Just add a few decimals. This is simply as a precaution for database manipulation when uploading and downloading files.





Manual Input

The second approach is manual input. Here, instead of linking to a large data file or database to run detailed computations, sometimes your data set might be small or manually typed in. In this case, select the manual input option **[I]**. The default prefix for manual input is "MI" **[P]**. Start by entering a name for this variable and select if you wish to reuse the same value for each record in the database **[Q]** (i.e., suppose you chose a model requiring 5 input variables, X1, X2,…, X5, where each variable has 100,000 data points (rows), and you are mapping variable X5 currently, and would like to use the same value such as 0.10 for all 100,000 rows without having to enter each one of them), upload the date from a text file **[R]**, or manually enter in the data **[S]** (you can also paste the data from another file or the computer's clipboard **[T]**). Click Finish when you are done.



TIP: Manually Typing in Data

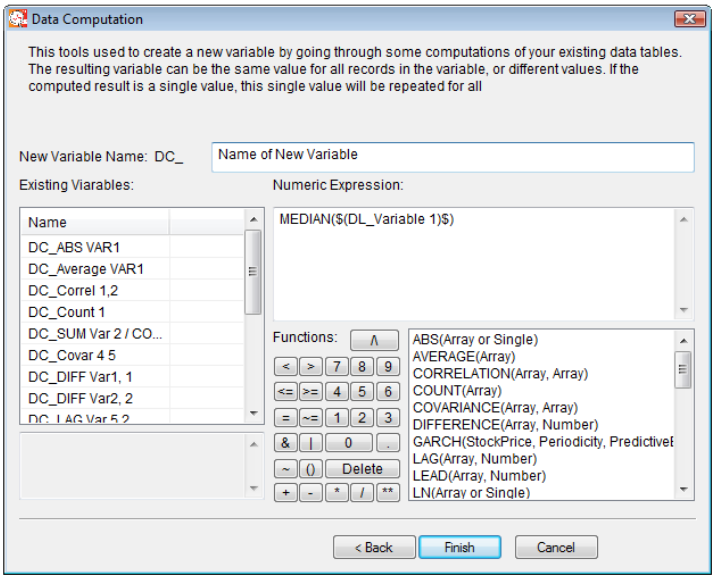
As sometimes you will be manually typing in some data, it is important to briefly understand the data format that is required.

- If the data you wish to enter is from a single variable then use COMMA as a data item separator. For instance, suppose you are entering 4 months of sales data, they should be: 54256, 222930.23, 111202.3, 132334 separated by commas (you can use as many decimal precision points as you wish). Therefore, do not use commas as the thousand-separator. In other words, one thousand dollars and fifty cents should be entered as 1000.5 or 1000.50 instead of using 1,000.50 and do not use any currency signs (e.g., $).
- If the variable requires multiple lines, then the SEMI-COLON is the line separator. For instance, suppose you need to enter in a 2x2 correlation matrix (there will be two columns and two rows) then the data can be entered in as 1, 0.1; 0.1, 1 with commas as the item-separator on the same row and semi-colon to denote a new row. You can see this clearly later on in the ROV Risk Valuator tool for models such as the Value at Risk or Portfolio Risk calculations.
- These are the default separators but you can change them if you wish, by going to the menu item and selecting Configuration and choosing Data Separator.

Data Compute

The third option is apply the data compute approach on previously mapped data **[U]**. For instance, you may map additional variables that are not used directly in a model (see the Variable Management section for details) but used as an intermediate variable. For example, in the screen shot **[U]**, we see that the data link variable "DL_Variable 1" that is probably linked from an existing database of multiple values is used as an intermediate variable. The value that is used is the median of the data. Note that the list of previously mapped variables is shown on the left side called Existing Variables and the functions as well as mathematical computations are available at the bottom. You can perform any types of computations as required using this data compute method.

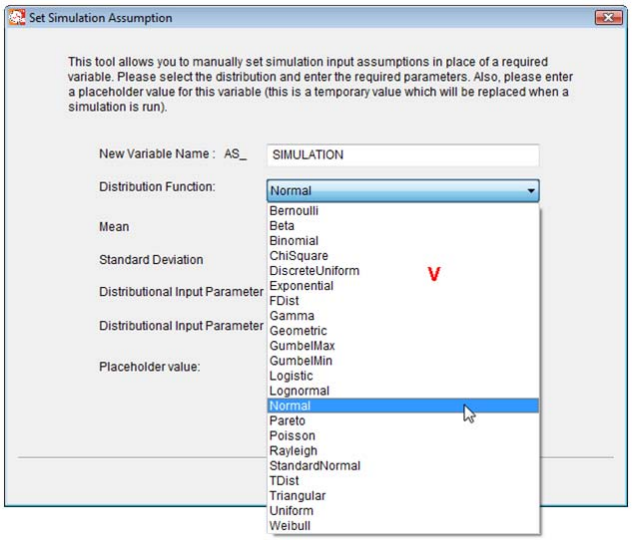
U



Set Simulation Assumption

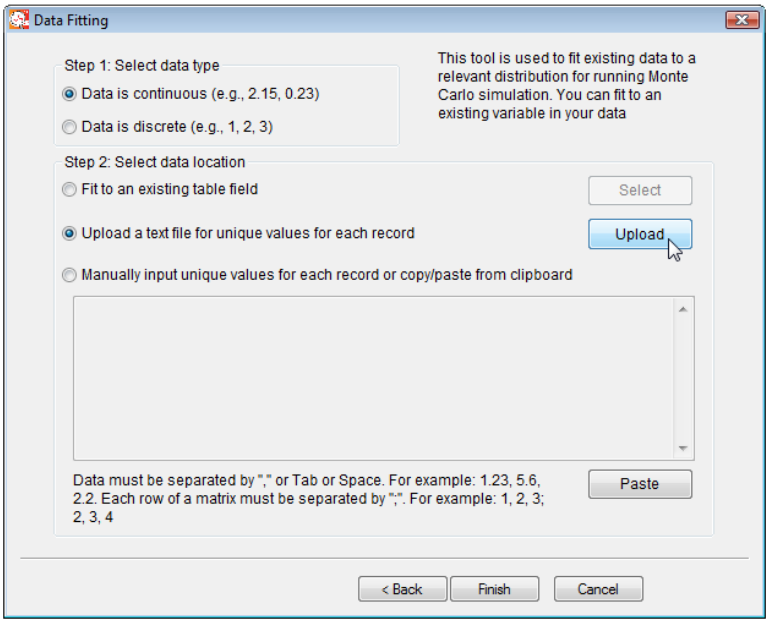
The fourth option is to set a simulation distributional assumption to the variable [I]. That is, in order to run a simulation of thousands of trials in the model, you will first need to set the appropriate simulation assumption by choosing the distribution of choice and enter in the relevant input parameters [V]. The default prefix is "AS" for assumption. Start by entering a new variable name and then selecting the distribution of choice. There are 22 distributions available for you to select. Enter in the required input parameters for the distribution selected (e.g., enter in the Mean and Standard Deviation values if you selected the Normal Distribution). You should also enter in a Placeholder Value (this is the temporary value used to compute the model if you do not run a simulation when running the model).

Note: It is outside the scope of this quick help document to illustrate the mathematical and probability details of each distribution. Please refer to *"Modeling Risk: Applying Monte Carlo Simulation, Real Options Analysis, Stochastic Forecasting, and Portfolio Optimization" by Dr. Johnathan Mun (Wiley Finance 2006)* for more details on each of the distributions, what they mean, how to select the right distribution to apply and so forth, or refer to the user manual for the Risk Simulator software, available on the website at www.realoptionsvaluation.com under the "Download" section.



Data Fitting

The fourth approach is that of data fitting **[I]**. Sometimes, you may not know what probability distribution is appropriate for the variable you wish to simulate. If you have existing data, you can apply the data fitting routine to determine the best distribution to use. First choose if the data should be fitted to a continuous (e.g., 1.235, -12.23, etc) or discrete distribution (e.g., -1, 20, 300, etc) **[W]** and where the data resides (select from a data field previously mapped, upload the data text file or copy, versus manually enter in or paste the data directly) **[X]**. Click Finish when you are done.

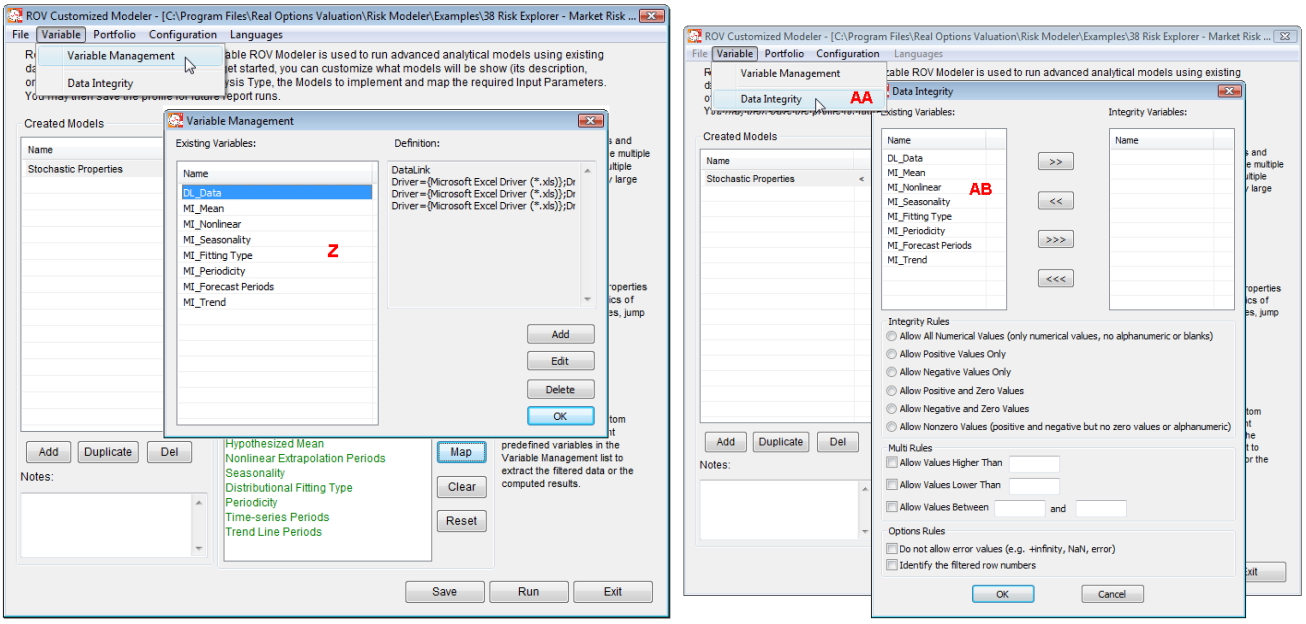


TIP: Variable Management

As a power user of the ROV Modeler software, the Variable Management tool is indispensable. You can click on the menu item Variable and select Variable Management **[Y]** to show the list of previously mapped variables **[Z]**. Using this approach, you can Add, Edit or Delete any existing variables. The power of this variable management is evident in the Data Compute examples above, where you can link in as many variables as you wish from a data set or database and then perform subsequent manipulations as desired. By using this combination of data linking, data variable management and data compute, you can essentially control the sequence of events and manipulate the data as required, before they are used in the model.

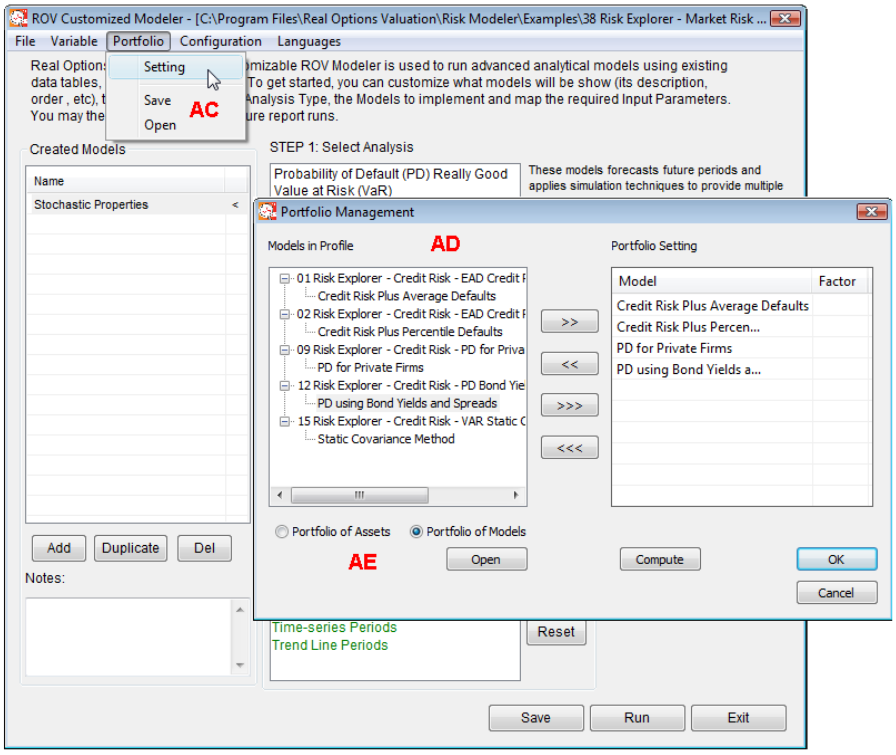
In addition, there is also a Data Integrity **[AA]** tool where you can filter your linked data by allowing only positive values, negative values, values higher or lower than a specific value, and so forth. You can do so by selecting from a list of previously mapped variables **[AB]** and adding them to the integrity variables list and selecting the rules you desire.

Finally, do not forget that you can perform even more advanced data filtering methods using SQL commands when you perform the data linking approach **[N]**. For more details on using the SQL commands, please refer to the appendix for some examples of practical uses.



TIP: Running Multiple Models using ROV Portfolio

In ROV Modeler, you can create multiple models and save all of them as a single file (we call these files profiles). When you need to run a model, you would select the model from the list of created models and run one model at a time. Sometimes, you may have created multiple models in multiple profiles and you would like to run all of them together. This can be accomplished using the ROV Portfolio tool **[AC].** You can access this tool through the Start, Programs, Real Options Valuation, and ROV Modeler shortcut or from the ROV Modeler menu item by selecting Portfolio and clicking on Setting **[AC]**. The Portfolio Management user interface appears and from here you can click Open to open any saved profiles. The models in all of the profiles will then be listed **[AD]**. You can then select the models you wish to run and click on the >> to add the model, or >>> to add all models, versus << to delete a selected model or <<< to delete all selected models. Then you can select if the models are run independently by choosing Portfolio of Models or run as a portfolio with different weights by choosing Portfolio of Assets and entering a Factor value for each model **[AE]**. For instance, if you have 5 models you need computed, and want the 5 results individually, but all 5 models are run at once, select the portfolio of models approach. Alternatively, if you have 5 assets' returns and you wish to compute the total returns of the portfolio and each asset has a different weighting (they all add to 100% or 1.0), select the portfolio of assets approach and enter in the relevant weighting for each asset.



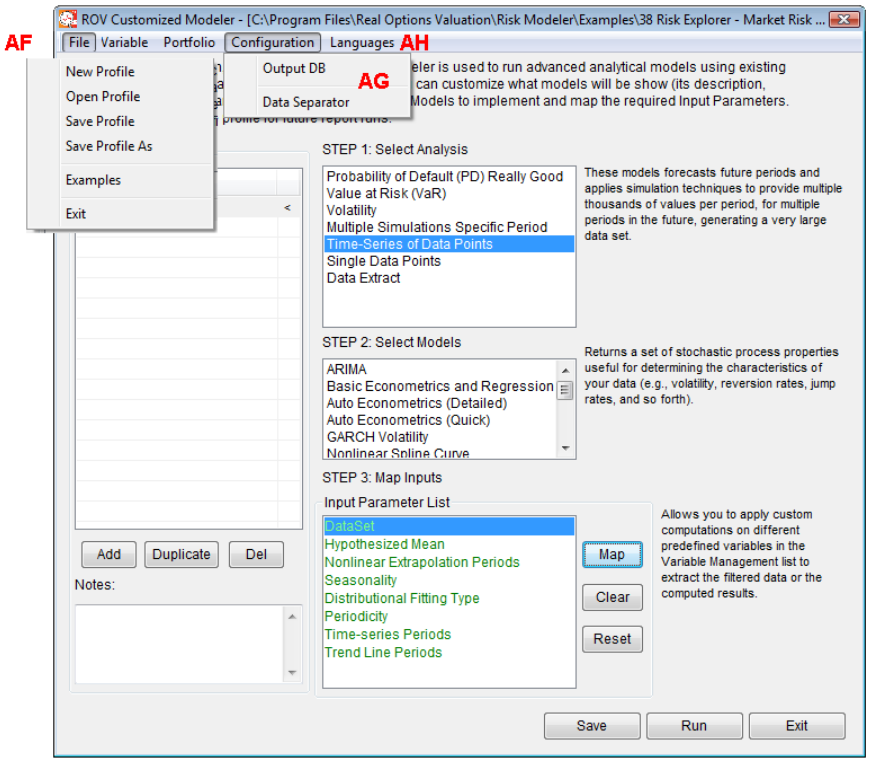
TIP: Saving Profiles, Data Separators, Output Databases and Languages

In ROV Modeler, you can click on the File **[AF]** menu item to save or open a profile. Remember that a profile is simply an XML file that contains all the data variables, links, manual data computations, location of linked data source files, models used, parameters, variable list, and everything that is done in ROV Modeler.

You can also open the Examples here. There are approximately 100 example profiles that come with ROV Modeler and you can open and use them as examples on getting started and learning how to use the software. Some of the models are specific to different industries and applications such as financial services, banks, investments, corporate finance, forecasting, project selection, portfolio optimization, and many others. It is beyond the scope of this document to go into the technical details of each model. Please refer to "Advanced Analytical Models: Over 800 Models and 300 Applications from Basel II to Wall Street and Beyond" by Dr. Johnathan Mun (Wiley 2008) for more details on each of these models.

In addition, the Configuration menu [AG] allows you to define the output database such that the results are loaded directly into a database. In addition, you can access the Data Separator menu here, to define the item separator and line separator as discussed previously.

Finally, there is a Languages [AH] menu where you can access different languages such as English, Chinese, Japanese, Spanish, French, German, Russian, Italian and others. Please be aware that most of the words will be translated whereas some technical terms are not (this is because the origin of these technical terms is English and there sometimes are no good direct translations in other languages) and these terms are kept in English on purpose.

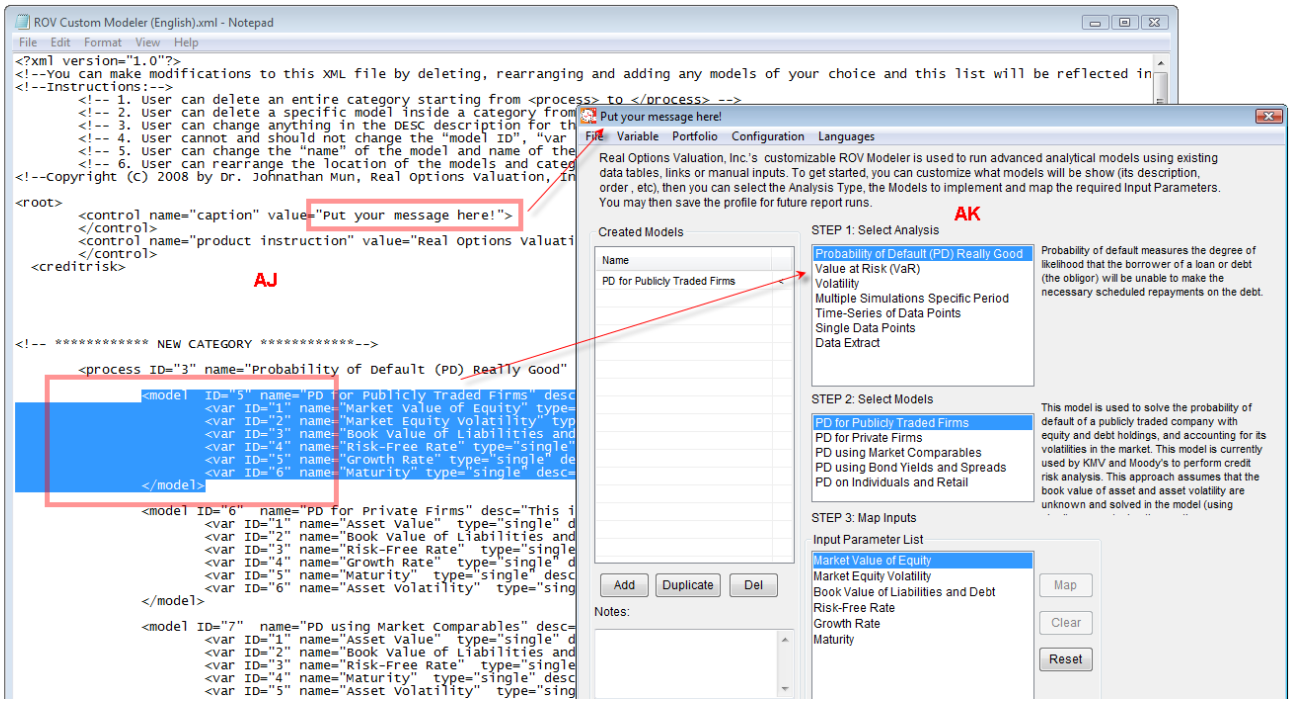


TIP: Customizing the ROV Modeler User Interface and List of Functions and Models

In the installation path of the software (e.g., c:\program files\real options valuation\risk modeler) there should be a file named "ROV Custom Modeler.xml" **[AI]**. Different languages have different XML files, e.g., the English version will be named "ROV Custom Modeler (English).xml". This XML file controls the user interface names and desctiptions. You can edit this file directly using an XML editor or using Notepad (Start, Programs, Accessories, Notepad, and then drag the XML file and drop it into Notepad to edit it) **[AJ]**. In the XML file, there are several things you can do, including:

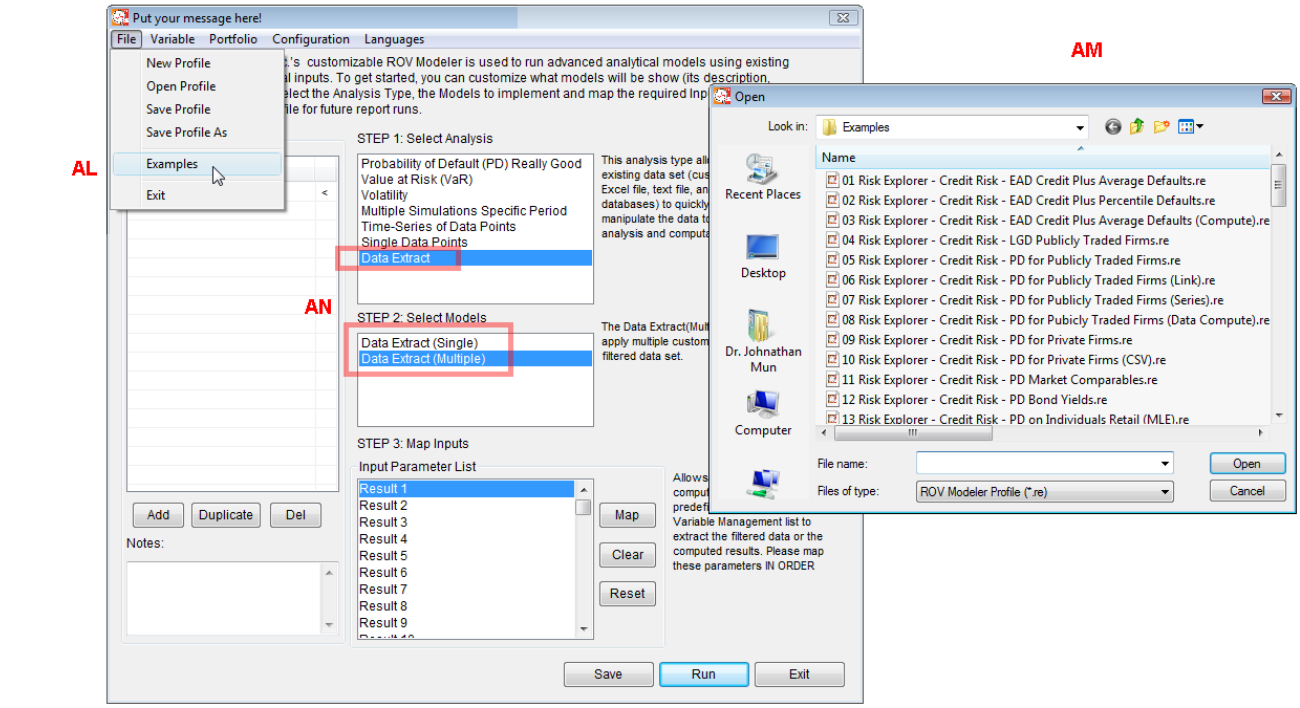
- User can delete an entire category starting from <process> to </process>.
- User can delete a specific model inside a category from <model> to </model>.
- User can change anything in the DESC description for the model.
- User cannot and should not change the "model ID", "var ID", "type" and "param_style" values.
- User can change the "name" of the model and name of the parameters as long as the ID values are not changed.
- User can rearrange the location of the models and categories to have certain models and categories appear first or appear later.

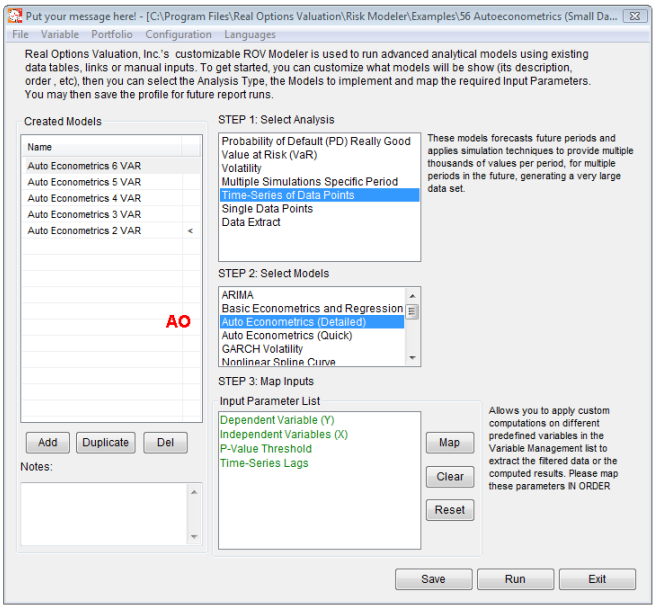
When you make the changes to the XML file and save it **[AJ]**, then start the ROV Modeler, you will see the effects **[AK]**.

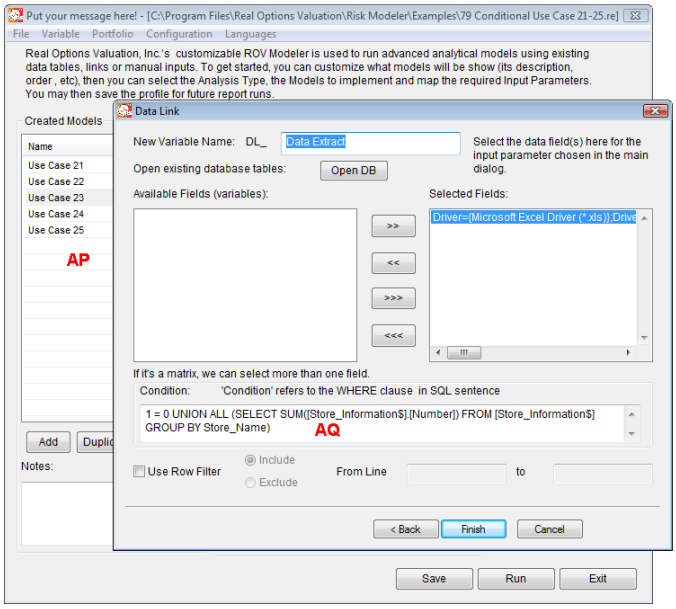


TIP: Example Models and Advanced Functionality

One good way to get started learning the software is to go into the Example profiles **[AM]**. There are multiple profiles available by clicking on the File menu and choosing Examples **[AL]**. There are several key functions that are probably most commonly used, including the Data Extract **[AN]** model (useful for extracting data directly from the data files or database, perform any computations as required, and returning the analysis results), Autoeconometrics model **[AO]** (available under the Time Series of Data Points model type, useful for performing up to hundreds of thousands of econometric-combinatorics models to find the best-fitting model for your data), and Use Case **[AP]** examples to learn how to use SQL functions **[AQ]** (see the appendix for more examples on how to use these SQL function calls).


Put your message here!
File Variable Portfolio Configuration Languages
New Profile
Open Profile
Save Profile
Save Profile As
Examples
Exit
AL
AM
AN
STEP 1: Select Analysis
Probability of Default (PD) Really Good
Value at Risk (VaR)
Volatility
Multiple Simulations Specific Period
Time-Series of Data Points
Single Data Points
Data Extract
STEP 2: Select Models
Data Extract (Single)
Data Extract (Multiple)
STEP 3: Map Inputs
Input Parameter List
Result 1
Result 2
Result 3
Result 4
Result 5
Result 6
Result 7
Result 8
Result 9
Add
Duplicate
Del
Map
Clear
Reset
Notes:
Save
Run
Exit
Open
Look in: Examples
Recent Places
Desktop
Dr. Johnathan Mun
Computer
Name
01 Risk Explorer - Credit Risk - EAD Credit Plus Average Defaults.re
02 Risk Explorer - Credit Risk - EAD Credit Plus Percentile Defaults.re
03 Risk Explorer - Credit Risk - EAD Credit Plus Average Defaults (Compute).re
04 Risk Explorer - Credit Risk - LGD Publicly Traded Firms.re
05 Risk Explorer - Credit Risk - PD for Publicly Traded Firms.re
06 Risk Explorer - Credit Risk - PD for Publicly Traded Firms (Link).re
07 Risk Explorer - Credit Risk - PD for Publicly Traded Firms (Series).re
08 Risk Explorer - Credit Risk - PD for Pubicly Traded Firms (Data Compute).re
09 Risk Explorer - Credit Risk - PD for Private Firms.re
10 Risk Explorer - Credit Risk - PD for Private Firms (CSV).re
11 Risk Explorer - Credit Risk - PD Market Comparables.re
12 Risk Explorer - Credit Risk - PD Bond Yields.re
13 Risk Explorer - Credit Risk - PD on Individuals Retail (MLE).re
File name:
Files of type: ROV Modeler Profile (*.re)
Open
Cancel

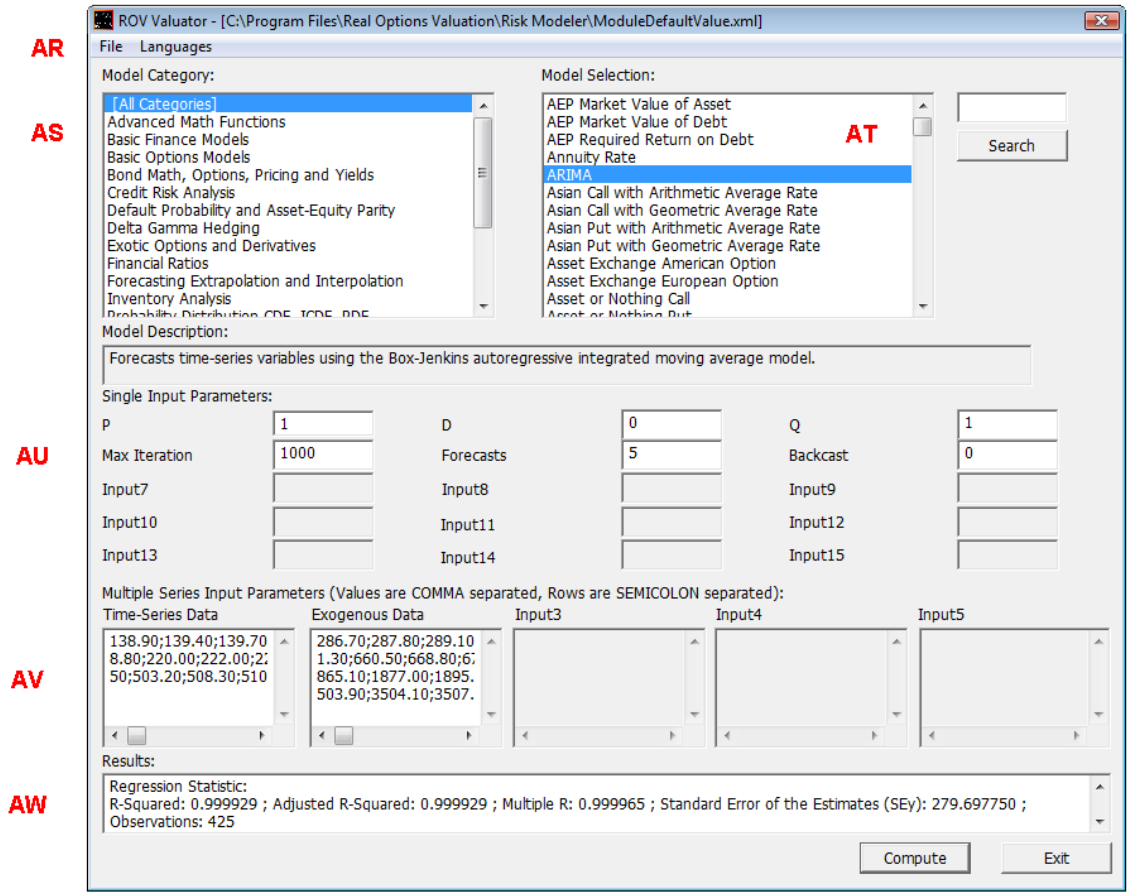

Put your message here! - [C:\Program Files\Real Options Valuation\Risk Modeler\Examples\56 Autoeconometrics (Small Da...
File Variable Portfolio Configuration Languages
Real Options Valuation, Inc.'s customizable ROV Modeler is used to run advanced analytical models using existing data tables, links or manual inputs. To get started, you can customize what models will be show (its description, order , etc), then you can select the Analysis Type, the Models to implement and map the required Input Parameters. You may then save the profile for future report runs.
Created Models
Name
Auto Econometrics 6 VAR
Auto Econometrics 5 VAR
Auto Econometrics 4 VAR
Auto Econometrics 3 VAR
Auto Econometrics 2 VAR
AO
STEP 1: Select Analysis
Probability of Default (PD) Really Good
Value at Risk (VaR)
Volatility
Multiple Simulations Specific Period
Time-Series of Data Points
Single Data Points
Data Extract
These models forecasts future periods and applies simulation techniques to provide multiple thousands of values per period, for multiple periods in the future, generating a very large data set.
STEP 2: Select Models
ARIMA
Basic Econometrics and Regression
Auto Econometrics (Detailed)
Auto Econometrics (Quick)
GARCH Volatility
Nonlinear Spline Curve
STEP 3: Map Inputs
Input Parameter List
Dependent Variable (Y)
Independent Variables (X)
P-Value Threshold
Time-Series Lags
Allows you to apply custom computations on different predefined variables in the Variable Management list to extract the filtered data or the computed results. Please map these parameters IN ORDER
Add
Duplicate
Del
Map
Clear
Reset
Notes:
Save
Run
Exit


Put your message here! - [C:\Program Files\Real Options Valuation\Risk Modeler\Examples\79 Conditional Use Case 21-25.re]
File Variable Portfolio Configuration Languages
Real Options Valuation, Inc.'s customizable ROV Modeler is used to run advanced analytical models using existing data tables, links or manual inputs. To get started, you can customize what models will be show (its description, order , etc), then you can select the Analysis Type, the Models to implement and map the required Input Parameters. You may then save the profile for future report runs.
Created Models
Name
Use Case 21
Use Case 22
Use Case 23
Use Case 24
Use Case 25
AP
Data Link
New Variable Name: DL_ Data Extract
Select the data field(s) here for the input parameter chosen in the main dialog.
Open existing database tables:
Open DB
Available Fields (variables):
Selected Fields:
Driver={Microsoft Excel Driver (*.xls)};Drive
>>
<<
>>>
<<<
If it's a matrix, we can select more than one field.
Condition: 'Condition' refers to the WHERE clause in SQL sentence
1 = 0 UNION ALL (SELECT SUM([Store_Information$].[Number]) FROM [Store_Information$] GROUP BY Store_Name)
AQ
Use Row Filter
Include
Exclude
From Line
to
< Back
Finish
Cancel
Add
Notes:
Save
Run
Exit

Risk Valuator

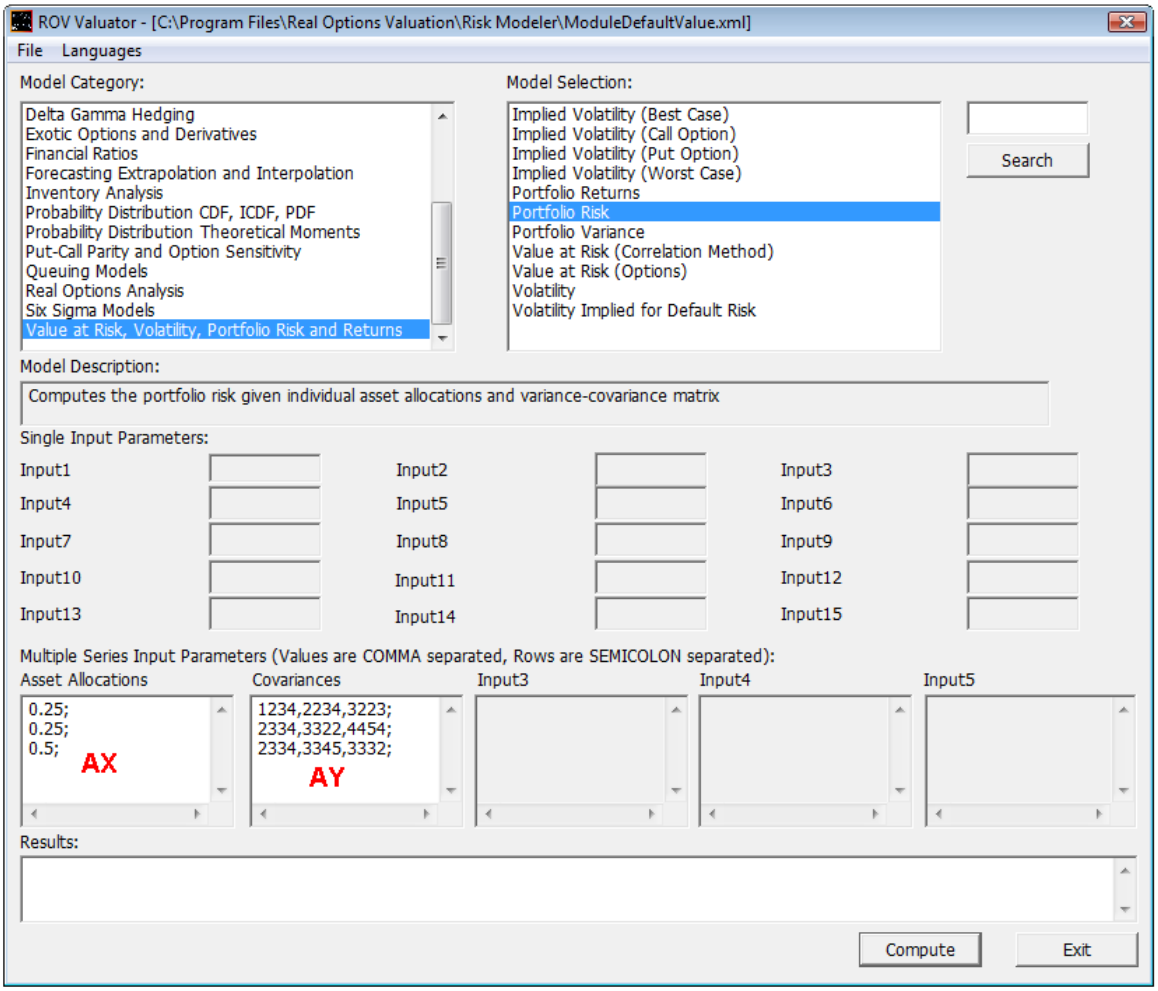
Risk Valuator is the application of over 600+ advanced analytical functions. It affords hundreds of models in different categories for the user to select. User can input the required data for the selected model and this application will return the computed results very quickly. This module is useful for valuing derivative instruments, debt instruments, exotic options, options-embedded instruments, as well as multiple types of financial models. The 600+ advanced models are categorized into the following groups of applications:

- Advanced Math Functions
- Basic Finance Models
- Basic Options Models
- Bond Math, Options, Pricing, and Yields
- Credit Risk Analysis
- Delta Gamma Hedging
- Exotic Options and Derivatives
- Financial Ratios
- Forecasting, Extrapolation and Interpolation
- Probability Distributions
- Put-Call Parity and Option Sensitivities
- Real Optfions Analysis
- Value at Risk, Volatility, Portfolio Risk and Return



Risk Valuator **[AR]** is used to perform quick computations from simple and basic models to advanced analytical models, and can handle single point values or a series of values. After installing the software, start Risk Valuator. Simply select the model type in the Model Category **[AS]** box and select the model of interest in the Model Selection **[AT]** box. The required input parameters will then be listed. Single point inputs (e.g., 10 or 10.4532) will be in the single input parameters area **[AU]**, whereas multiple data requirements will be shown in the multiple series input parameters area **[AV]**. When entering a single series of multiple data points, use commas or spaces to separate the values (e.g., a time-series of 6 months of interest rates can be entered either as 0.12, 0.124, 0.112, 0.1, 0.09, 0.16 or simply as 0.12 0.124 0.112 0.1 0.09 0.16). Hit COMPUTE and the analysis is run and the results are returned **[AW]**.

Sometimes, certain models such as the Value at Risk model using the standard correlation method, requires different columns of data and a correlation matrix. For instance, the goal is to compute the portfolio VaR using this model, where there are 3 asset classes, each with its own amounts, specific daily volatility for each asset class, and a square correlation matrix among these asset classes. In such a situation, the amounts and volatility inputs will have to be entered as a single column (hit ENTER at the end of entering a value, to create a new line, designating a new asset class or use the semi-colon as a line separator **[AX]**) and the correlation matrix will be separated by commas for the same row with different columns, and semi-colons for different rows **[AY]**. This Risk Valuator module does not allow the user to link to various databases or simulate. To do so, use the ROV Risk Modeler modules instead. Many of the same models exist in both places. The ROV Risk Modeler module is used to quickly obtain results without having to link to databases and so forth.



To get started learning how to use this tool, click on File menu and select Load Sample Inputs. Then, select a model category and select a model of interest. You will see the sample inputs loaded. You can then click on Compute to obtain the results. Use these sample inputs as a guide to get started with your modeling needs.

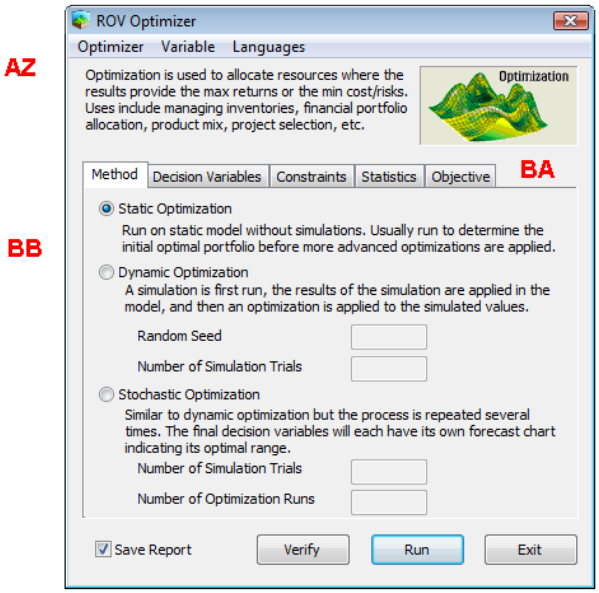
Just like in ROV Modeler, you can customize the list of models that appear in the ROV Valuator, as well as the descriptions for each model. Simply go to the installation path (e.g., c:\program files\real options valuation\risk modeler) and look for files "ROV Custom Valuator (English).xml" and select the correct files depending on your language. This XML file controls the user interface names and desctiptions. You can edit this file directly using an XML editor or using Notepad (Start, Programs, Accessories, Notepad, and then drag the XML file and drop it into Notepad to edit it). In the XML file, there are several things you can do, including:

1. You can delete an entire category starting from <category> to </category>
2. You can delete a specific function inside a category from <function> to </function>
3. You can change anything in the "category name", "displayname" and "desc" description for the model
4. You cannot and should not change the "function name", "type" and "param_type" values
5. You can but should not change the "var name" of the model (you run the risk that sample values loaded might not have a valid value)
6. You can rearrange the location of the models and categories to have certain models and categories appear first or appear later
7. Instead of deleting models, try commenting them out instead using the "open triangular bracket, apostrophe and two dashes" and "two dashes and close triangular bracket" such that if you need the models again later, they are available
8. You can also create your own category of models using the examples in this document, with your own favorite list of models...

Risk Optimizer

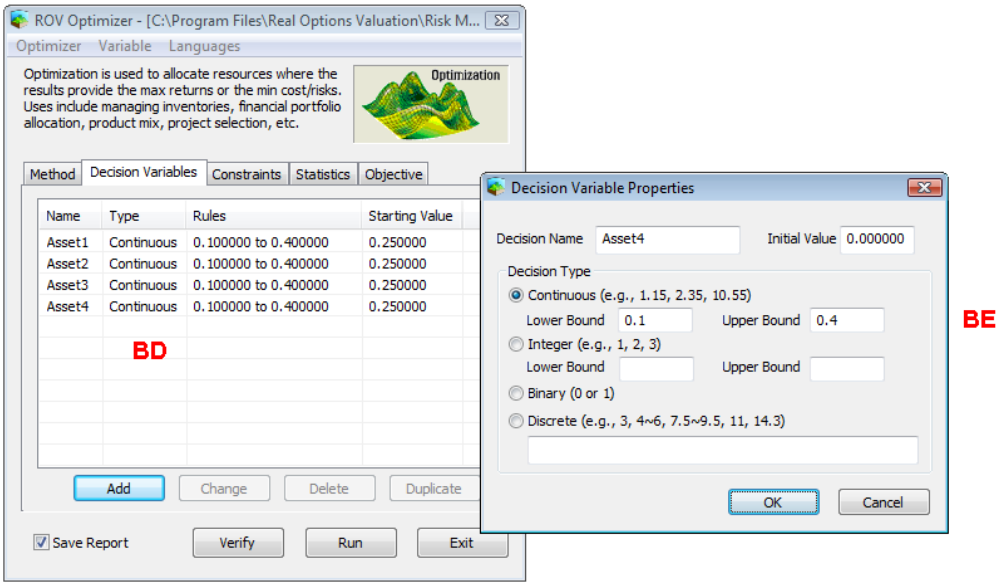
Risk Optimizer is an advanced optimization module that can be used to optimize portfolios and to find optimal investment decisions and optimal project selections for a corporation, a bank, an investment firm, a manufacturer, an R&D outfit, and many others. The decision variables can be discrete, continuous, integer, or binary, and the objective function can be linear or nonlinear. In addition, Risk Optimizer allows the user to link to existing data tables to run simulations, find the best-fitting models, and couple these techniques with optimization. The technical details of optimization fall outside the scope of this document. For more details and examples, please see *"Modeling Risk: Applying Monte Carlo Simulation, Real Options Analysis, Stochastic Forecasting, and Portfolio Optimization" by Dr. Johnathan Mun (Wiley Finance 2006)*.

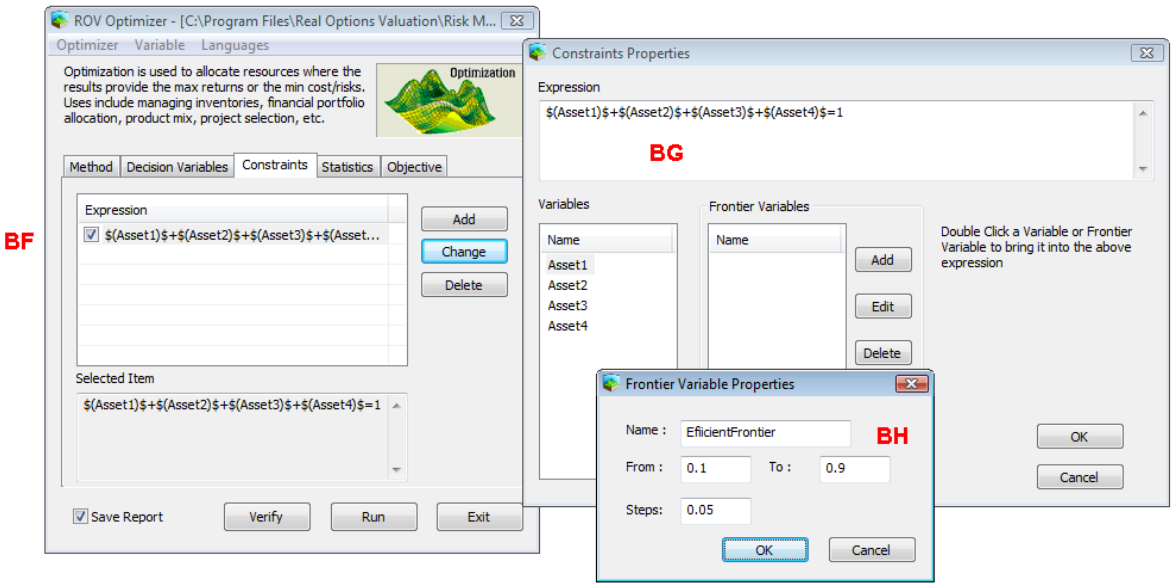
Here is a simple example on how to use the ROV Optimizer **[AZ]** (we also suggest you click on the File menu and select Examples to load some predefined models to learn how the models can be set up). When you install the Risk Optimizer, you can open and see the UI of the software. "Method", "Decision Variables", "Constraints" will show in front of the user. Choose the "Method" **[BA]** tab and select "Static Optimization" **[BB]**. Again, for details on the differences among static, dynamic and stochastic optimization, please contact our technical support department, review Dr. Mun's book above, or attend one of Real Options Valuation, Inc.'s training seminars.



Next, click on the Decision Variables tab **[BA]** and hit ADD to add some variables. For instance, we have 4 different variables **[BD]** (Asset1 to Asset4), and each asset can be set to take continuous, integer, binary, or discrete values **[BE]**. For our simple illustration, set the variables to all be Continuous between 0.10 and 0.40 (i.e., only asset allocations between 10% and 40% are allowed). Keep adding 4 different asset classes as decision variables.

Next, click on the Constraints tab and select ADD **[BF]**. Then, in the expressions input box, enter in the constraint (you can double click on the list of variables and the variable string will be transferred up to the expressions box). In our simple example, the total decision variable values must sum to 1.0 (i.e., the total allocation of asset classes must total 100% in an investment portfolio) **[BG]**. You can also create an Efficient Frontier by adding the Frontier Variables **[BH]**. Again, for details on efficient frontiers, review the modeling risk book by Dr. Mun.

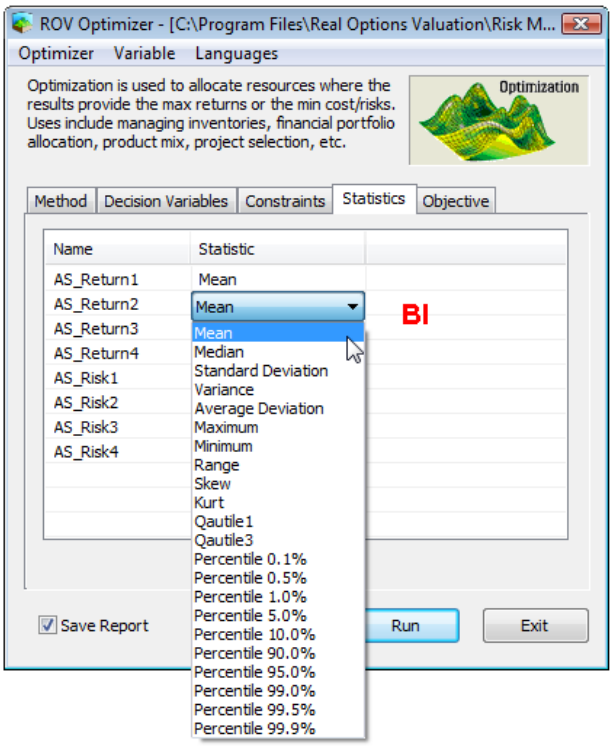

ROV Optimizer - [C:\Program Files\Real Options Valuation\Risk M...
Optimizer Variable Languages
Optimization is used to allocate resources where the results provide the max returns or the min cost/risks. Uses include managing inventories, financial portfolio allocation, product mix, project selection, etc.
Optimization
Method Decision Variables Constraints Statistics Objective
Name Type Rules Starting Value
Asset1 Continuous 0.100000 to 0.400000 0.250000
Asset2 Continuous 0.100000 to 0.400000 0.250000
Asset3 Continuous 0.100000 to 0.400000 0.250000
Asset4 Continuous 0.100000 to 0.400000 0.250000
BD
Add Change Delete Duplicate
Save Report Verify Run Exit
Decision Variable Properties
Decision Name Asset4 Initial Value 0.000000
Decision Type
Continuous (e.g., 1.15, 2.35, 10.55)
Lower Bound 0.1 Upper Bound 0.4
Integer (e.g., 1, 2, 3)
Lower Bound Upper Bound
Binary (0 or 1)
Discrete (e.g., 3, 4~6, 7.5~9.5, 11, 14.3)
OK Cancel
BE

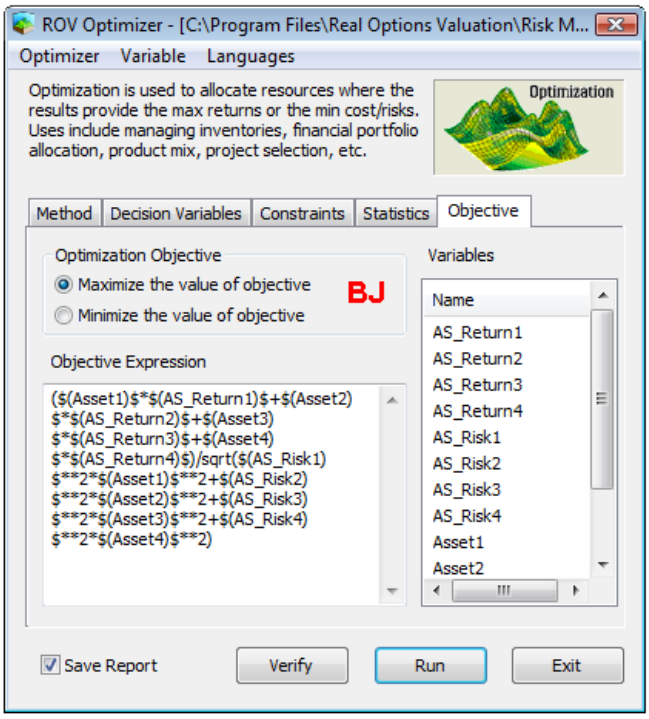

BF
ROV Optimizer - [C:\Program Files\Real Options Valuation\Risk M...
Optimizer Variable Languages
Optimization is used to allocate resources where the results provide the max returns or the min cost/risks. Uses include managing inventories, financial portfolio allocation, product mix, project selection, etc.
Optimization
Method Decision Variables Constraints Statistics Objective
Expression
$(Asset1)$+$(Asset2)$+$(Asset3)$+$(Asset...
Add Change Delete
Selected Item
$(Asset1)$+$(Asset2)$+$(Asset3)$+$(Asset4)$=1
Save Report Verify Run Exit
Constraints Properties
Expression
$(Asset1)$+$(Asset2)$+$(Asset3)$+$(Asset4)$=1
BG
Variables
Name
Asset1
Asset2
Asset3
Asset4
Frontier Variables
Name
Add Edit Delete
Double Click a Variable or Frontier Variable to bring it into the above expression
OK Cancel
Frontier Variable Properties
Name : EficientFrontier
BH
From : 0.1 To : 0.9
Steps: 0.05
OK Cancel

In addition, if you are using static optimization, you can skip the Statistics tab, whereas the statistics tab is important when you are running a dynamic or stochastic optimization, when some of the variables are mapped to probability distributions and simulations will be run before and after the optimization **[BI]**.

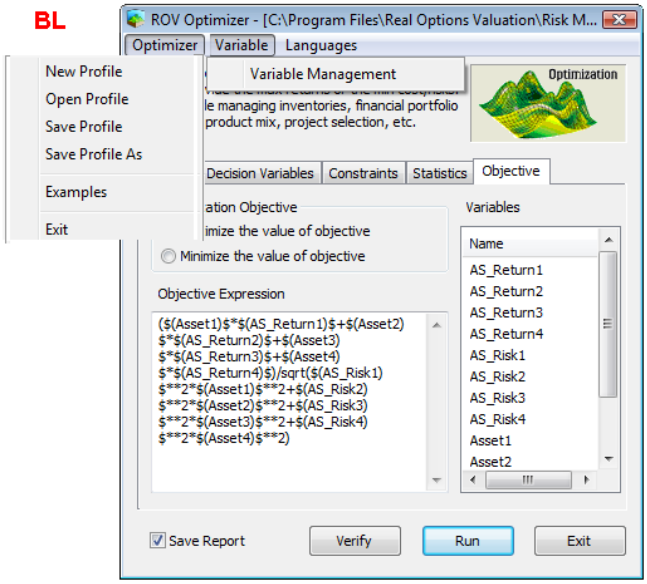
Next, select the Objective tab **[BJ]** and select if you wish to run Maximization or Minimization on your objective. In addition, enter in the relevant objective expression as outlined below. You can double click on the list of Variables to bring the variable name string to the objective expression input box. When completed, click on RUN to obtain the results of your optimization, or you can first click on Verify to test if the model has been set up correctly.

The optimization results **[BK]** will appear if the optimization model is set up correctly. The results will show the number of iterations, the specific model configuration, the parameters, the intial and optimized results of the objective and decision variables, the technical analytics (Lagrange multipliers, Hessian matrices and others), and an optimization objective chart.





There are also two important functionalities in ROV Optimizer, available in the File menu, including Examples and Variable Management under the Variable menu item **[BL]**. The Variable Management tool allows you to Add, Edit or Delete variables. For instance, by clicking on ADD, the familiar Input Parameter Mapping tool appears, allowing you to link, compute, paste, simulate, or fit existing data for use in the optimization process. Finally, if Dynamic or Stochastic Optimization is selected, and if the variables have risk simulation assumptions associated with them, you can then access the Statistics tab, whereby you can make use of the simulated statistical properties to run optimization on.



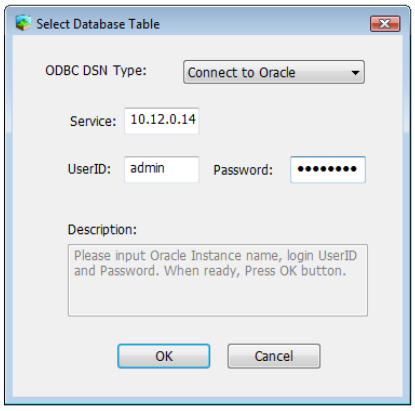
Linking to Other Databases

ROV Risk Modeler can link to different types of data using ODBC standard. When mapping to the database, you can select the Data Link in the method input. Click on Next and type in a name in the New Variable Name. Then select Open DB to open a database type and you can select your data source in terms of different data types that ROV Risk Modeler can connect to, including CSV, Excel, SQL Server, Oracle, User DSN, System DSN, and Connection Strings, with the ODBC data source standard.

Case One: Link to Oracle

When you choose ODBC DSN as Connect to Oracle, input the local IP address of the Database Server and the relevant User ID and Password to log in **[BM]**. You can then find the available fields (variables) which can be selected. You can also write SQL sentences in the Condition box until the right variables and values are linked to the ROV Risk Modeler. It is important to notice that the database compontents must be Oracle version 7.3 or higher.

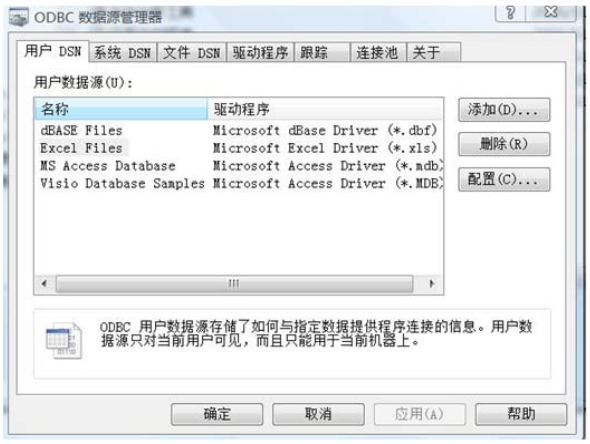
BM



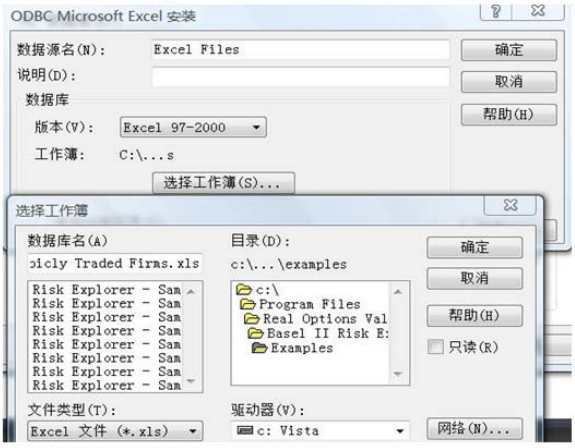
Case Two: Link to User DSN

Before you choose ODBC DSN as User DSN you must set the DSN to a certain file in the first step. Click Start, select Control Panel and select Management Tools where you can see the Data Source (ODBC) selection **[BN]**. Choose the User DSN tag and click Excel Files and then click Configuration. In the new dialog, click on Choose Workshop, and find an existing Excel file, click OK **[BO]**. You can now return to the ROV Risk Modeler, map a variable using Data Link, click on Open DB and User DSN, choose Excel Files, a list of tables will be listed. You can now map the existing table data to the Selected Fields.

BN

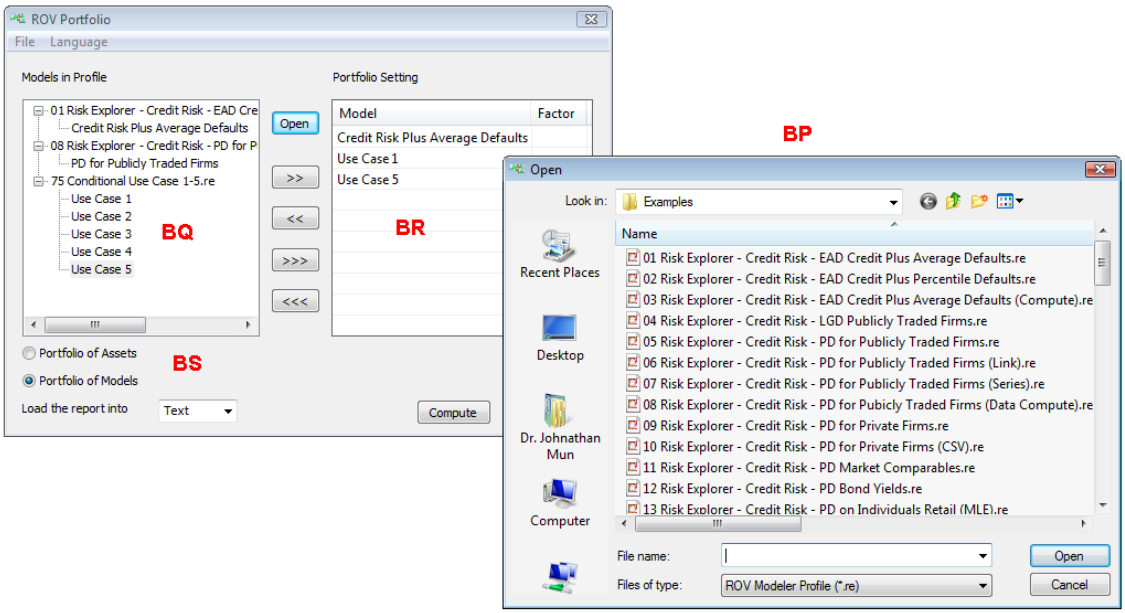


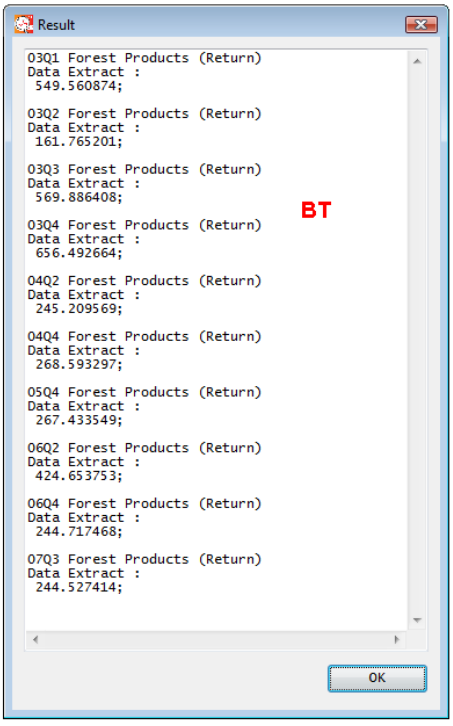
BO



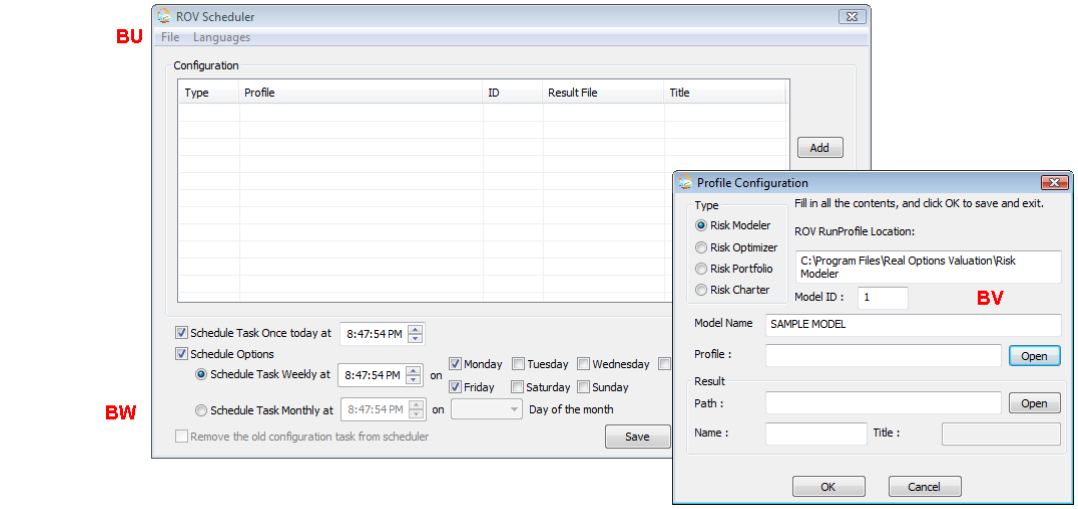
ROV Scheduler, ROV Portfolio, ROV Charter

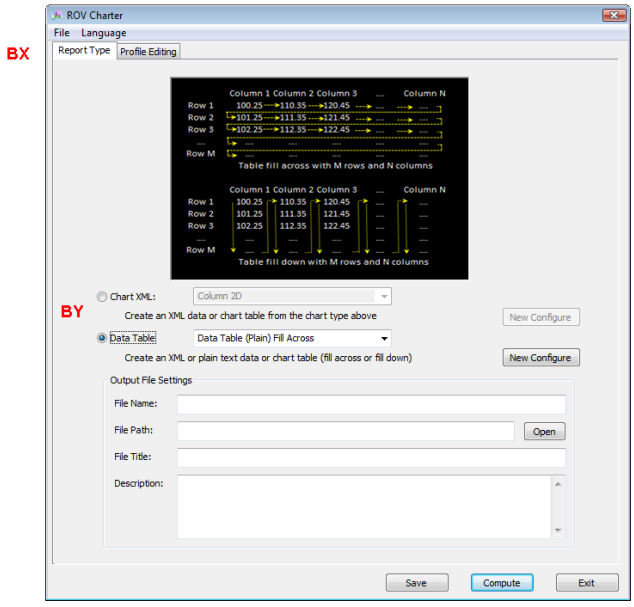
Earlier in this document we briefly illustrated the use of ROV Portfolio, ROV Scheduler and ROV Charter. The ROV Portfolio is used to run multiple models at once from a single profile or different profiles **[BP]**. You can click on Open to open different profiles **[BQ]** where you can open and select the models you wish to run **[BR]** and decide if you wish to run all the models as independent models (portfolio of models) or as dependent models in a portfolio (portfolio of assets) **[BS]** where you can add different weighting Factors for each model. When the portfolio of models is run, the results will be a list of values **[BT]**. These portfolios can be saved. This means that a single portfolio can contain multiple models and multiple profiles.

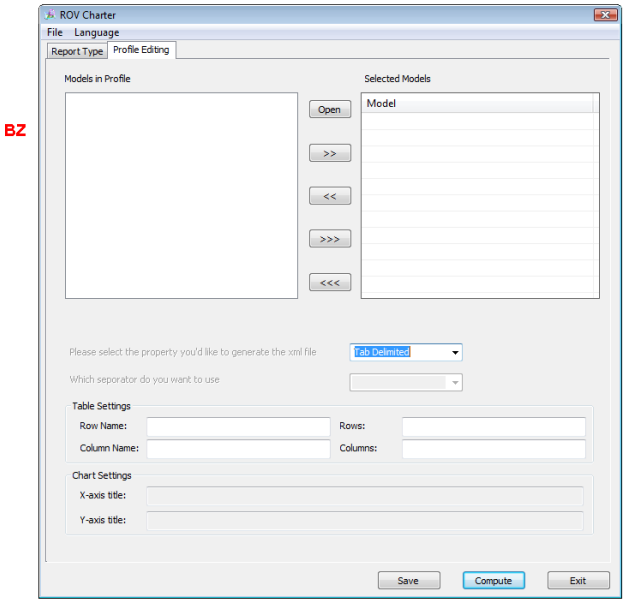




Similarly, you can set up the ROV Scheduler **[BU]** to run multiple models, portfolios from different profiles **[BV]**, just like in ROV Portfolio. The difference is that you can now schedule when these models are set to run **[BW]**. Finally, there is another ROV Charter **[BX]** tool where you can run multiple models from different profiles **[BZ]** and set the results as XML files **[BY]** that can be run in ROV Dashboard as charts or tables in a web-based environment, another software tool developed by Real Options Valuation, Inc.







Running XML without the User Interface

Sometimes there is a situation where you would like to run the ROV Modeler's XML profiles directly without using ROV Risk Modeler, Risk Valuator and Risk Optimizer user interface. You can regard this approach as an SDK and the approach is akin to integrating the SDK to your existing system. Using ROV Modeler and ROV Optimizer without the user interface is very simple because all of the model-logics are combined into the ModelLgc.dll file and it is applicable in a C-like interface:

int RunProfile (std::wstring& pszWorkDir, std::wstring& pszProfile, std::wstring& pszTitle, int nModID, int nType, std::wstring& pszFileName)

For now, it supports three types of profiles:
RiskModel profile, nType=0
Optimization profile nType=1
Portfolio profile nType=2

The required header files of the ModelLgc.dll, including three ROV header files: ModelLgc.h, ModelHandle.h and ResultSet.h; and the related library headers, includes STL, GLib, LibXML. Below is the sample code to call the ModelLgc.dll and generating the report.

```
// sample.cpp: Defines the entry point for the console application.
 //Step1 adjust project setting, adding include directories and dependencies:
//Below is additional include directories:
//"G:\RSCPP\Codes\ROV Risk Modeler/LibExtra/libxml++-2.18.2";
//"G:\RSCPP\Codes\ROV Risk Modeler/LibExtra/libxml++-2.18.2/MSVC_Net2003/libxml++";
//"G:\RSCPP\Codes\ROV Risk Modeler/LibExtra/glibmm-2.14.2/glib";
//"G:\RSCPP\Codes\ROV Risk Modeler/LibExtra/glibmm-2.14.2/glib/glibmm";
//"G:\RSCPP\Codes\ROV Risk Modeler/LibExtra/glibmm-2.14.2/MSVC_NET2003/glibmm";
//"G:\RSCPP\Codes\ROV Risk Modeler/LibExtra/glib-2.14.3";
//G:\RSCPP\Codes\Include
//Below is additional dependencies:
//ModelLgc.lib
//Step2 include the ModelLgc.h header

#include "stdafx.h"
#include <string>
#include <iostream>
#include "G://RSCPP//Codes//ROV Risk Modeler//ModelLgc//Modellgc.h"

using std::wstring;

//Step3 call RunProfile method to compute profile
int callRiskModelProfile()
{
 //Location of configuration files where contains the sepcfg.xml file
      wstring s1(L"G://RSCPP//Profiles");

      //Absolute path of profile
      wstring s2(L"G://RSCPP//Profiles//Examples//01 ROV Risk Modeler - Credit Risk - EAD Credit Plus Average Defaults.re");
      //No use
      wstring s3(L"");
      //Location of output file
      wstring s4(L"G://output.txt");
      //The model's ID in the profile
      int nMode = 1;
      //Run RiskModel type
      int nType = 0;
      int i = RunProfile(s1, s2, s3, nMode, nType, s4);
      //Return 0 means success, else failed
      std::cout<<i<<std::endl;
      return 0;
}
```

```
int callOptimizationProfile()
{
	//Location of configuration files
	wstring s1(L"G://RSCPP//Profiles");
	//Absolute path of profile
	wstring s2(L"G://RSCPP//Profiles//ROSample//ROSample//dis9.ro");
	//Title of report
	wstring s3(L"Example report!");
	//Location of output file
	wstring s4(L"G://output_ro.txt");
	//No use
	int nMode = 0;
	//Run Optimization type
	int nType = 1;
	int i = RunProfile(s1,s2,s3, nMode, nType,s4);
	std::cout<<i<<std::endl;
	return 0;
}

int callPortfolioProfile()
{
	//Location of configuration files
	wstring s1(L"G://RSCPP//Profiles");
	//Absolute path of profile
	wstring s2(L"G://RSCPP//Profiles//ROSample//ROSample//dis9.ro");
	//No use
	wstring s3(L"");
	//Location of output file
	wstring s4(L"G://output_ro.txt");
	//No use
	int nMode = 0;

	//Run Portfolio type
	int nType = 2;
	int i = RunProfile(s1,s2,s3, nMode, nType,s4);
	std::cout<<i<<std::endl;
	return 0;
}

int main()
{
	callRiskModelProfile();
	callOptimizationProfile();
}
```

Integration to Other Systems

ROV Risk Modeler is also a DLL SDK where you can integrate the SDK to your own system and perform further development. The Risk Modeler DLL comprises a lot of models and algorithms, and the data in the XML files can be loaded by each model. You can manually input the data or use the existing data or link to a database.When users load the Risk Modeler, the algorithms will parse the data in the XML file and return the computed results. You can use these results and output to a certain file and location. The following are examples of DLL calls:

- Load DLL: In the Windows system, you can load the Risk Modeler DLL using:
  LoadLibrary(_T("ModelLgc.dll")
- Make sure you have the correct input parameters in the model used. All of the input parameters are based on the XML file structure as follows. The number of inputs and parameter styles are dependent on different models and algorithms. The data value can be a single number, a variable, from Excel or a series of numbers in the database and it depends on the location of user data.

This is a simple XML structure which stores the model parameter:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
 <risktype type="credit"/>
 <model name="PD for Publicly Traded Firms" ID="5" process="Probability of Default (PD)" note="">
  <var name="MarketValueEquity" ID="1" datatype="">
   <map maptype="Manual" name="MI_1">
    <data name="MI_1" uniquevalue="2" text="3000"/>
   </map>
  </var>
  <var name="MarketEquityVolatility" ID="2" datatype="">
   <map maptype="Manual" name="MI_2">
    <data name="MI_2" uniquevalue="2" text="0.45"/>
   </map>
  </var>
  <var name="BookValueLiabilitiesandDebt" ID="3" datatype="">
   <map maptype="Manual" name="MI_3">
    <data name="MI_3" uniquevalue="2" text="10000"/>
   </map>
  </var>
  <var name="RiskFree" ID="4" datatype="">
   <map maptype="Manual" name="MI_4">
    <data name="MI_4" uniquevalue="2" text="0.05"/>
   </map>
  </var>
  <var name="GrowthRate" ID="5" datatype="">
   <map maptype="Manual" name="MI_5">
    <data name="MI_5" uniquevalue="2" text="0.07"/>
   </map>
  </var>
  <var name="Maturity" ID="6" datatype="">
   <map maptype="Manual" name="MI_6">
    <data name="MI_6" uniquevalue="2" text="1.00"/>
   </map>
  </var>
 </model>
 <simulation runsim="1" trials="500" seed="0" runstress="0" percent="0"/>
 <output_option radio="0"/>

 <variable>
 </variable>
</root>
```

Model is the algorithm and *VAR* is the required variable for the model. Simulation is some information for the simulation functions. Please note that the rules and nodes for the XML file cannot be changed as all of the Risk Modeler models are computed based on the data in the XML file. Otherwise, the Risk Modeler algorithms cannot load without the right number of parameter inputs.

- The structure of the model returns are as follows:

```
std::vector<ResultsSet*> *pResult;
template<class T>
struct CResultSet
{
    std::vector<CResultSet<T>*>* child;
    int type;  // double, string
    int setType;  // single, vector, vector<vector>
    int identity;
    bool bChild;
    std::vector<std::vector<T>*>* thisValue;
};
```

All of the models will run back to the pointer of *pResult* after calculating the input parameters from the XML. You can output these values to the appointed file and location.

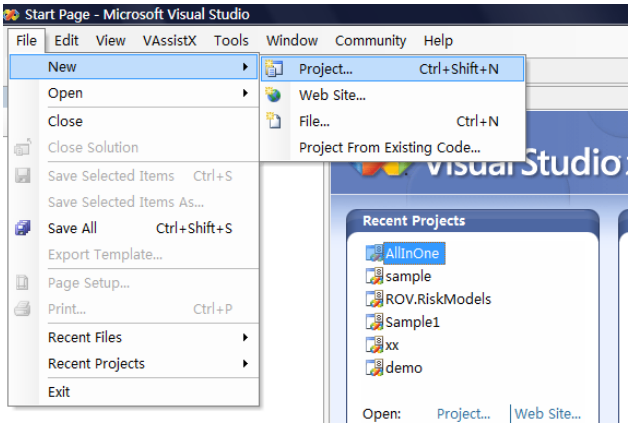
- Method of calling the models:

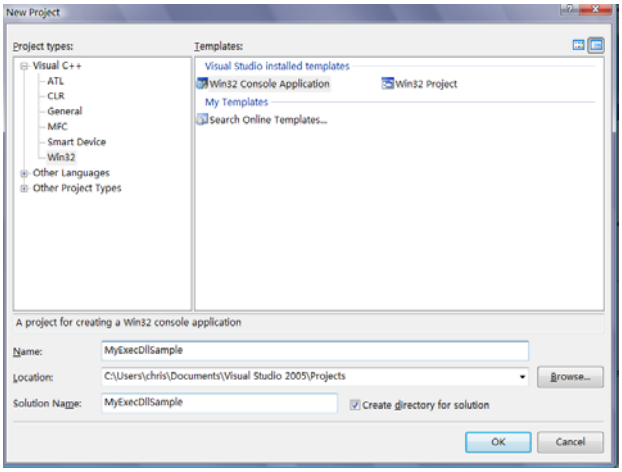
The following code demonstrates the data port and functions when calling the Risk Modeler algorithms in the Windows environment. The parameter *pszProfile* is the file name which stores the XML file and *nModID* is the ID number for each model.

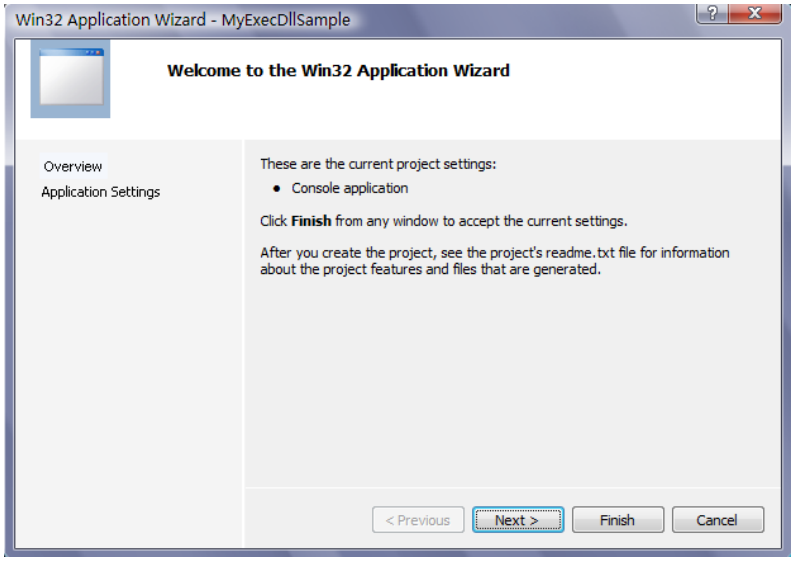
```
//define a function pointer。
typedef void* (*Compute)(wchar_t* pszProfile, int nModID);
//Cteate a function pointer case and point to Compute Model。
Compute pFunc = (Compute)GetProcAddress(m_hModelLgcDll, "ComputeModel");
if(pFunc != NULL)
  //Call ComputeModelMethod ：This will locate the needed model in terms of the nModID 该方法会根据 nModID
  pResult = pFunc(pszProfile, uID);
//Define a function pionter
typedef void (*FreeResult)(void* pResult);
//Create a function pointer case and point to the FreeResult ：Release the memory space used by the return value
FreeResult pFunc = (FreeResult)GetProcAddress(m_hModelLgcDll, "FreeResult");
if(pFunc != NULL)
pFunc((void*)pResult);
```

Case Examples

1. **Step One:**
   a. Create a new XML file called *PD.xml* and copy the xml content above to this xml and save the file
2. **Step Two:**
   a. Open Visual Studio 2005 and click on *File*, *New*, and *Project* to create a new project
   b. Choose *Win32 Console Application* and name the project as *MyExecDllSample*
   c. Click *OK* and setup *Win32 Application Wizard*. In the wizard dialog *Overview* page, click *Next*
   d. In the Win32 Application Wizard's *Application Settings* page, choose *Application Type* and Console Application
   e. In the Win32 Application Wizard's *Application Settings* page, disable the checkbox *Precompiled Header* in theAdditional Options section







3. **Step Three:** Use the following code to replace the existing code in the C++ file:

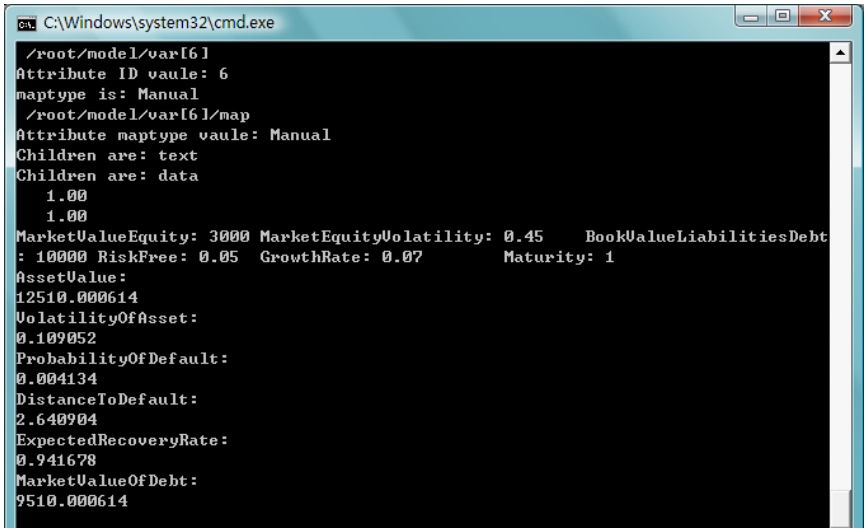
```
#include "stdafx.h"
#include "ResultSet.h"
#include <iostream>
#include <vector>
#include <string>
#include <stdio.h>
   int _tmain(int argc, _TCHAR* argv[])
   {
     //load the library and use the models to compute
     HINSTANCE hDll = LoadLibrary(_T("ModelLgc.dll"));
if(hDll == NULL)
   ::MessageBox(NULL, _T("Load calculation dll failed!"), _T("Error"), MB_OK);
     std::vector<RCBaseIII::ResultsSet*> *pResult = NULL;
     typedef void* (*Compute)(wchar_t* pzProfile, int nModID);
     Compute pFunc = (Compute)GetProcAddress(hDll, "ComputeModel");
     if(pFunc != NULL)
                    pResult = (std::vector<RCBaseIII::ResultsSet*> *)pFunc(_T("PD.xml"), 5);

                    //Save the structure which the pResult point:Output the Result
     string strOut;
     typedef std::vector<double> columns;
```

```
    for(int i=0; i<pResult->size();i++)
    {
                RCBaseIII::ResultsSet* pRet = pResult->at(i);
                strOut.append(pRet->name);
                strOut.append(":\n");
                char szData[32] = {0};
                std::vector<columns> *pValueTmp = (std::vector<columns>*)pRet->pValue;
                _snprintf(szData,sizeof(szData)/sizeof(szData[0])-1, "%lf", pValueTmp->at(0).at(0));
                strOut.append(szData);
                strOut.append("\r\n");
                }
                cout << strOut;

                //Release the return memory space
                typedef void (*FreeRes)(void* pResult);
                FreeRes pFunction = (FreeRes)GetProcAddress(hDll, "FreeResult");
                if(pFunction != NULL)
                        pFunction((void*)pResult);
                FreeLibrary(hDll);
    return 0;
}
Build>Build Solution，generate the EXE file
```

4. **Step Four**：Run the generated program
   a. Copy the *PD.xml* file to the same directory. Setup command and go to the directory, click the name of the program and run.
   b. If it does not compile or run, first check if the location of the SDK is in the compiler directory. Then, check if the XML file is in the same directory of the application program and try again.



APPENDIX: SQL USE CASES AND EXAMPLES

SQL Conditional Use Cases

The following are some common use cases in which a large data set can be screened, cleaned and filtered to return the required rows of data for analysis in Risk Modeler. Each use case shows a quick summary of the problem to be solved, accompanied by the XLS sample data file, Risk Modeler RE profile and model name, as well as some simple screen shots to illustrate the existing data set, the approach taken and the results. Clearly, the sample data set are intentionally kept small to facilitate the learning experience, but the same approaches and techniques illustrated in this use case document is applicable for all dataset sizes. Below is a quick summary of the key items in these use case examples:

- **Variable > Value** obtains rows above a specific threshold value
- **Variable > 80 AND Variable < 100** allows you to append with AND to create multiple filters
- **Variable < 80 OR Variable > 100** allows you to select data with OR condition filters
- **(Variable > 80 AND Variable < 90) OR (Variable > 100)** allows you to append nested AND/OR
- **Variable IN ('aaa', 'ccc')** allows you to match rows with certain strings in the data
- **Variable BETWEEN 80 AND 100** allows a selection of values between two numbers
- **Variable LIKE '%AN%'** uses wildcard matching % long strings and characters including spaces
- **Variable LIKE '_AN'** allows wildcard matching of a single character (_)
- **Variable1 / Variable2, Variable1 * Variable2, Variable1 + Variable2…** runs computations
- **(Y/100 + Z /10)/ 3 > X OR (Z - Y/100) > X** allows combinations of OR with computations
- **X < 4 UNION SELECT X FROM [first$] WHERE X > 10** allows union of multiple queries
- **ISNUMERIC(Variable)** allows the selection of numerical values only
- **1 = 2 UNION SELECT TOP 5 [first$].X,[first$].Y,[first$].Z FROM [first$]** allows choosing top few rows by incorporating the union and top functions
- **NOT X IN (SELECT TOP 5 [first$].[X] FROM [first$])** selects data not in the first few top rows
- **EXISTS (SELECT [first$].Z FROM [first$] WHERE Z>75)** checks to see if the query returns any values or if not, returns an empty set
- **Variable1 IN (SELECT [second$].[A] FROM [second$])** combines multiple data tables
- NOTES: **`Long Variable Names`** use back-tick to apply long variable names and regular ticks (apostrophe) for values (e.g., `Country of Origin` = 'United States')
- NOTES: **Union** will always sort the results of the first column in ascending order

Use Case 1: Selection of Rows by Value

Situation: In a large data set, we can use the conditional statement to select the rows with specific values (e.g., greater than a required threshold).

SQL Statement: Variable > Value

Example: Number > 100

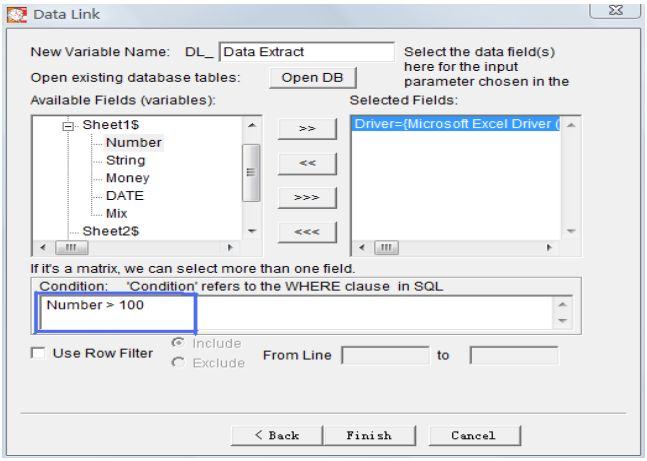
Example Profile and Model: Use Case 1-5.re and Use Case 1 model

Example Data File: Sample Data 1.xls

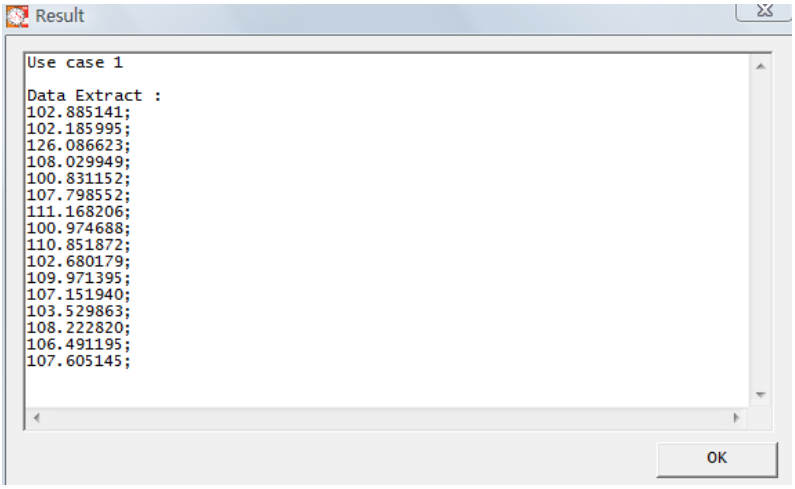
Note: You can use >=, <=, >, < inequalities

	A	B	C	D	E
1	Number	String	Money	DATE	Mix
2	102.885141	aaa	¥108.16	4/13/1900	102.88
3	84.705038	bbb	¥100.89	4/13/1900	aaa
4	92.160695	ccc	¥108.16	4/8/1900	122.26
5	102.185995	ddd	¥104.63	3/31/1900	108.46
6	91.309775	eee	¥111.91	4/28/1900	100.64
7	126.086623	fff	¥99.98	3/27/1900	99.74
8	108.029949	ggg	¥91.16	4/3/1900	110.64
9	88.869916	hhh	¥100.39	4/27/1900	83.52
10	95.844675	iii	¥108.17	4/4/1900	106.53
11	100.831152	jjj	¥115.29	4/2/1900	#$@#@#
12	107.798552	kkk	¥98.99	4/1/1900	107.64
13	111.168206	lll	¥78.03	4/19/1900	96.88
14	93.873964	mmm	¥109.57	4/7/1900	101.14
15	100.974688	nnn	¥104.37	4/16/1900	113.02

	A	B	C	D	E
16	110.851872	ooo	¥83.37	4/13/1900	103.05
17	102.680179	ppp	¥82.69	4/5/1900	102.87
18	69.971395	qqq	¥92.11	4/21/1900	*
19	93.250333	rrr	¥98.85	4/6/1900	89.48
20	96.570041	sss	¥104.34	3/26/1900	&&
21	97.653884	ttt	¥101.73	4/4/1900	108.71
22	75.886155	uuu	¥115.73	3/29/1900	108.09
23	107.151940	vvv	¥86.08	4/5/1900	95.85
24	103.529863	www	¥114.24	3/28/1900	95.75
25	108.222820	xxx	¥100.62	4/20/1900	()
26	106.491195	yyy	¥106.88	4/16/1900	108.62
27	80.822858	zzz	¥117.50	4/16/1900	100.64
28	91.103886	abc	¥103.53	4/8/1900	95.56
29	92.807726	def	¥85.72	4/14/1900	103.20
30	107.605145	ghi	¥84.81	4/4/1900	%%%%
31	94.677514	jkm	¥101.05	4/13/1900	FCCC



Result:



Use Case 2: Use of 'AND'

Situation: Use 'AND' to connect two or more conditions together, if all conditions are "TRUE", then the data is selected.

SQL Statement: condition AND condition AND condition AND…

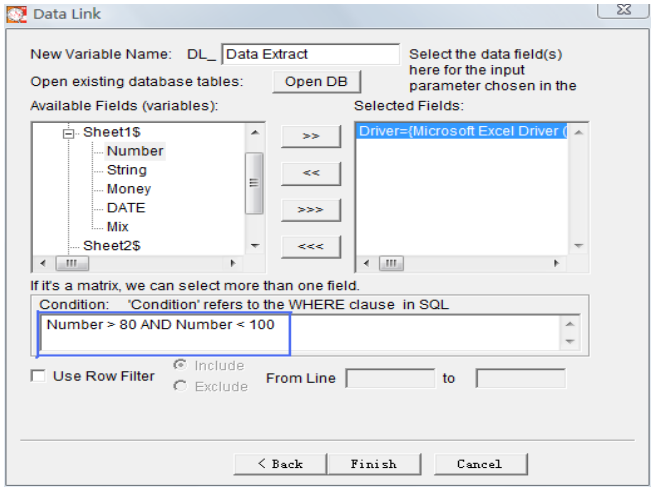
Example: Number > 80 AND Number < 100

Example Profile and Model: Use Case 1-5.re and Use Case 2 model

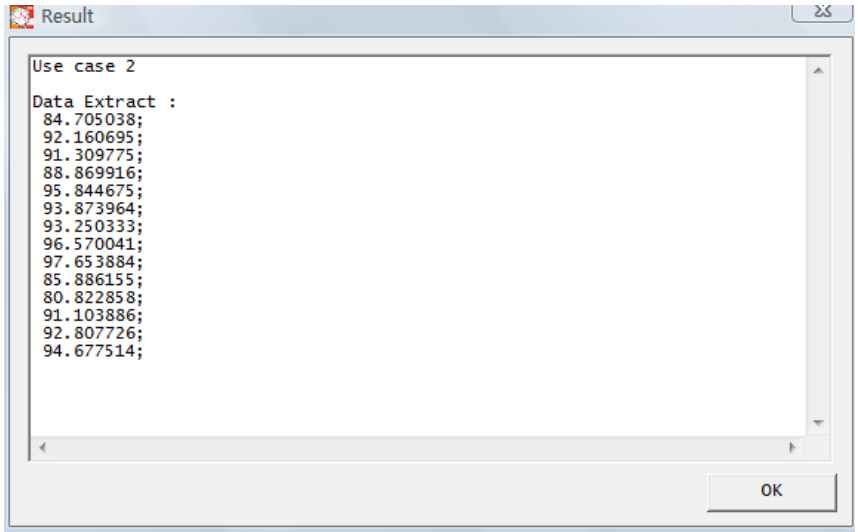
Example Data File: Sample Data 1.xls

	A	B	C	D	E
1	Number	String	Money	DATE	Mix
2	102.885141	aaa	¥108.16	4/13/1900	102.88
3	84.705038	bbb	¥100.89	4/13/1900	aaa
4	92.160695	ccc	¥108.16	4/8/1900	122.26
5	102.185995	ddd	¥104.63	3/31/1900	108.46
6	91.309775	eee	¥111.91	4/28/1900	100.64
7	126.086623	fff	¥99.98	3/27/1900	99.74
8	108.029949	ggg	¥91.16	4/3/1900	110.64
9	88.869916	hhh	¥100.39	4/27/1900	83.52
10	95.844675	iii	¥108.17	4/4/1900	106.53
11	100.831152	jjj	¥115.29	4/2/1900	#$@#@#
12	107.798552	kkk	¥98.99	4/1/1900	107.64
13	111.168206	lll	¥78.03	4/19/1900	96.88
14	93.873964	mmm	¥109.57	4/7/1900	101.14
15	100.974688	nnn	¥104.37	4/16/1900	113.02

	A	B	C	D	E
16	110.851872	ooo	¥83.37	4/13/1900	103.05
17	102.680179	ppp	¥82.69	4/5/1900	102.87
18	69.971395	qqq	¥92.11	4/21/1900	*
19	93.250333	rrr	¥98.85	4/6/1900	89.48
20	96.570041	sss	¥104.34	3/26/1900	&&
21	97.653884	ttt	¥101.73	4/4/1900	108.71
22	75.886155	uuu	¥115.73	3/29/1900	108.09
23	107.151940	vvv	¥86.08	4/5/1900	95.85
24	103.529863	www	¥114.24	3/28/1900	95.75
25	108.222820	xxx	¥100.62	4/20/1900	()
26	106.491195	yyy	¥106.88	4/16/1900	108.62
27	80.822858	zzz	¥117.50	4/16/1900	100.64
28	91.103886	abc	¥103.53	4/8/1900	95.56
29	92.807726	def	¥85.72	4/14/1900	103.20
30	107.605145	ghi	¥84.81	4/4/1900	%%%%
31	94.677514	jkm	¥101.05	4/13/1900	FCCC



Result:



Use Case 3: Use of 'OR'

Situation: Use 'OR' to connect two or more conditions together, once ant condition is "TRUE", the data is selected even when other conditions are "FALSE".

SQL Statement: condition OR condition OR condition OR…

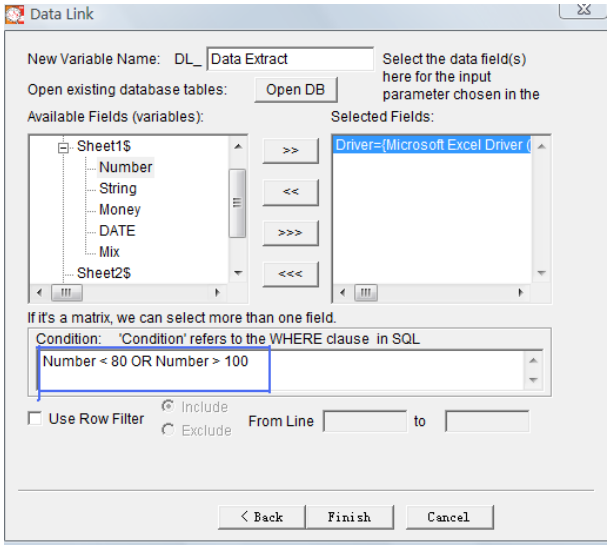
Example: Number < 80 OR Number > 100

Example Profile and Model: Use Case 1-5.re and Use Case 3 model

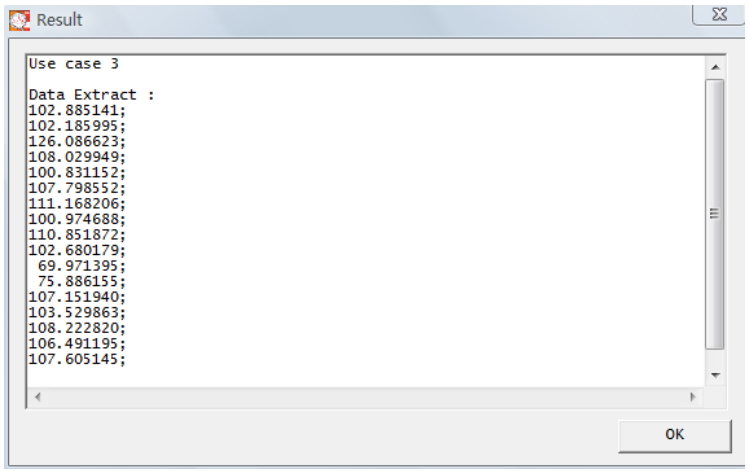
Example Data File: Sample Data 1.xls

	A	B	C	D	E
1	Number	String	Money	DATE	Mix
2	102.885141	aaa	¥108.16	4/13/1900	102.88
3	84.705038	bbb	¥100.89	4/13/1900	aaa
4	92.160695	ccc	¥108.16	4/8/1900	122.26
5	102.185995	ddd	¥104.63	3/31/1900	108.46
6	91.309775	eee	¥111.91	4/28/1900	100.64
7	126.086623	fff	¥99.98	3/27/1900	99.74
8	108.029949	ggg	¥91.16	4/3/1900	110.64
9	88.869916	hhh	¥100.39	4/27/1900	83.52
10	95.844675	iii	¥108.17	4/4/1900	106.53
11	100.831152	jjj	¥115.29	4/2/1900	#$@#@#
12	107.798552	kkk	¥98.99	4/1/1900	107.64
13	111.168206	lll	¥78.03	4/19/1900	96.88
14	93.873964	mmm	¥109.57	4/7/1900	101.14
15	100.974688	nnn	¥104.37	4/16/1900	113.02

	A	B	C	D	E
16	110.851872	ooo	¥83.37	4/13/1900	103.05
17	102.680179	ppp	¥82.69	4/5/1900	102.87
18	69.971395	qqq	¥92.11	4/21/1900	•
19	93.250333	rrr	¥98.85	4/6/1900	89.48
20	96.570041	sss	¥104.34	3/26/1900	&&
21	97.653884	ttt	¥101.73	4/4/1900	108.71
22	75.886155	uuu	¥115.73	3/29/1900	108.09
23	107.151940	vvv	¥86.08	4/5/1900	95.85
24	103.529863	www	¥114.24	3/28/1900	95.75
25	108.222820	xxx	¥100.62	4/20/1900	()
26	106.491195	yyy	¥106.88	4/16/1900	108.62
27	80.822858	zzz	¥117.50	4/16/1900	100.64
28	91.103886	abc	¥103.53	4/8/1900	95.56
29	92.807726	def	¥85.72	4/14/1900	103.20
30	107.605145	ghi	¥84.81	4/4/1900	%%%%
31	94.677514	jkm	¥101.05	4/13/1900	FCCC



Result:



Use Case 4: Use of 'AND' and 'OR' together

Situation: We can use 'AND' and 'OR' together to build a complex query command.

SQL Statement: **condition AND condition OR condition…**

Example: **(Number > 80 AND Number < 90) OR (Number > 100)**

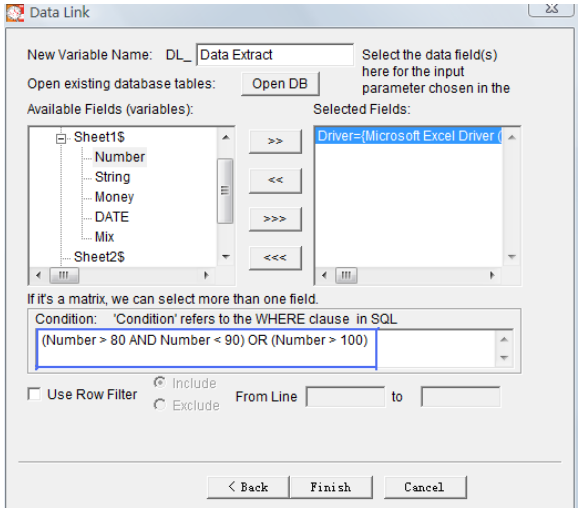
Example Profile and Model: Use Case 1-5.re and Use Case 4 model

Example Data File: Sample Data 1.xls

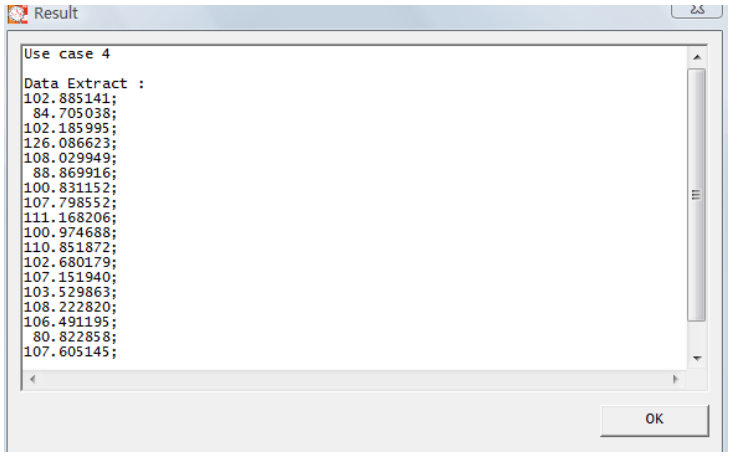
Notes: you can group commands using parenthesis ()

	A	B	C	D	E
1	Number	String	Money	DATE	Mix
2	102.885141	aaa	¥108.16	4/13/1900	102.88
3	84.705038	bbb	¥100.89	4/13/1900	aaa
4	92.160695	ccc	¥108.16	4/8/1900	122.26
5	102.185995	ddd	¥104.63	3/31/1900	108.46
6	91.309775	eee	¥111.91	4/28/1900	100.64
7	126.086623	fff	¥99.98	3/27/1900	99.74
8	108.029949	ggg	¥91.16	4/3/1900	110.64
9	88.869916	hhh	¥100.39	4/27/1900	83.52
10	95.844675	iii	¥108.17	4/4/1900	106.53
11	100.831152	jjj	¥115.29	4/2/1900	#$@#@#
12	107.798552	kkk	¥98.99	4/1/1900	107.64
13	111.168206	lll	¥78.03	4/19/1900	96.88
14	93.873964	mmm	¥109.57	4/7/1900	101.14
15	100.974688	nnn	¥104.37	4/16/1900	113.02

	A	B	C	D	E
16	110.851872	ooo	¥83.37	4/13/1900	103.05
17	102.680179	ppp	¥82.69	4/5/1900	102.87
18	69.971395	qqq	¥92.11	4/21/1900	*
19	93.250333	rrr	¥98.85	4/6/1900	89.48
20	96.570041	sss	¥104.34	3/26/1900	&&
21	97.653884	ttt	¥101.73	4/4/1900	108.71
22	75.886155	uuu	¥115.73	3/29/1900	108.09
23	107.151940	vvv	¥86.08	4/5/1900	95.85
24	103.529863	www	¥114.24	3/28/1900	95.75
25	108.222820	xxx	¥100.62	4/20/1900	()
26	106.491195	yyy	¥106.88	4/16/1900	108.62
27	80.822858	zzz	¥117.50	4/16/1900	100.64
28	91.103886	abc	¥103.53	4/8/1900	95.56
29	92.807726	def	¥85.72	4/14/1900	103.20
30	107.605145	ghi	¥84.81	4/4/1900	%%%%
31	94.677514	jkm	¥101.05	4/13/1900	FCCC



Result:



Use Case 5: Use of 'IN'

Situation: Use 'IN' command to specific a value (or multiple values) to match.

SQL Statement: Variable IN ('value1', 'value2'…)

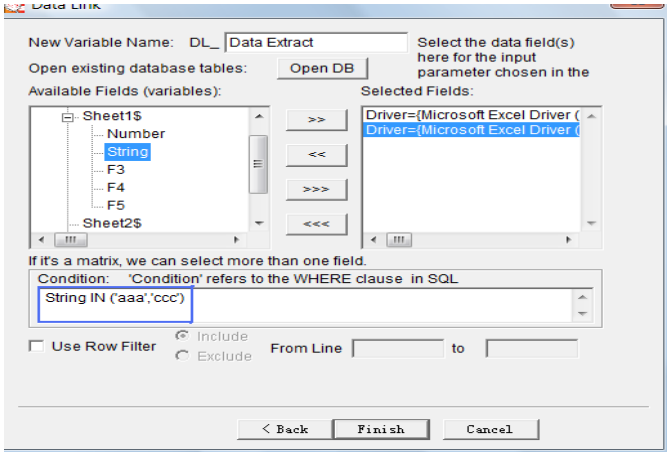
Example: String IN ('aaa', 'ccc')

Example Profile and Model: Use Case 1-5.re and Use Case 5 model

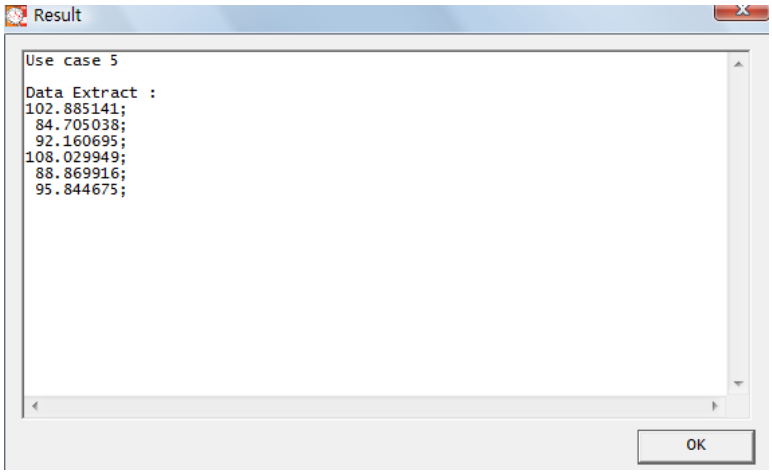
Example Data File: Sample Data 2.xls

Note: If the values filtered are strings, use 'quotes'

	A	B
1	**Number**	**String**
2	102.885141	aaa
3	84.705038	aaa
4	92.160695	aaa
5	102.185995	bbb
6	91.309775	bbb
7	126.086623	bbb
8	108.029949	ccc
9	88.869916	ccc
10	95.844675	ccc
11	100.831152	ddd
12	70.121340	ddd
13	65.121212	ddd



Result:



Use Case 6: Use of 'BETWEEN'

Situation: Using 'BETWEEN' selects data within a specific range.

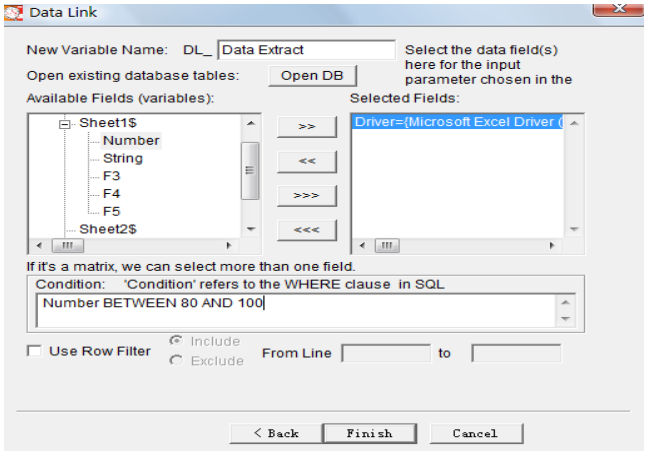
SQL Statement: Variable BETWEEN 'value1' AND 'value2'

Example: Number BETWEEN 80 AND 100

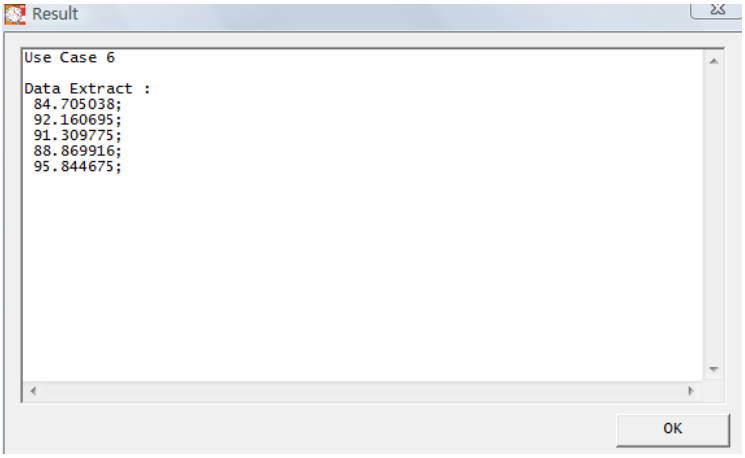
Example Profile and Model: Use Case 5-10.re and Use Case 6 model

Example Data File: Sample Data 2.xls

	A	B
1	**Number**	**String**
2	102.885141	aaa
3	84.705038	aaa
4	92.160695	aaa
5	102.185995	bbb
6	91.309775	bbb
7	126.086623	bbb
8	108.029949	ccc
9	88.869916	ccc
10	95.844675	ccc
11	100.831152	ddd
12	70.121340	ddd
13	65.121212	ddd



Result:



Use Case 7: Use of 'LIKE'

Situation: The 'LIKE' condition allows you to use wildcards in the Where clause, allowing you to perform pattern matching.

SQL Statement:

The patterns that you can choose from are:

% allows you to match any string of any length (including zero length)

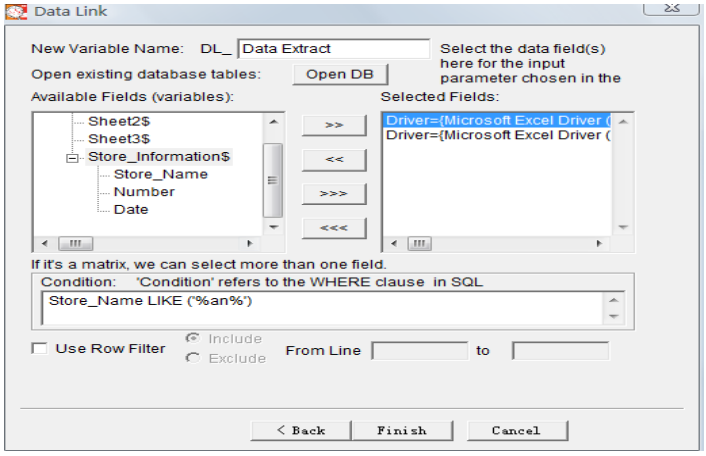
_ allows you to match on a single character

Example: store_name LIKE '%AN%'

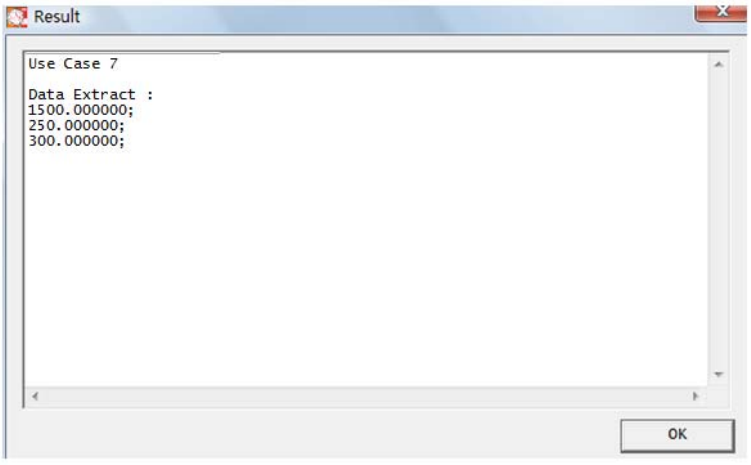
Example Profile and Model: Use Case 5-10.re and Use Case 7 model

Example Data File: Sample Data 3.xls

	A	B	C
1	**Store_Name**	**Number**	**Date**
2	Los Angeles	1500.00	8/1/2008
3	San Diego	250.00	5/1/2008
4	San Francisco	300.00	2008/6/31
5	Boston	700.00	4/23/2008



Result:



Use Case 8: Simple Math Functions

Situation: Basic mathematical functions can be applied on variables.

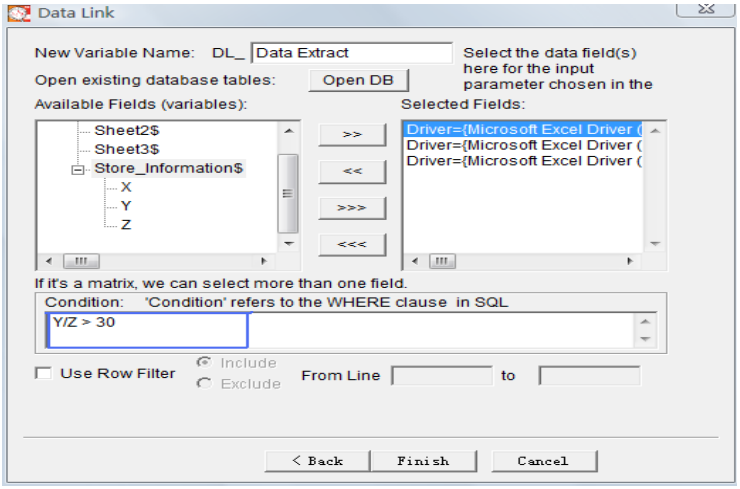
SQL Statement: **Variable1 / Variable2, Variable1 * Variable2, Variable1 + Variable2 …**

Example: **Y/Z > 30**

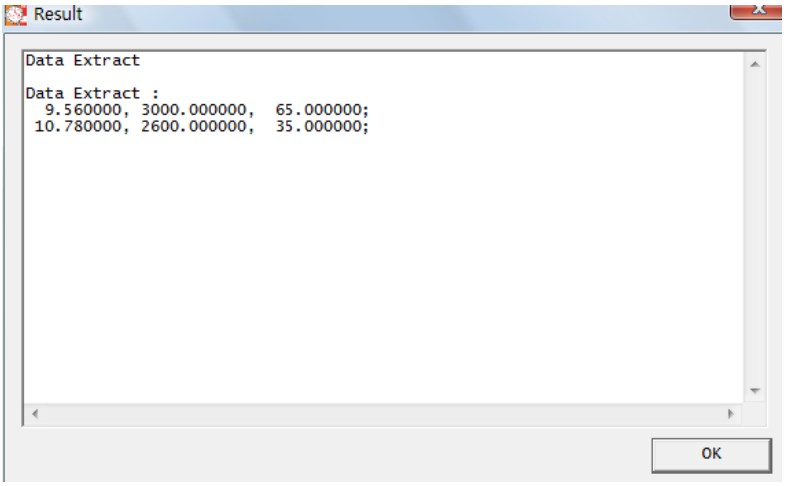
Example Profile and Model: Use Case 5-10.re and Use Case 8 model

Example Data File: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 9: Nested Math Functions

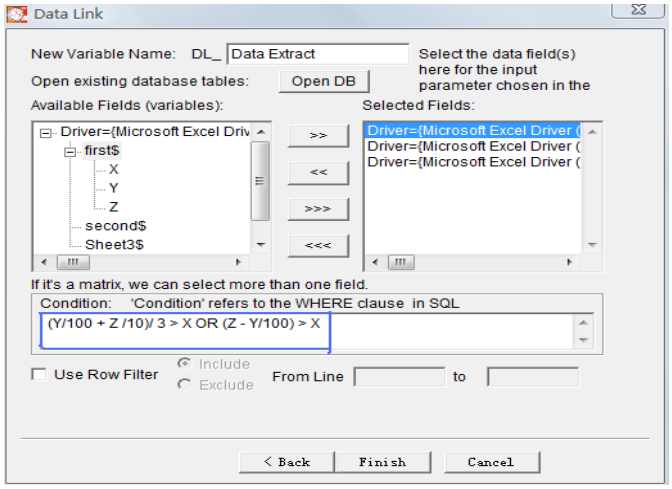
Situation: The math functions can be very complex (just like the mathematic functions).

Example: **(Y/100 + Z /10)/ 3 > X OR (Z - Y/100) > X**

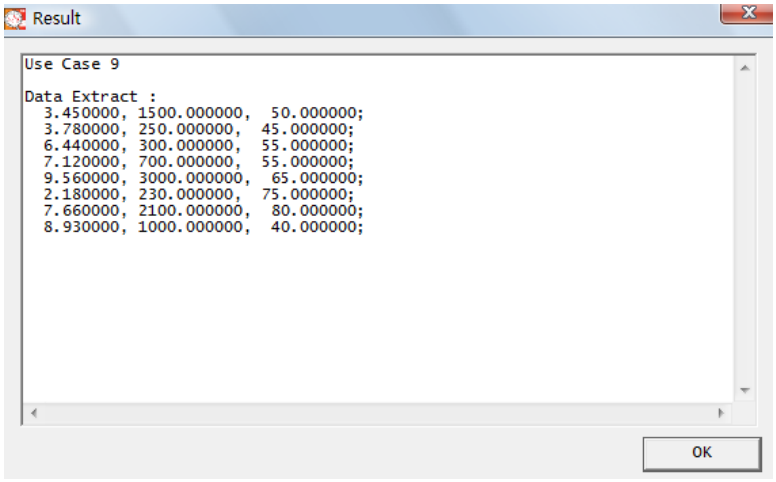
Example Profile and Model: Use Case 5-10.re and Use Case 9 model

Example Data File: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 10: Use of 'Union' to Connect Commands

Situation: Union is a very important command to connect two or more query results together. When creating complex commands, divide the entire command into small pieces and apply 'Union'.

SQL Statement: **CONDITION1 UNION SELECT COLUMN FROM TABLENAME WHERE CONDITION2**

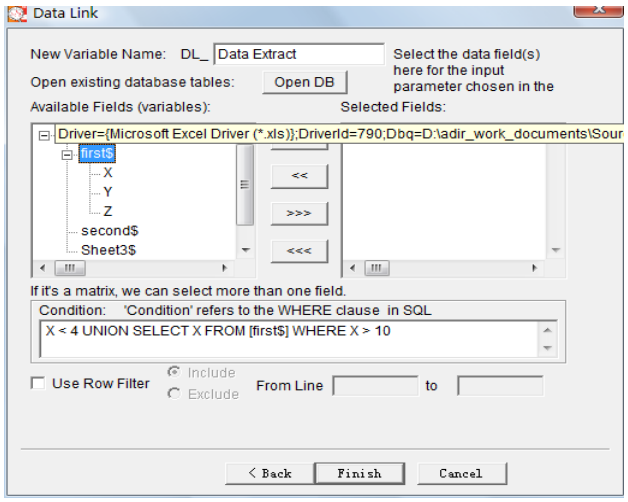
Example: **X < 4 UNION SELECT X FROM [first$] WHERE X > 10**

Example Profile and Model: Use Case 5-10.re and Use Case 10 model

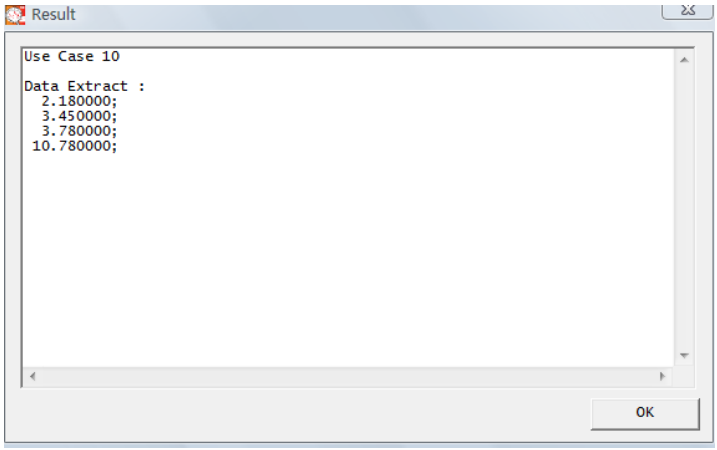
Example Data File: Sample Data 4.xls

Note: using Union can sometimes sort the resulting dataset

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 11: Filtering Different Value Types

Situation: If a column of data has mixed numbers and strings or other value types, we can filter in numerical data by applying the 'ISNUMERIC' command.

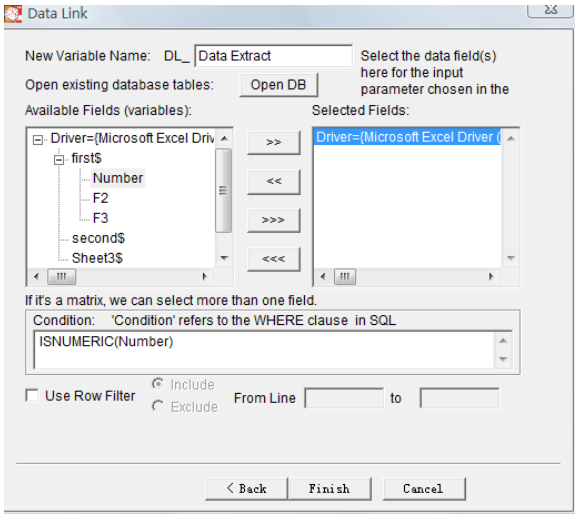
SQL Statement: ISNUMERIC(Variable)

Example: ISNUMERIC(Number)

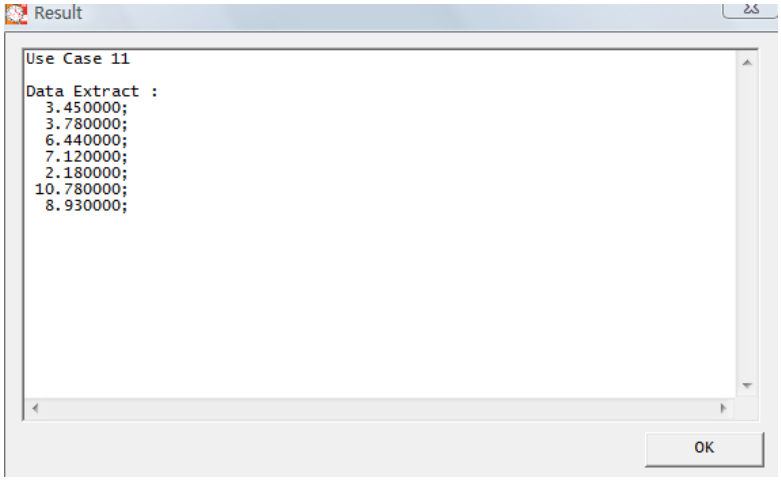
Example Profile and Model: Use Case 11-15.re and Use Case 11 model

Example Data File: Sample Data 5.xls

	A
1	**Number**
2	3.45
3	3.78
4	6.44
5	7.12
6	AaA
7	2.18
8	BBB
9	10.78
10	8.93



Result:



Use Case 12: Choosing the Top N Rows

Situation: To select the top N rows in a table, use 'UNION' and 'TOP' commands together.

SQL Statement: **TOP N * FROM TABLE_NAME**

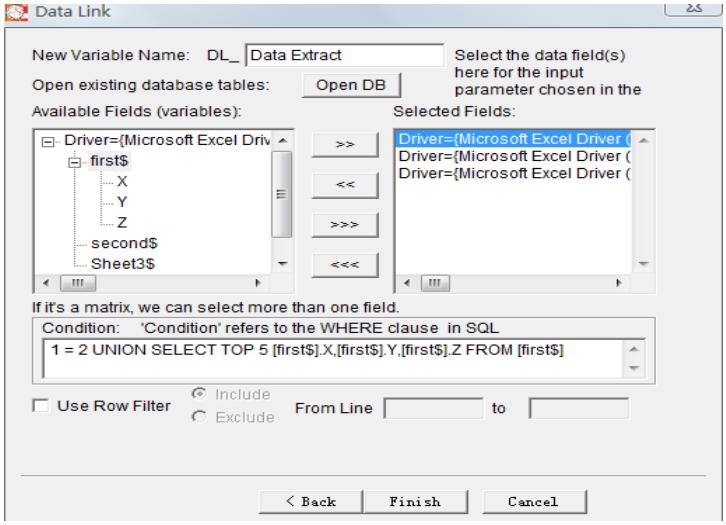
Example: **1 = 2 UNION SELECT TOP 5 [first$].X,[first$].Y,[first$].Z FROM [first$]**

Caution: The second SELECT command's selected rows must the same with the selected rows in list box. '1 = 2' means Forever FALSE so make the first select condition has no result.

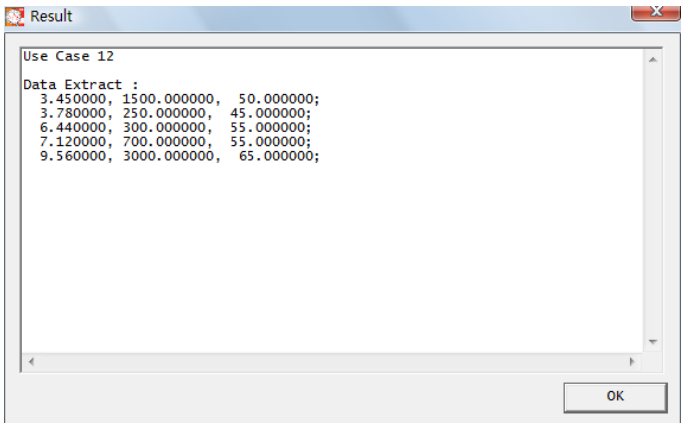
Example Profile and Model: Use Case 11-15.re and Use Case 12 model

Example Data File: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 13: Use of 'NOT IN'

Situation: 'NOT IN' is used to filter out values obtained from the next condition command. If the column's value is unique, it can be used to obtain values from a range of rows.

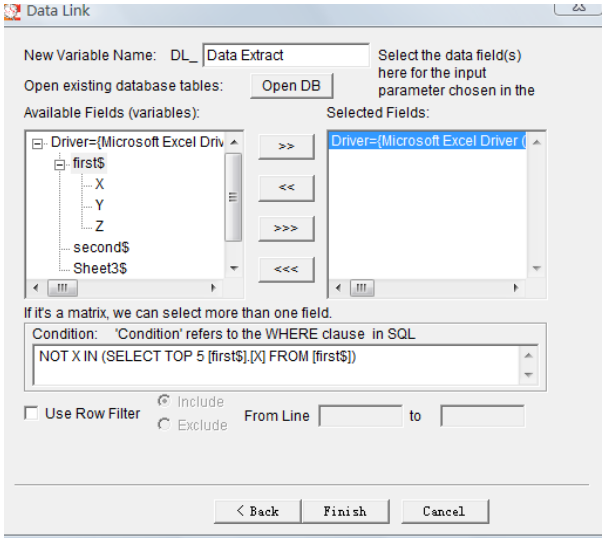
SQL Statement: NOT Variable IN (command)

Example: NOT X IN (SELECT TOP 5 [first$].[X] FROM [first$])

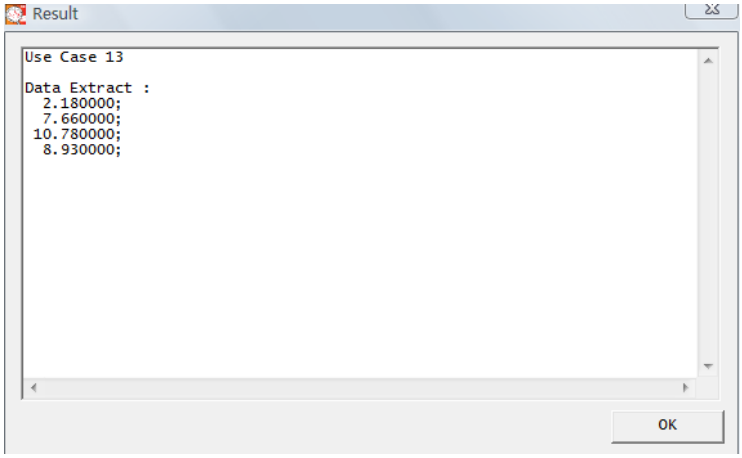
Example Profile and Model: Use Case 11-15.re and Use Case 13 model

Example Data File: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 14: Use of 'EXISTS'

Situation: 'EXISTS' simply tests whether the inner query returns any rows. If it does, then the outer query proceeds. If not, the outer query does not execute, and the entire SQL statement returns nothing.

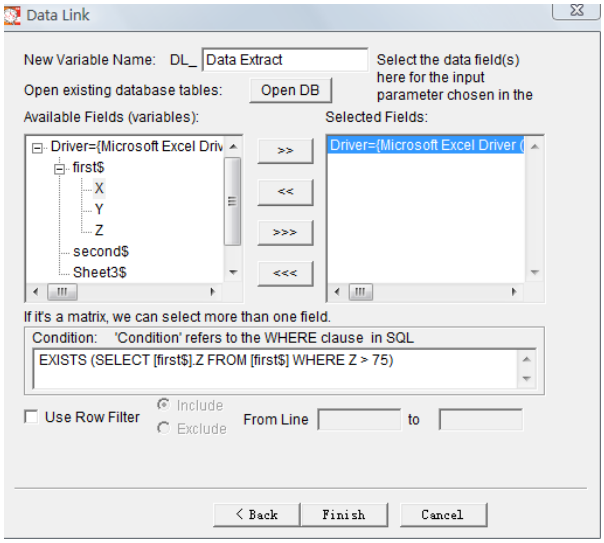
SQL Statement: **EXISTS (SELECT * FROM "table_name2" WHERE [Condition])**

Example: **EXISTS (SELECT [first$].Z FROM [first$] WHERE Z>75)**

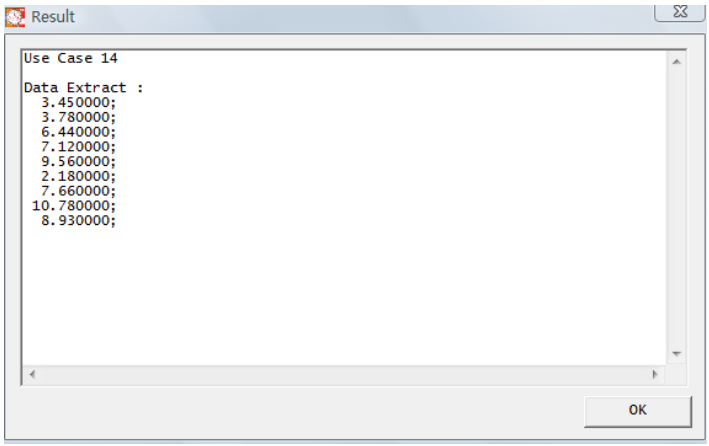
Example Profile and Model: Use Case 11-15.re and Use Case 14 model

Example Data File: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 15: Use of Multiple Table

Situation: Use the 'SELECT' command to connect multiple tables for matching elements.

SQL Statement: **Variable1 IN (SELECT Variable2 FROM Table_Name2 WHERE Condition2)**

Example: **X IN (SELECT [second$].[A] FROM [second$])**

Example Profile and Model: Use Case 11-15.re and Use Case 15 model

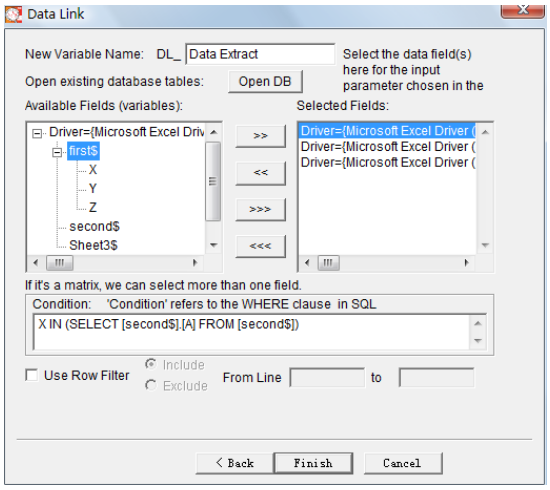
Example Data File: Sample Data 4.xls

	A	B	C
1	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00

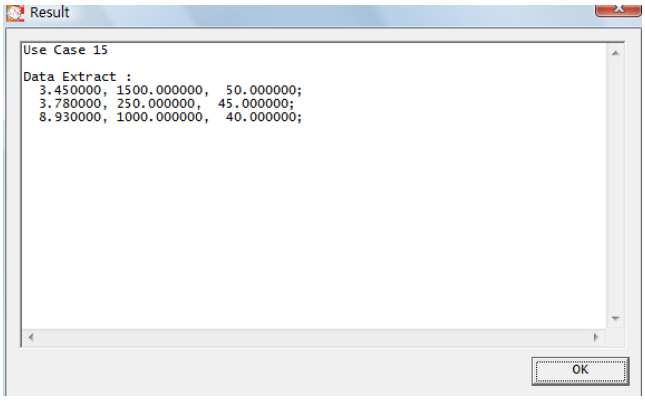
FIRST TABLE

	A
1	A
2	3.45
3	3.78
4	8.93
5	6.66

SECOND TABLE



Result:



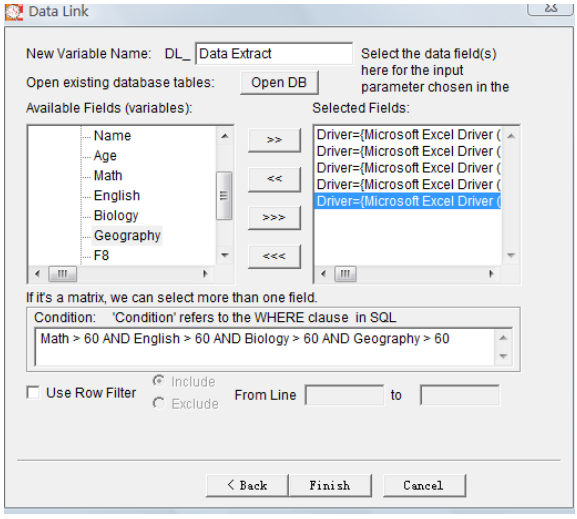
Use Case 16: Example using AND

Situation: Select the student's numbers for those who passed every test.

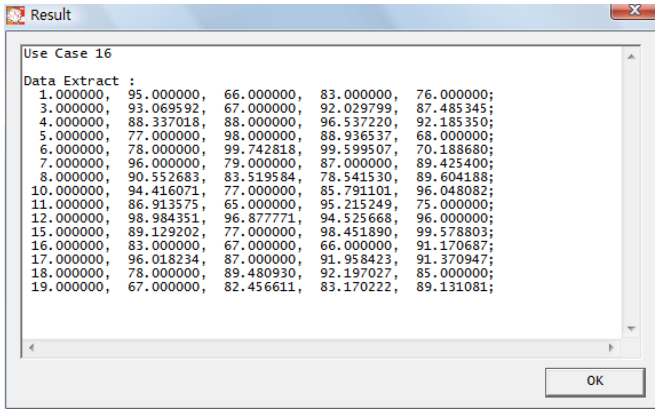
Example Profile and Model: Use Case 16-20.re and Use Case 16

Example Data File: Sample Data 6.xls

	A	B	C	D	E	F	G
1	Students No	Name	Age	Math	English	Biology	Geography
2	1	John	16	95	66	83	76
3	2	Tom	15	67	78	55	89
4	3	Jerry	16	93	67	92	87
5	4	Bob	17	88	88	97	92
6	5	Alexandra	16	77	98	89	68
7	6	William	18	78	100	100	70
8	7	Lily	15	96	79	87	89
9	8	Rose	16	91	84	79	90
10	9	Jack	14	99	57	92	93
11	10	Vivi	18	94	77	86	96
12	11	Vicky	15	87	65	95	75
13	12	Babala	15	99	97	95	96
14	13	Chris	17	76	57	87	98
15	14	Amanda	16	56	78	95	90
16	15	Alice	16	89	77	98	100
17	16	Amy	15	83	67	66	91
18	17	Annie	17	96	87	92	91
19	18	Cindy	16	78	89	92	85
20	19	Cora	17	67	82	83	89
21	20	Ella	18	67	65	86	56



Result:



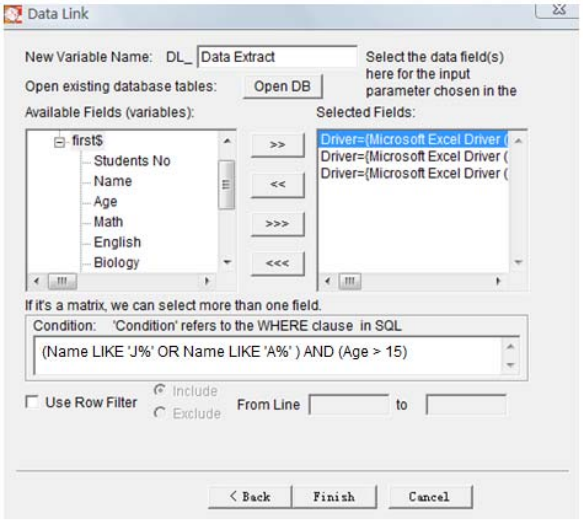
Use Case 17: Example using Wildcards with AND

Situation: Select the student's numbers whose Names begin with 'A' or 'J' and Age older than 16

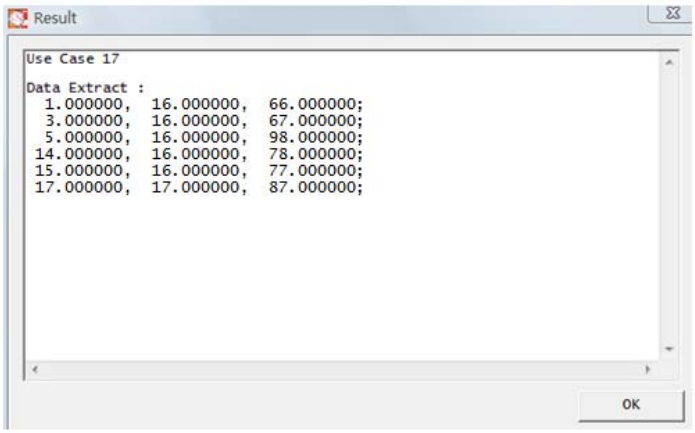
Example Profile and Model: Use Case 16-20.re and Use Case 17 model

Example Data File: Sample Data 6.xls

	A	B	C	D	E	F	G
1	Students No	Name	Age	Math	English	Biology	Geography
2	1	John	16	95	66	83	76
3	2	Tom	15	67	78	55	89
4	3	Jerry	16	93	67	92	87
5	4	Bob	17	88	88	97	92
6	5	Alexandra	16	77	98	89	68
7	6	William	18	78	100	100	70
8	7	Lily	15	96	79	87	89
9	8	Rose	16	91	84	79	90
10	9	Jack	14	99	57	92	93
11	10	Vivi	18	94	77	86	96
12	11	Vicky	15	87	65	95	75
13	12	Babala	15	99	97	95	96
14	13	Chris	17	76	57	87	98
15	14	Amanda	16	56	78	95	90
16	15	Alice	16	89	77	98	100
17	16	Amy	15	83	67	66	91
18	17	Annie	17	96	87	92	91
19	18	Cindy	16	78	89	92	85
20	19	Cora	17	67	82	83	89
21	20	Ella	18	67	65	86	56



Result:



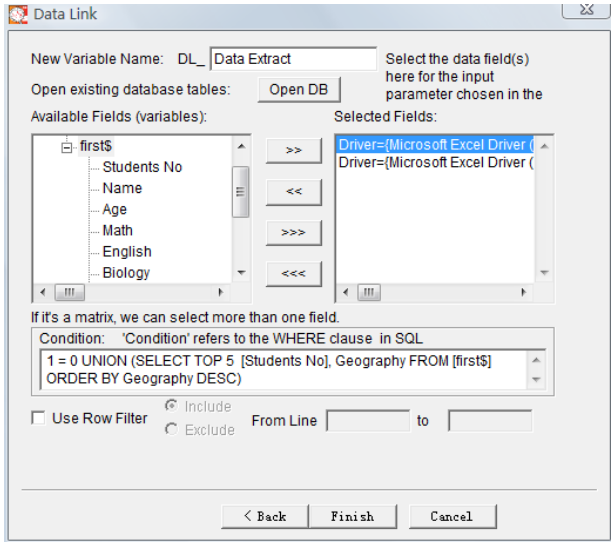
Use Case 18: Example using Union with Sorting

Situation: Select the top 5 highest scores in Geography.

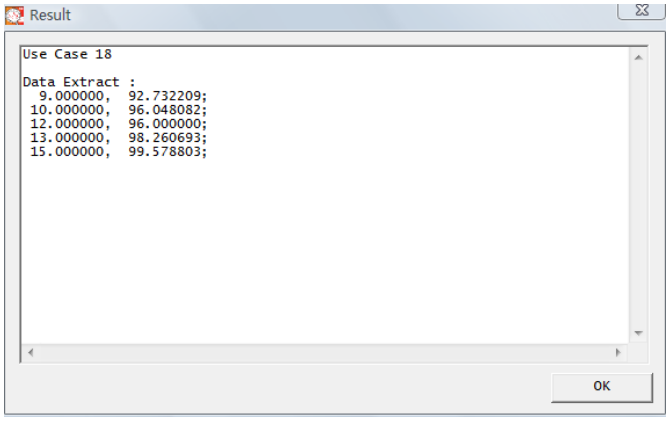
Example Profile and Model: Use Case 16-20.re and Use Case 18 model

Example Data File: Sample Data 6.xls

	A	B	C	D	E	F	G
1	Students No	Name	Age	Math	English	Biology	Geography
2	1	John	16	95	66	83	76.000000
3	2	Tom	15	67	78	55	88.601113
4	3	Jerry	16	93	67	92	87.485345
5	4	Bob	17	88	88	97	92.185350
6	5	Alexandra	16	77	98	89	68.000000
7	6	William	18	78	100	100	70.188680
8	7	Lily	15	96	79	87	89.425400
9	8	Rose	16	91	84	79	89.604188
10	9	Jack	14	99	57	92	92.732209
11	10	Vivi	18	94	77	86	96.048082
12	11	Vicky	15	87	65	95	75.000000
13	12	Babala	15	99	97	95	96.000000
14	13	Chris	17	76	57	87	98.260693
15	14	Amanda	16	56	78	95	89.711824
16	15	Alice	16	89	77	98	99.578803
17	16	Amy	15	83	67	66	91.170687
18	17	Annie	17	96	87	92	91.370947
19	18	Cindy	16	78	89	92	85.000000
20	19	Cora	17	67	82	83	89.131081
21	20	Ella	18	67	65	86	56.000000



Result:



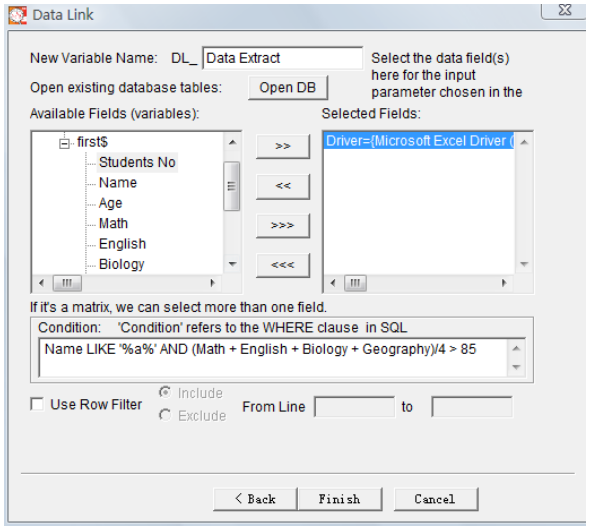
Use Case 19: Example using Wildcards and Math

Situation: Select the students whose names contain the character 'A' and the average score is greater than 85.

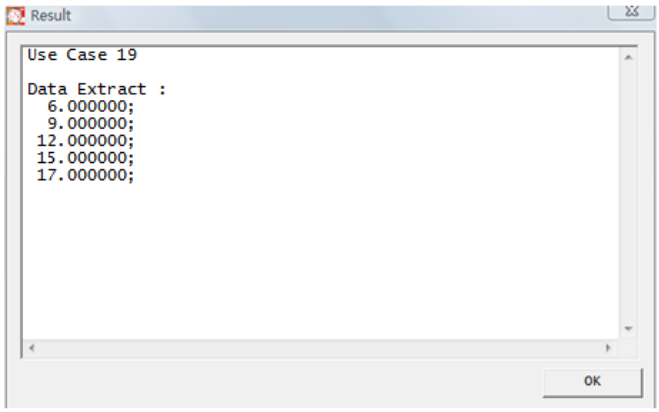
Example Profile and Model: Use Case 16-20.re and Use Case 19 model

Example Data File: Sample Data 6.xls

	A	B	C	D	E	F	G	H
1	Students No	Name	Age	Math	English	Biology	Geography	Average
2	1	John	16	95	66	83	76	*80.00*
3	2	Tom	15	67	78	55	89	*72.15*
4	3	Jerry	16	93	67	92	87	*84.90*
5	4	Bob	17	88	88	97	92	*91.26*
6	5	Alexandra	16	77	98	89	68	*82.98*
7	6	William	18	78	100	100	70	*86.88*
8	7	Lily	15	96	79	87	89	*87.86*
9	8	Rose	16	91	84	79	90	*85.55*
10	9	Jack	14	99	57	92	93	*85.13*
11	10	Vivi	18	94	77	86	96	*88.31*
12	11	Vicky	15	87	65	95	75	*80.53*
13	12	Babala	15	99	97	95	96	*96.60*
14	13	Chris	17	76	57	87	98	*79.57*
15	14	Amanda	16	56	78	95	90	*79.79*
16	15	Alice	16	89	77	98	100	*91.04*
17	16	Amy	15	83	67	66	91	*76.79*
18	17	Annie	17	96	87	92	91	*91.59*
19	18	Cindy	16	78	89	92	85	*86.17*
20	19	Cora	17	67	82	83	89	*80.44*
21	20	Ella	18	67	65	86	56	*68.50*



Result:



Use Case 20: Example using Nested AND/OR with Math

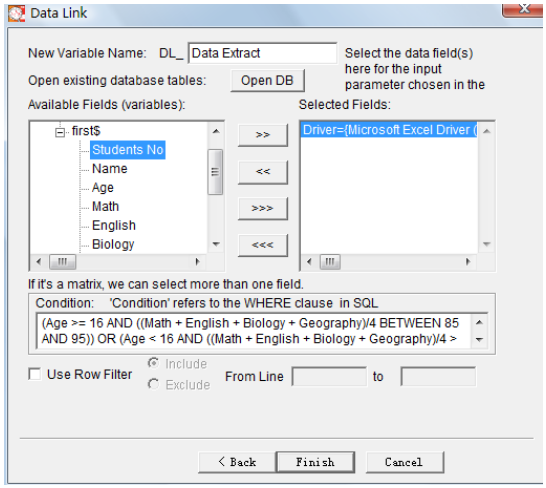
Question: Select the students who have an average score between 85 and 95 when the student's age is >= 16 or has an average score higher than 80 when the student's age is < 16.

Example Profile and Model: Use Case 16-20.re and Use Case 20 model

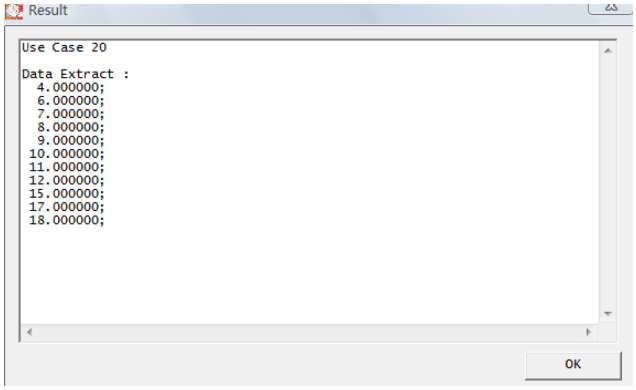
Example Data File: Sample Data 6.xls

	A	B	C	D	E	F	G	H
1	Students No	Name	Age	Math	English	Biology	Geography	Average
2	1	John	16	95	66	83	76	80.00
3	2	Tom	15	67	78	55	89	72.15
4	3	Jerry	16	93	67	92	87	84.90
5	4	Bob	17	88	88	97	92	91.26
6	5	Alexandra	16	77	98	89	68	82.98
7	6	William	18	78	100	100	70	86.88
8	7	Lily	15	96	79	87	89	87.86
9	8	Rose	16	91	84	79	90	85.55
10	9	Jack	14	99	57	92	93	85.13
11	10	Vivi	18	94	77	86	96	88.31
12	11	Vicky	15	87	65	95	75	80.53
13	12	Babala	15	99	97	95	96	96.60
14	13	Chris	17	76	57	87	98	79.57
15	14	Amanda	16	56	78	95	90	79.79
16	15	Alice	16	89	77	98	100	91.04
17	16	Amy	15	83	67	66	91	76.79
18	17	Annie	17	96	87	92	91	91.59
19	18	Cindy	16	78	89	92	85	86.17
20	19	Cora	17	67	82	83	89	80.44
21	20	Ella	18	67	65	86	56	68.50

(Age >= 16 AND ((Math + English + Biology + Geography)/4 BETWEEN 85 AND 95)) OR (Age < 16 AND ((Math + English + Biology + Geography)/4 > 80))



Result:



Use Case 21: Use of 'UNION ALL'

Situation: The purpose of the 'UNION ALL' command is to combine the results of two queries. The difference between 'UNION ALL' and 'UNION' is that, while 'UNION' only selects distinct values, 'UNION ALL' selects all values.

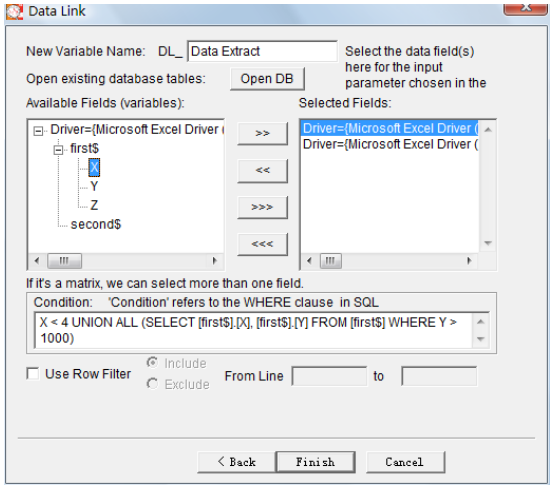
SQL Statement: **[SQL Statement 1] UNION ALL [SQL Statement 2]**

Example: **X < 4 UNION ALL (SELECT [first$].[X], [first$].[Y] FROM [first$] WHERE Y > 1000)**

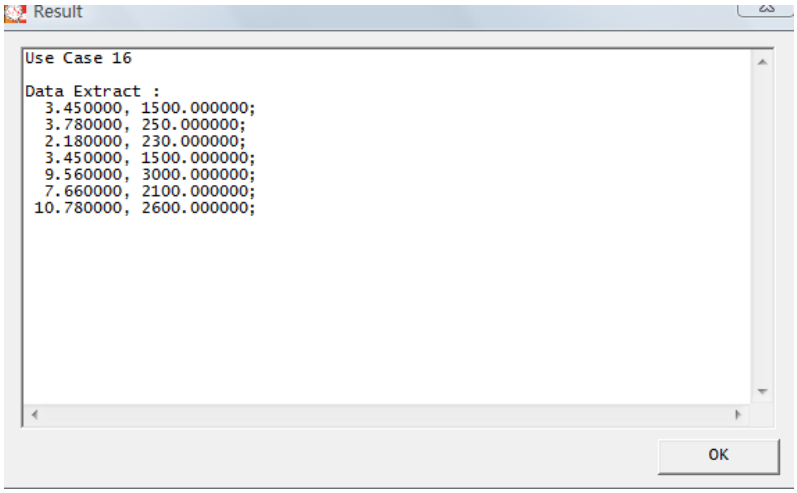
Example Profile and Model: Use Case 21-25.re and Use Case 21 model

Example Data File: Sample Data 4.xls

	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 22: Use of SQL Functions

Situation: SQL has several arithmetic functions, they are 'AVG', 'COUNT', 'MAX', 'MIN', 'SUM'. They are useful when we have to do some function with the result.

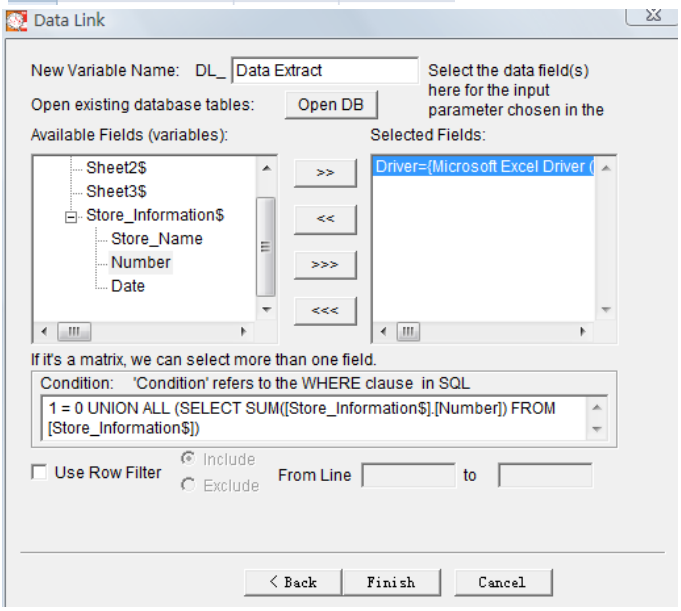
SQL Statement: **SELECT "function type"("column_name") FROM "table_name"**

Example: **1 = 0 UNION ALL (SELECT SUM([Store_Information$].[Number]) FROM [Store_Information$])**

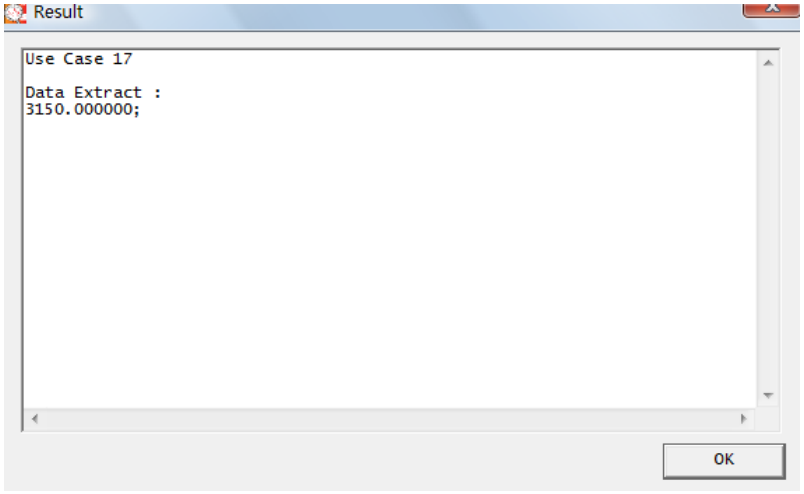
Example Profile and Model: Use Case 20-25.re and Use Case 22 model

Example Data File: Sample Data 7.xls

	Store_Name	Number	Date
2	Los Angeles	1500.00	2008/8/1
3	San Diego	250.00	2008/5/1
4	San Francisco	300.00	2008/6/31
5	Boston	700.00	2008/4/23
6	Los Angeles	400.00	2008/6/1



Result:



Use Case 23: Use of 'GROUP BY'

Situation: In Use Case 22 we use 'sum' to compute the total number of all store, what do we do if wanted to compute each store's number? We can accomplish this by using 'GROUP BY'.

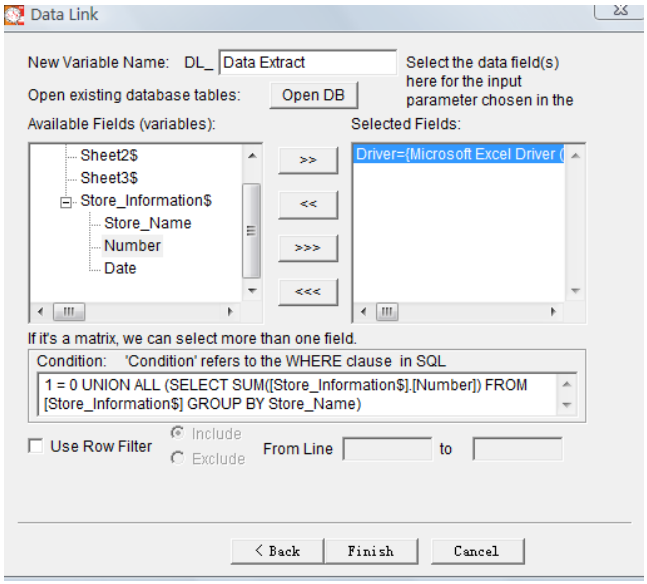
SQL Statement: **SELECT "column_name1", SUM("column_name2") FROM "table_name" GROUP BY "column_name1"**

Example: **1 = 0 UNION ALL (SELECT SUM([Store_Information$].[Number]) FROM [Store_Information$] GROUP BY Store_Name)**

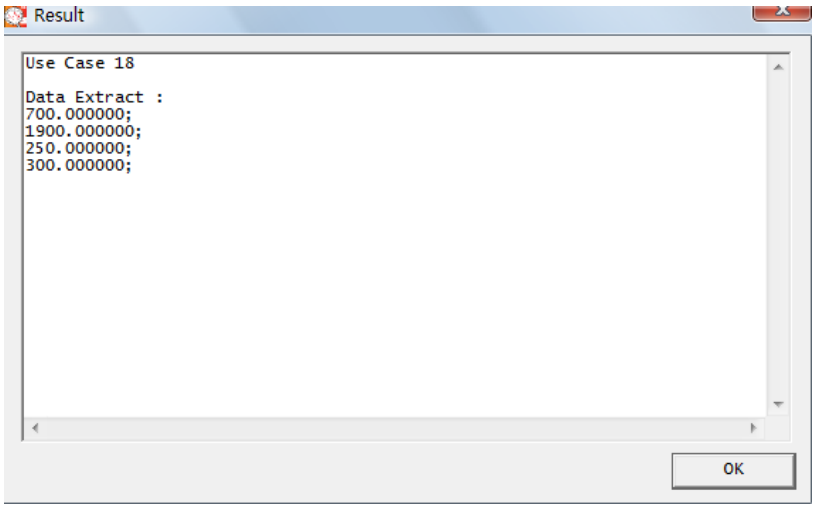
Example Profile and Model: Use Case 20-25.re and Use Case 23 model

Example Data File: Sample Data 7.xls

	Store_Name	Number	Date
2	Los Angeles	1500.00	2008/8/1
3	San Diego	250.00	2008/5/1
4	San Francisco	300.00	2008/6/31
5	Boston	700.00	2008/4/23
6	Los Angeles	400.00	2008/6/1



Result:



Use Case 24: Use of 'DISTINCT'

Situation: When in a column where some values are similar and you don't want to show them, use the 'DISTINCT' command for showing unique values.

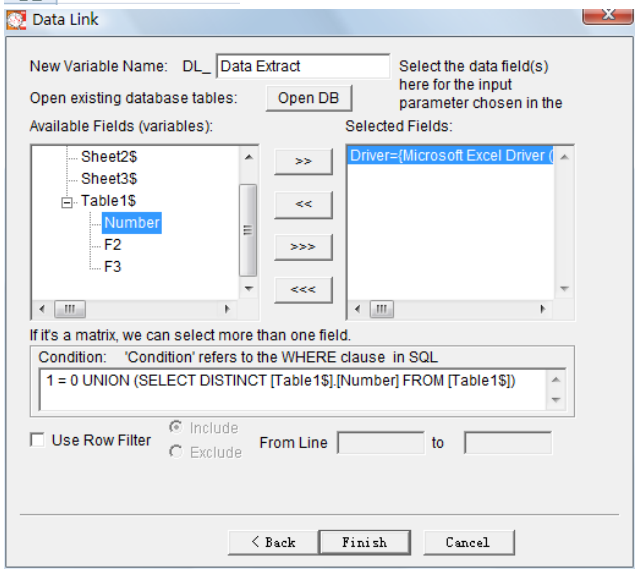
SQL Statement: **SELECT DISTINCT Variable FROM Table_name**

Example: **1 = 0 UNION (SELECT DISTINCT [Table1$].[Number] FROM [Table1$])**

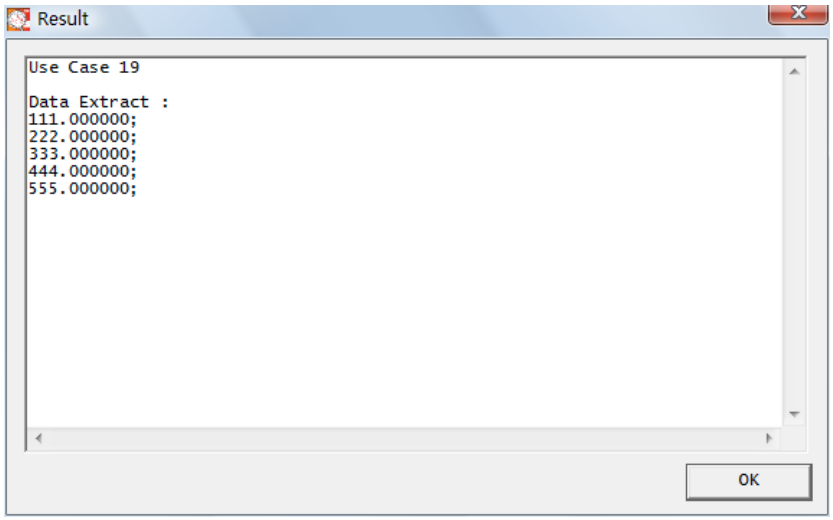
Example Profile and Model: Use Case 20-25.re and Use Case 24 model

Example Data File: Sample Data 8.xls

	Number
1	
2	111
3	111
4	111
5	222
6	222
7	222
8	333
9	444
10	555
11	444
12	222



Result:



Use Case 25: Use of 'ORDER BY'

Situation: When you need to list the data in a particular order, use the 'ORDER BY' command.

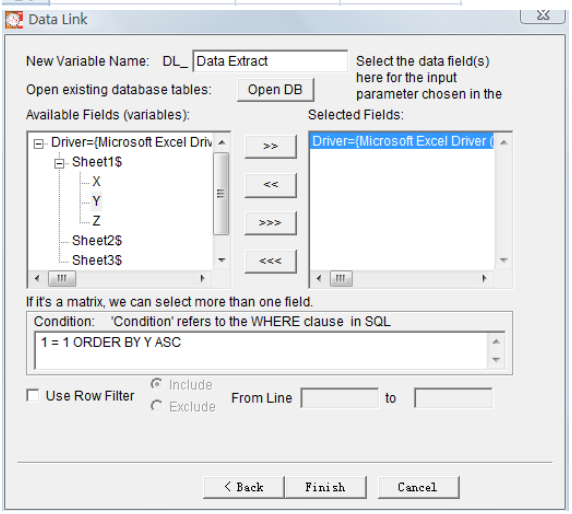
SQL Statement: **SELECT "column_name" FROM "table_name" [WHERE "condition"]**
ORDER BY "column_name" [ASC, DESC]

Example: **Number > 80 AND Number < 100**

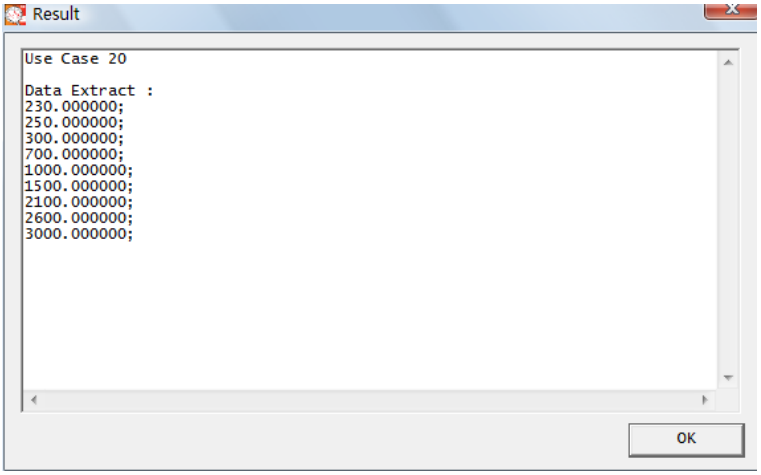
Example Profile and Model: Use Case 20-25.re and Use Case 25 model

Example Data File: Sample Data 4.xls

	X	Y	Z
2	3.45	1500.00	50.00
3	3.78	250.00	45.00
4	6.44	300.00	55.00
5	7.12	700.00	55.00
6	9.56	3000.00	65.00
7	2.18	230.00	75.00
8	7.66	2100.00	80.00
9	10.78	2600.00	35.00
10	8.93	1000.00	40.00



Result:



Use Case 26: Selection by Dates with 'BETWEEN'

Situation: 'Between' can be used in a Date variable but requires a special format to use.

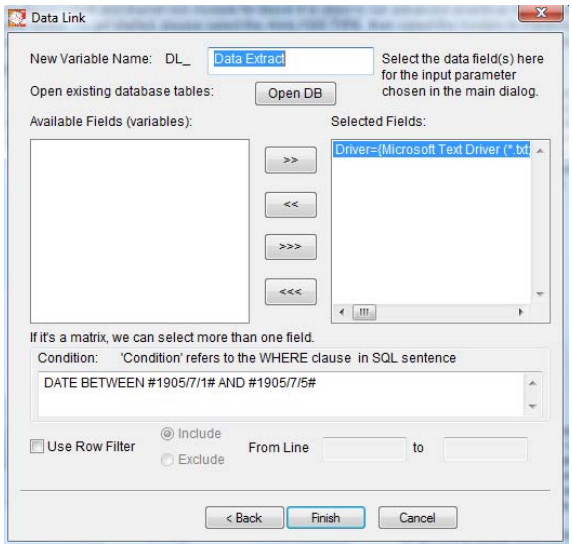
SQL Statement: **BETWEEN #date1# AND #date2#**

Example: **DATE BETWEEN #1905/7/1# AND #1905/7/5#**

Example Profile and Model: Use Case Dates.re and the two model in this profile

Example Data File: Sample Data 9.xls and Sample Data 10.csv

	A	B	C	D
1	Normal (Multi)	Uniform	Binomial	DATE
2	87.53	45.29	6	7/1/1905
3	abc	45.29	6	7/2/1905
4	99.66	46.94	6	7/3/1905
5	108.75	45.96	6	7/4/1905
6	108.75	#$45.96	6	7/5/1905



Result: