TIMi

CREATIVITY THROUGH EFFICIENCY

Anatella

analytical ETL

Anatella Quick User's guide v3.05

Company headquarters:

Address: CheMin des 2 Villers, 11 - 7812 Ath (V.N.D.) - Belgium

Phone (global): +32 479 99 27 68

Phone (Americas): + 57 300 675 13 69

Creation date: March 2010

Last Edited: April 2023

Last Edited by Frank Vanden Berghen

1. Introduction

Welcome to the Quick user's guide to Anatella. Anatella is part of the "TIMi Suite[tm]" that is an integrated set of tools designed to solve Advanced Analytics, Big Data and Business Intelligence problems. The acronym "TIMi" stands for "The Intelligent Mining Machine (TIMi)".

Anatella is a data manipulation tool, also known as "ETL tool" ("ETL" is the acronym of "Extract, Transform & Load") that is designed for analytical and "predictive dataMining" tasks. The name "Anatella" is the contraction of "Analytical ETL": indeed Anatella is the first ETL tool of its kind: it's the only ETL tool designed specifically for Analytical Tasks on large data volume. Anatella offers some features that are unique and extremely valuable in this field (e.g. meta-data-free data transformations, ability to handle tables with more than 50000 columns, etc.).

Before any data Mining activity, before building any predictive model, the first task to accomplish by the data Miner is to place all the available data into the proper format (when using TIMi or Stardust, it usually means obtaining one single very large dataset). Once you have a dataset, you can analyze it with TIMi and/or StarDust.

1.1. General Data Manipulation Features

Anatella offers you the following capabilities:

- Anatella is **100% Unicode compliant** and will accept any character set (Chinese, Cyrillic, Japanese, etc.) without losing any information.

- Classical ETL features :
  - Join tables
  - Columns & Rows filtering out of tables,
  - Sorting,
  - Format conversions (CSV, SQL ,..),
  - Automatic Scoring (using the TIMi predictive models)
  - Automatic segmentation (using the StarDust segmentation models)
  - Derivation of new columns for predictive modeling:
    - Automatic generation of hundreds of thousands of new, "derivate" columns ;
    - a full scripting language based on JavaScript (standard ECMA-262) that allows you to express the most complex transformations, validations, aggregations, derivations. The small Anatella-specific extensions to the "standard" JavaScript language are easy to use. Furthermore, thanks to these extensions, the JavaScript code becomes siMilar (but more versatile) to a "SAS datastep", so that you can even leverage your SAS skills.
  - Complete Meta-Data extraction & management.

- Inside Anatella, most of the data transformation operators are "**meta-data free**". It means that it is not necessary to define "metas-datas" to use 99% of the various transformations available in Anatella.

©2010 TIMi SPRL. E-mail: sales@tiMi.eu - Visit us at www.tiMi.eu

In this regard, Anatella is very much like MS-Excel: Inside Excel, you don't need to specify "by hand" the data-types of each of your columns or cells: Excel automatically find the right data-types, so that the equations inside your sheets are working without you spending time to define any meta-datas. The same principle applies with Anatella.

The "**meta-data free**" functionality of Anatella makes it completely different than all the other ETL's currently on the market. It also greatly simplifies the usage of Anatella: The difficulty in using Anatella is comparable to the difficulty of using Excel (it's only slightly more complex). This means that business-users without technical training are usually able to use Anatella without too much headaches.

The "**meta-data free**" functionality is also important because, in predictive dataMining, it is very common to manipulate tables of tens of thousands of columns and it is impossible to specify "by hand" the "meta-datas" of all these columns (as required by nearly all other ETL software).

- Anatella features **two highly-optiMized data file formats** that have the extension ".gel_Anatella" (row-based file format) and ".**c**gel_anatella" (column-based file format). These two file formats are primarily optiMized for speed.

  One ".gel_anatella" file (or one ".cgel_anatella" file) contains one data table (and all its meta-data).

  Usually you can read ".gel_anatella" file at a throughput of 70 MB/sec (on common hardware). The data inside a simple ".gel_anatella" file is compressed by a factor around 4. This means that reading a table out of a ".gel_Anatella" file at a speed of 70 MB/sec (compressed) is actually equivalent to reading the same table at a speed of **280 MB/sec** (uncompressed).

  A very common situation in business-intelligence is to compute aggregates based on a subset of the columns available inside the data file. Aggregates are typically computed using only about 4 columns out of the many columns stored inside the data file. The columnar file-format ".**c**gel_anatella" allows you to just extract, out of the hard-drive, the few bytes that are composing the 4 columns required to compute the aggregation (and avoid reading&decompressing the bytes that belong to all the other columns). In practice, this means that, for most business-intelligence tasks, the "**c**gel_anatella" files are usually from five to hundred times faster than the simpler ".gel_anatella" files (at the cost of a slightly higher RAM memory consumption)(simply because you avoid reading the data/columns that you don't need to read).

  These performances place Anatella ahead of most competitive offerings (as evidenced by its score on the TPC-H benchmark).

- Anatella can read in "native mode" compressed dataset files in text format (CSV). The supported **compression formats** are RAR (unique!), ZIP, GZ, Z, LZO.

  This functionality allows to reduce the need for hard-drive space on your server.

  Let's give an example: the open source "census-income" database stored in a .RAR compressed text file "weights" 4.04 MB. The same database in an un-compressed text file "weights" 96MB (let's say around 100MB). The same database in a classical SAS .sas7bdat dataset file "weights" around 250MB. The numbers given on this example are quite common and perfectly illustrates

©2010 TIMi SPRL. E-mail: sales@tiMi.eu - Visit us at www.tiMi.eu

why an ETL should be able to process compressed data streams (even more so when dealing with "*Big Data*").

- Anatella reads natively **SAS datasets** file (.sas7bdat file), SPSS datasets files (.sav and .por files) and STATA (.dta files) datasets files.

- Anatella is heavily **multi-threaded**. This means that one data transformation running in Anatella can exploit all the CPU's inside your server to decrease the computation time. Classical ETL's are only able to run different data transformations on different CPU's but not one particular data transformation on many CPU's. The multi-threading capabilities of Anatella usually allow dividing the computing time of a data transformation graph by a factor between 4 to 10.

- Anatella offers you a direct access to all the "classical" **relational databases** via ODBC & OLEDB connectors (Oracle, SQLServer, MySQL, TerraData, ...)

- Anatella provides some crude **OLAP reporting functionalities** through the use of a "Microsoft Office Data Injection operator": This operator allows you to automatically inject "in batch" some data extracted from the Anatella-Graph into any chart or graphics contained in any Microsoft Office document. For example, you can obtain, in a few mouse clicks, each day, an automatic update of all the charts of your preferred PowerPoint presentation. Anatella can generate all types of MSOffice graphs: pie chart, 3D surface chart, bar chart, doughnut chart, bubble chart, etc...

1.2. Development functionalities

- The Anatella **integrated-development-environment** (IDE) that is used to create the new data-manipulation-scripts is extremely simple, intuitive & versatile. This environment is based on a unique hybrid technology:
    - The simple transformations are described using "little boxes" (that is the most intuitive way to represent a data transformation and is a « *de facto* » standard for all the modern ETL tools).
    - Complex transformations are programmed using a scripting language based on JavaScript (standard ECMA-262) which is simple, complete and extremely versatile. JavaScript is one of the most widely used programMing language currently used in the industry (see appendix C, E and F). You can leverage your already-existing-JavaScript-skills to become an ETL expert instantaneously!

- Anatella is the only ETL tool available on the market to offer you a direct access to a complete & powerful "debugger" with an interface siMilar to the famous *MS Visual Studio debugger* (to "debug" the scripts written in JavaScript/ECMA-262): you can add "break points" to your code, add some "watch" on variables, see the "stack",... This feature adds a lot of flexibility and control to the ETL process.

©2010 TIMi SPRL. E-mail: sales@tiMi.eu - Visit us at www.tiMi.eu

One major advantage of Anatella over any other ETL is that Anatella is easily extensible: you can easily add new, custoMized data-transformation-operators. These new operators can be developed in:

- JavaScript, R or Python.

  Anatella contains an automatic versioning tool that allows you to manage the different "JavaScript/R/python codes" that you have developed/downloaded. Anatella also offers you a direct access to a Javascript "debugger" to easily debug all your transformations.

- C/C++, for the most extreme speed & performances.

  A Software Developement Kit (SDK) to create new Actions coded in C++ inside Anatella is available for TIMi partners and clients. The SDK also allows you to extend the Anatella Javascript engine by adding new Javascript functions.

  This SDK was used to create all the actions inside Anatella: it offers the ultimate flexibility and speed.

  Deploying your new C++ Actions is easy: You only need to copy your "AnatellaPlugin*.dll" file next to the "anatella.exe" file and Anatella will automatically load your extensions at each startup.

1.3. Predictive Modeling Functionalities

- Anatella provides advanced **text-Mining** capabilities. Using the Anatella text-Mining operators, you can:
  - Automatically correct spelling Mistakes (in your "address" fields, for example…)
  - Translate text from one language to another
  - Do "fuzzy matching": For example, to join 2 tables (based on multilingual sound encoding)
  - Classify texts (in combination with TIM). These operators apply the classical "bag-of-word" technique to produce, starting from raw, unstructured text-data, many new columns and new variables directly exploitable inside TIM or Stardust, for predictive analytics. You can easily enrich your datasets with unstructured data to obtain the highest predictive modeling accuracy. The Anatella predictive-text-Mining functionalities are unique.

- **Graph Mining or Social Network Analysis** (SNA): This set of operators is mainly useful for telecommunication companies and banks, to create churn predictive models, cross-selling predictive models, up-selling predictive models, to estimate the share-of-wallet, etc. The objective of these operators is to extract out of the "phone- communication-network" valuable social-metrics. Typically, the "phone-communication-network" is defined in this way:
  - each individual is a node.
  - an "arc" of the network between the two individuals A and B represents the relation "A called B".

  The social-metrics that could be extracted from the network are: the best connected individual, the individual who plays the most important role in any group, the groups of friends, the proxiMity to a churner, the number of churners in the "neighborhood" of an individual. Those metrics can improve the accuracy of your predictive models.

©2010 TIMi SPRL. E-mail: sales@tiMi.eu - Visit us at www.tiMi.eu

- **Operational Research (OR) optiMization toolbox**: In particular, Anatella integrates:
  - An efficient multi-threaded LP/IP solver that allows you to solve large scale optiMization problem. The LP solver handles Millions of constraints and several thousands of variables.
  - An efficient solver for the GAP (General Assignment Problem). Typical GAP problems includes: Which product do I have to offer to which customer when I have the following business-constraints: The stock of each product is liMited, Each selected customer receives a folder with N offers (no less and no more than N offers), The margin on each product is different,... The GAP solver included inside Anatella handles campaigns with several Millions of customers and thousands of products.

  The Anatella OptiMization plugin is typically used for operation research (OR), sales & profit optiMization, stock optiMization, etc.
- **Modeling Factory** for large scale Predictive DataMining projects. Some of our customers need to re-build from scratch several thousands of predictive models every day or week. This can easy be accomplished using TIMi (as the analytical engine) and using Anatella to supervise & manage, in a 100% automated way, the whole procedure. The multithreading capabilities of Anatella allow you to exploit all the CPU's in your server to deliver a very high computing power.

2. Anatella Installation

Simply run the provided "AnatellaSetup.exe" wizard-based installation software.

3. How to run Anatella?

To run Anatella, you either double-click on the Anatella icon on your desktop:

... or you can double-click, inside a "MSWindow-File-Explorer-Window", on any Anatella-Data-Transformation-Graph file (i.e. any file with the extension ".anatella"). This will run Anatella and open the corresponding file.

To open an Anatella-data-transformation-graph file (i.e. an ".anatella" file), you can:

- Double-click an ".anatella" file inside your "Ms-Windows File Explorer" (this first option is my favourite).
- Use the drop-down menu FILE ⇨ OPEN inside Anatella
- Drag&drop your file from the "Windows File Explorer" into the Anatella window.

TIP: When you double-click an ".anatella" file inside the "MS-Windows File explorer", Anatella searches for all the opened Windows to check if the ".anatella" file is not already opened somewhere. If your ".anatella" graph file is indeed already opened, the focus changes to the proper Window, allowing you to instantaneously continue editing your ".anatella" file. In this way, there is no danger of editing the same ".anatella" file using several different Windows (because Anatella always prevents that from happening).

©2010 TIMi SPRL. E-mail: sales@tiMi.eu - Visit us at www.tiMi.eu

This is very convenient. Personnally, I always keep visible a “MS-Windows File explorer” Window showing me all the “.anatella” files that I am currently working on: I use this “File explorer” Window to quickly switch from editing one “.anatella” graph to another graph: I simply double-click on the file I want to edit to rapidly switch from one “.anatella” file to another.

©2010 TIMi SPRL. E-mail: sales@tiMi.eu - Visit us at www.tiMi.eu

4. How to use Anatella?

A good starting point to learn how to use Anatella is the tutorial available here:
http://timi.eu/support/anatella/
This tutorial is composed of 8 videos and covers everything that you need to know when starting to use Anatella.

The screen capture below illustrates the main window of Anatella in which all the development occurs.

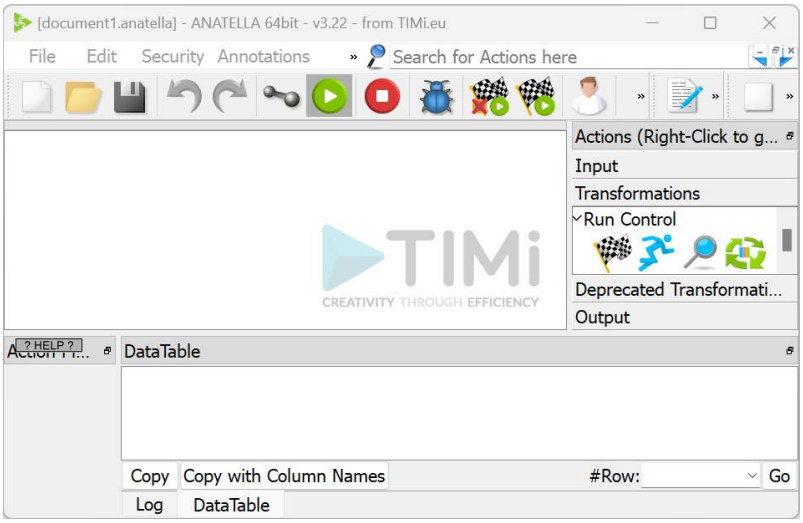


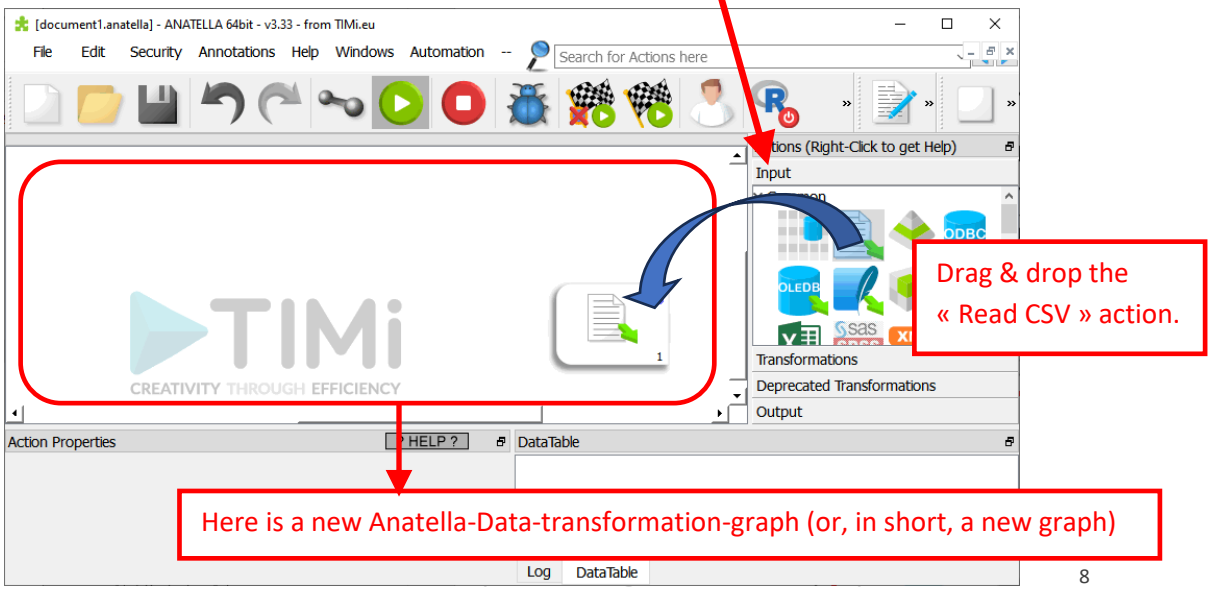
Figure 1: Main Window

4.1. A first example.

Let's start with a really simple example. We will create an Anatella-Data-Transformation-Graph (or, in short, a "graph") that reads a CSV/Text file.

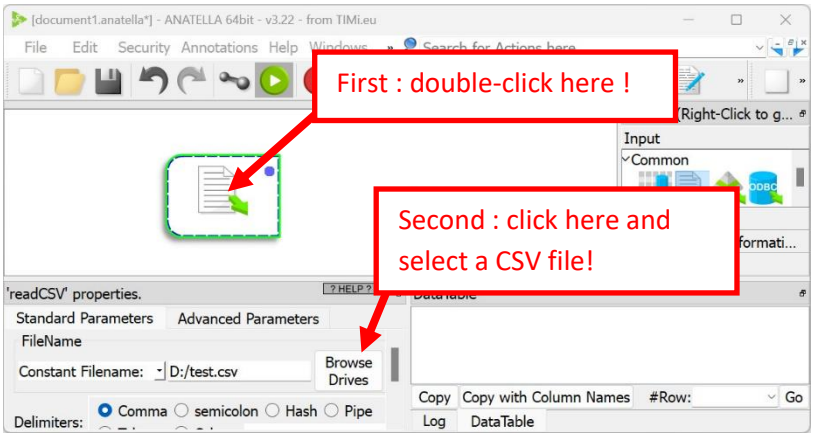
When you run Anatella (e.g. when you double-click on the Anatella icon on your desktop to open it), Anatella automatically creates a "blank" (empty) graph that is directly ready for you to use.

Click on the "Input" panel on the right-side of the window, here. From the "Input" panel, use your mouse to drag&drop into the central window the icon (taken from the right-side-panel). This icon represents a "*CSV/Text file reader*": it allows you to read the content of a CSV/Text file. Here is an illustration of the procedure:

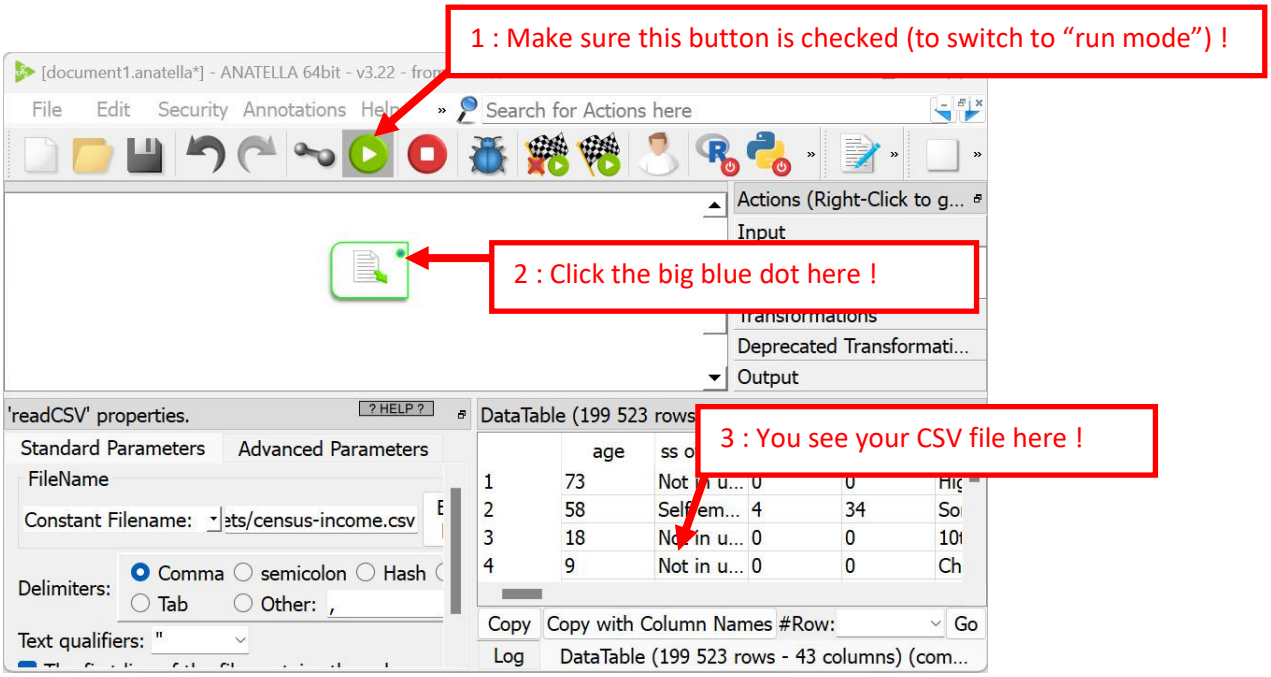


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Thereafter, double-click the icon inside the Anatella Graph: This will open the configuration screen of the "CSV file reader". Click on the Browse Drives button and select a test CSV file. You should obtain something like this:



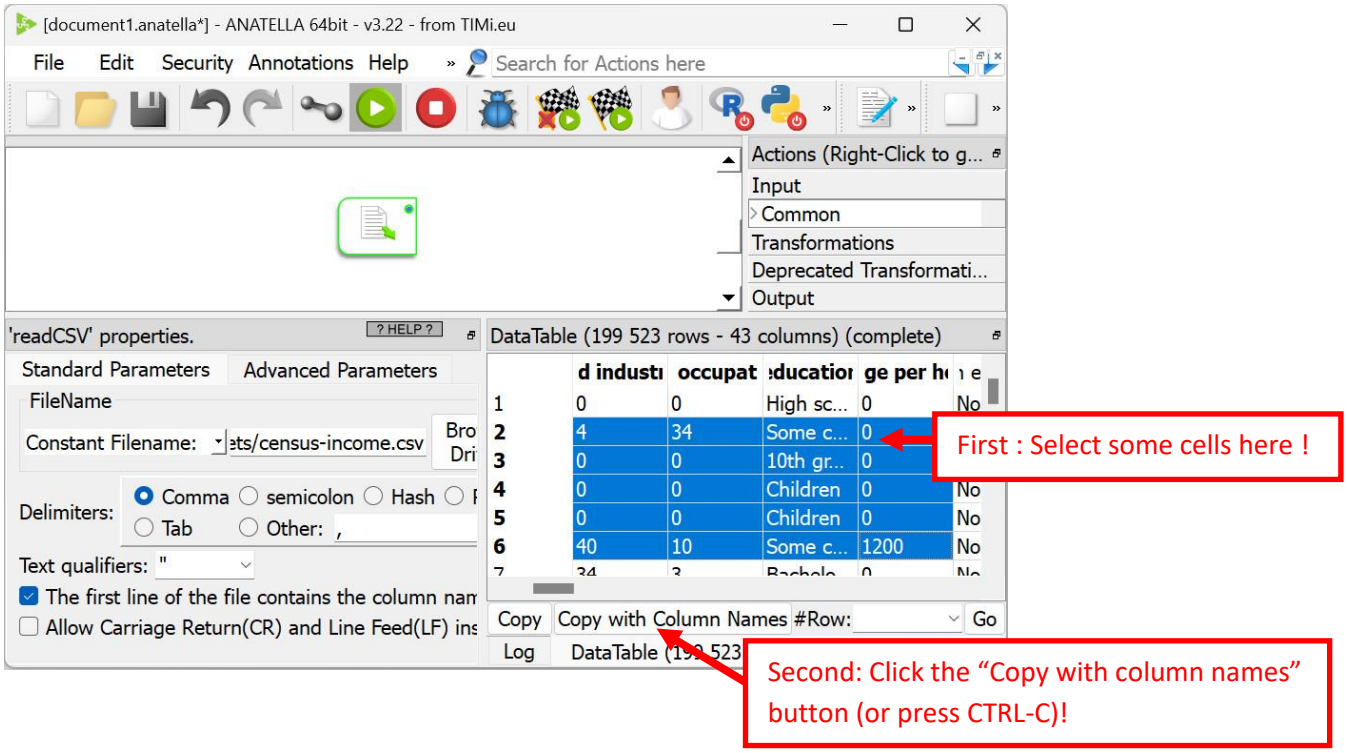
We just created a very small Anatella-Graph that reads a CSV/Text file named "d:\test.csv". Now, we would like to know if the settings of the "CSV File reader" are correct (Is it the right file? Is it the right separator?). To check if the settings are correct, we would like to "see" the CSV file. Make sure that the button in the main toolbar is checked (normally, it already is) and thereafter click the "big blue dot" inside the "CSV file reader" box. This "big blue dot" is named an "ouput pin" and it represents the "output table" of the Action. Here is an illustration of the procedure:



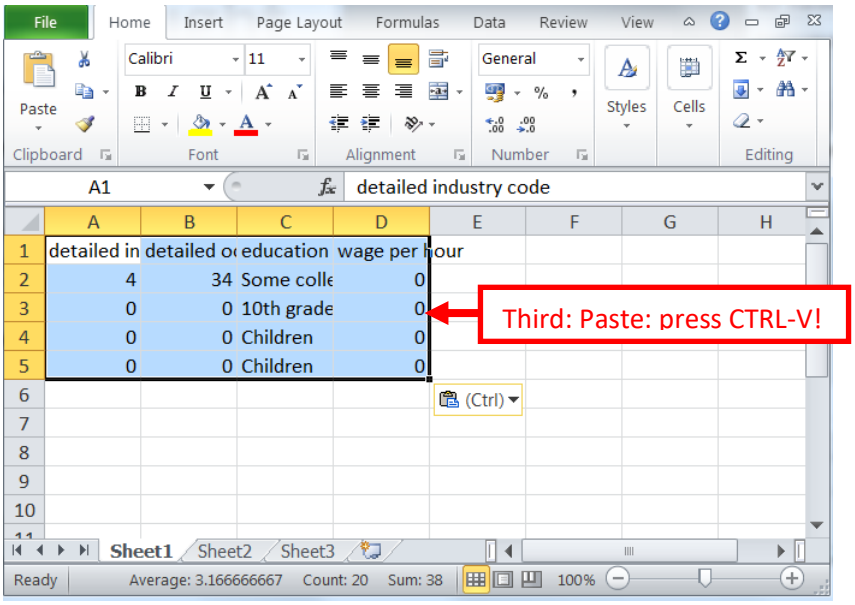
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.1.1. About the "Data-Preview" window

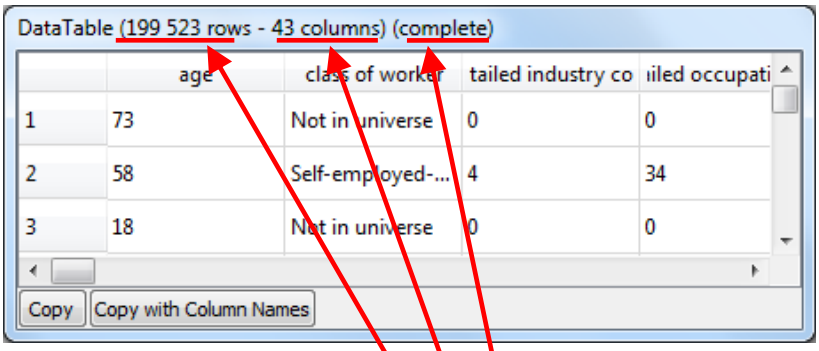
You can copy-paste the results that are inside the "Data-Preview-Table" into Excel. For example:



Open Excel and paste the results: (Press CTRL-V): You obtain:



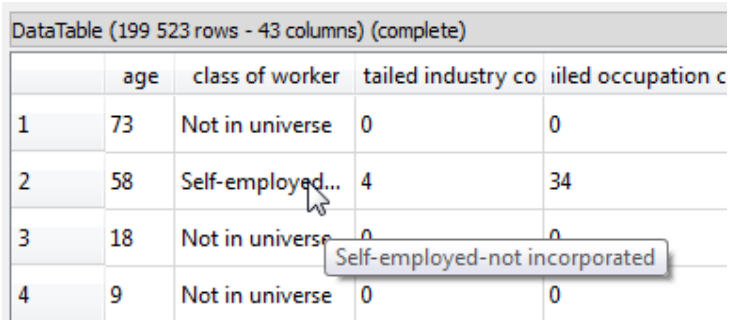
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

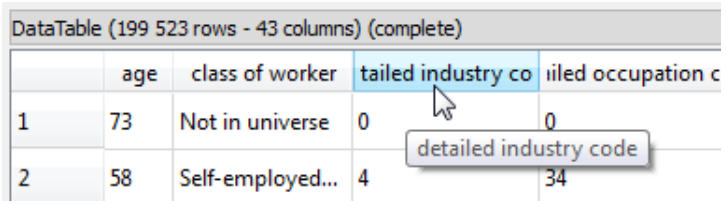


The Header of the "Data-Preview" table contains:

- The number of rows inside the table at this output pin.
- The number of columns inside the table at this output pin.
- The status of the output table and the status of the view.
  This status can be:
  - "**Complete":** The table at this output pin has been completely computed (it will thus been used to accelerate further computations along this data flow: See section 4.2.6. for more information about this subject) and the current "Data-Preview" window is able to directly show you the complete result table.
  - **"Complete (partial view):** The table at this output pin has been completely computed but only a partical view of this table is visible because the table is too big to show in its entirety.
  - **"Fragment":** Only a fragment (i.e. the first few rows) of the table at this output pin has been computed (typically because the user clicked the STOP button on the toolbar to abort the computations before completion). This fragment is now visible inside the "Data-Preview".
    Note: An incomplete table cannot be used to accelerate the computations later-on.

You can "hover" with your mouse above a cell (or above a column-title) to see the complete cell (or to see the complete title). For example:





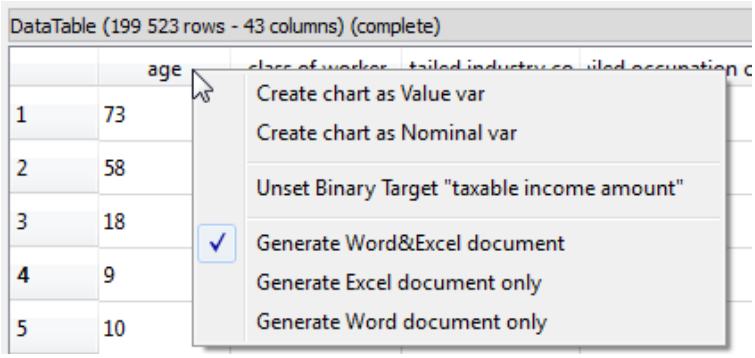
You can quickly obtain a "Data-Preview" of the data at any point inside your Anatella-Data-Transformation-Graph (there is one exception: see section 5.26.2.). To know how to create and use "Data-Previews", see the section 4.2.6. The "Data-Preview" window is thus a great tool to:

- check for data quality issues.
- check if you correctly parametrized your Actions.
- check if "everything goes as expected" in general.

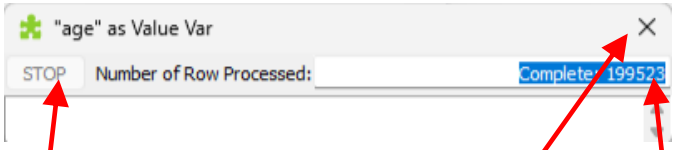
The "Data-Preview" allows you to visually inspect row-by-row all your tables but, sometimes, this is not enough: You want to have a visual summary of the content of a column (just to be sure this column

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

does not contain a small quantity of absurd values). You can easily get this summary by right-clicking the header of a column: For example, when I right-click the header of the column “Age”, I obtain:



When I select inside the above context menu the option “Create Chart as Value var”, I obtain a new window:



From this point, I can either:

1. Completely abort the chart computation: Click the ✕ button
2. Click the “STOP” button to compute the charts on the rows that have already been processed. The quantity of rows already processed is displayed here: “199523”
3. Simply wait for the complete table to be processed.

To start a new “chart computation”, you don’t need to wait for the complete table to be computed (i.e. you don’t need to wait for the data-preview status to reach “Complete”): You can directly start the computations while the status is still on “Fragment”. This allows you to quickly produce some charts on the first few rows of your table: No need to wait!

The charts are produced as MS-Word and/or MS-Excel documents.
Here is an example on the classical “census-income” database. Let’s assume that:

- we defined as “Binary Target” the column “Taxable Income Amount”.
- we asked to see the column “age” as a Value variable.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Then, we obtain:

TIMi - *The Intelligent Mining Machine* Audit report v12.11

General Counts

```
Number of rows in dataset    = 199523
Target variable              = taxable income amount
Number of rows in Target     = 12382 ( 6.21 %)
```

Variables Listing

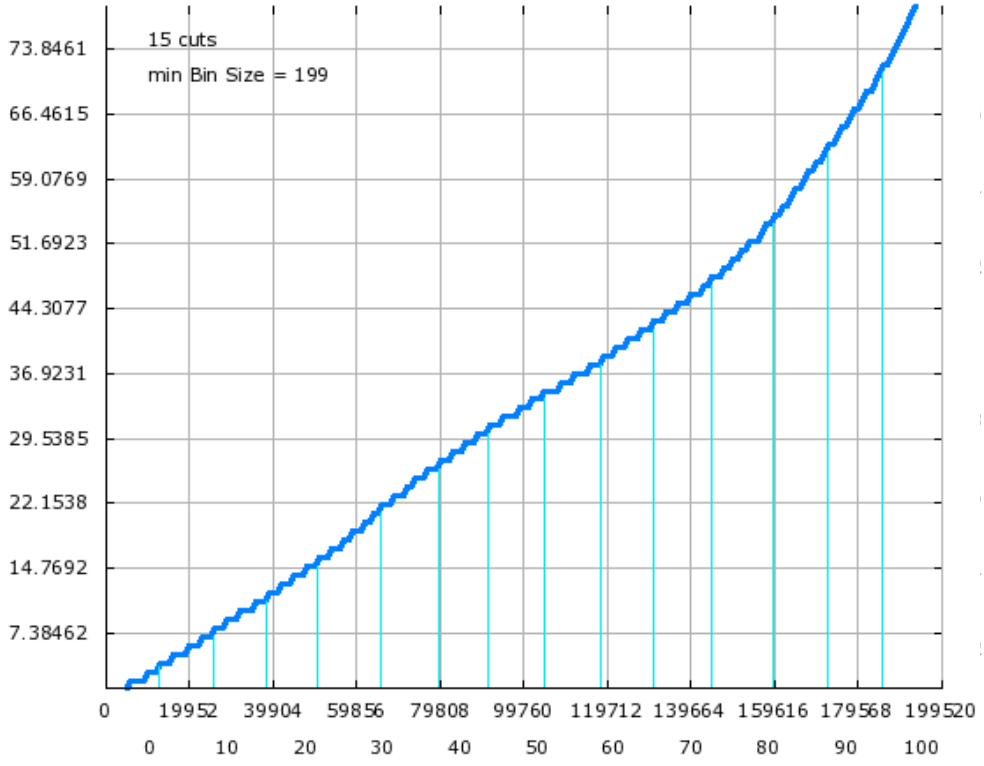
The final status before producing the reports is displayed here. This status can be:

- "Complete" (the charts are based on the whole data table)
- "Fragment" (the charts are based on the first few rows of the data table).

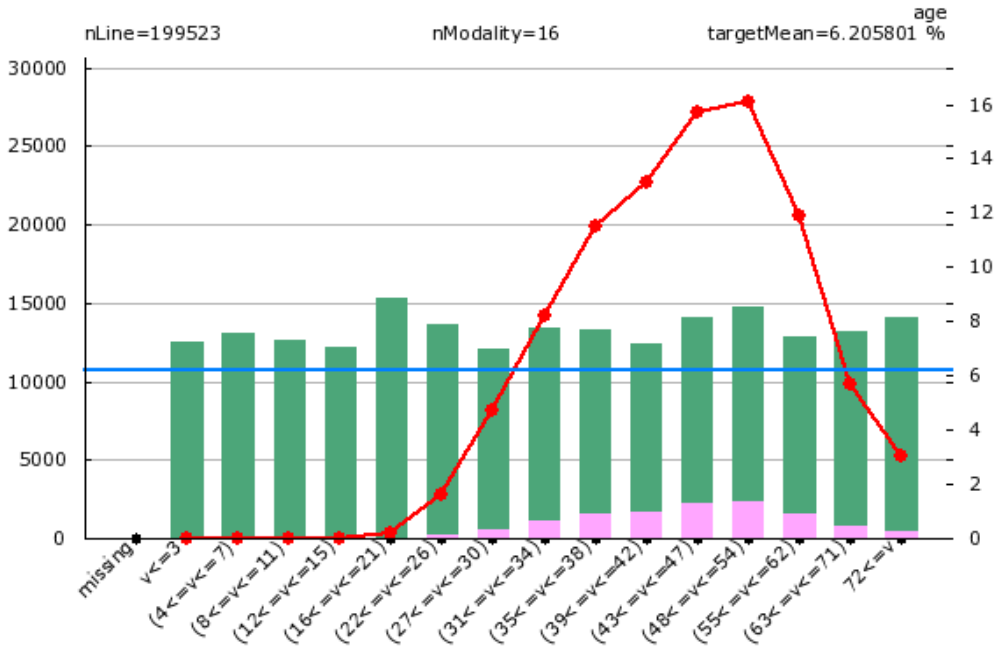
CONTINUOUS Variables

1) age (Complete)

Median= 33
Mean= 34.4942
stdDev= 22.3109
Min= 0
Max= 90
Number of different values= 91
Max Quantile limit= 40



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



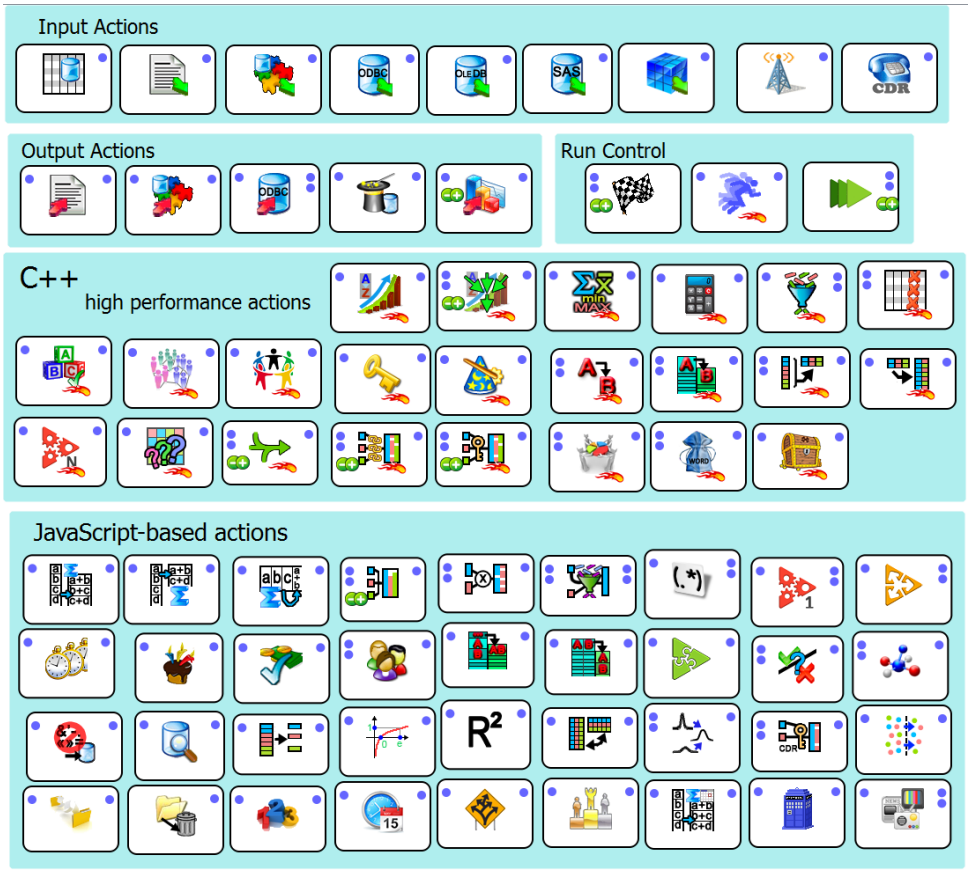
Univariate AUC= 57.11 %			Count Dataset	% DataSet	Count Target	Proba Target	% Index
6.3 %- 6.3 %(	v<=	3)	12492	6.3	0	0.0	0.0
12.8 %- 12.8 %(	4<=v<=	7)	13039	6.5	0	0.0	0.0
19.1 %- 19.1 %(	8<=v<=	11)	12611	6.3	0	0.0	0.0
25.2 %- 25.2 %(	12<=v<=	15)	12206	6.1	0	0.0	0.0
32.9 %- 32.9 %(	16<=v<=	21)	15323	7.7	28	0.2	2.9
39.8 %- 39.8 %(	22<=v<=	26)	13642	6.8	220	1.6	26.0
45.8 %- 45.8 %(	27<=v<=	30)	12024	6.0	566	4.7	75.9
52.5 %- 52.5 %(	31<=v<=	34)	13368	6.7	1097	8.2	132.2
59.2 %- 59.2 %(	35<=v<=	38)	13358	6.7	1534	11.5	185.0
65.4 %- 65.4 %(	39<=v<=	42)	12387	6.2	1629	13.2	211.9
72.5 %- 72.5 %(	43<=v<=	47)	14133	7.1	2223	15.7	253.5
79.9 %- 79.9 %(	48<=v<=	54)	14741	7.4	2374	16.1	259.5
86.3 %- 86.3 %(	55<=v<=	62)	12830	6.4	1528	11.9	191.9
92.9 %- 92.9 %(	63<=v<=	71)	13245	6.6	753	5.7	91.6
100.0 %-100.0 %(	72<=v	)	14124	7.1	430	3.0	49.1

Please refer to the “TIMiQuickUserGuide.pdf” for more information about the charts displayed inside this summary report.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.2. Anatella Actions

Inside Anatella, all the data transformations are represented as “small boxes”, also called “Actions” or nodes. Anatella-Actions always operate on tables: an Action takes one or several tables in input, performs some data-transformation, and gives one or several tables as output. All the Actions are accessible to the user from the right-side panel in the main Anatella Window. Here is a screenshot of some of the currently available Actions:



In the next sub-sections, we will study the graphical representation of an Action inside an Anatella—Data-Transformation-Graph.

4.2.1. Input / Output Specifications

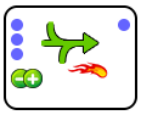
An Action takes one or several tables as input, performs some data-transformation, and gives one or several tables as output. The number of input & output tables of an Action is graphically represented by the number of small blue dots on the left & right of the Action. These blue dots are named “pins”.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example:

- Consider the following action:

  This Action takes one table as input, as represented by the unique pin on the left side of the Actions. This Action outputs two tables, as represented by the two pins on the right side of the Actions.

- Consider the following action: 

  This Action takes 3 tables as input, as represented by the 3 pins on the left side of the Actions. This Action outputs one table, as represented by the unique pins on the right side of the Actions. The number of input table is not fixed and can be changed by clicking on the and icons.

4.2.2. Computing speed of the Action

Inside Anatella, some actions are highly-optimized action coded for speed in C++. For example:

- This Action is optimized for speed in C++:
- This is a "normal" Action coded in JavaScript: , in R: or in Python:
- This is a (sub)Graph:

The Speed-Optimized-Actions are the ones developed in C (the ones that don't have any Javascript , R or Python logo). These optimized are running up to 100 times faster than the equivalent "normal" Actions developed in JavaScript/R/Python. See the section 9 to know how to create your own Javascript/R/python Actions.

The triple triangles icon means that this is actually not a single Action that will be executed but rather a complete (sub)Graph that might include many Actions (…and thus incur a very long running-time). See the sections 10.6. and 5.3.3. for more details. For example, when Anatella executes a (sub)Graph, you might see something like:

4.2.3. Breakpoints

Anatella include a powerful debugger that allows you to "debug" the Actions based on JavaScript. The debugger starts only when you have placed some "break points" inside the "JavaScript source code". This is illustrated in this way:

- This Action contains some **active** break points and the debugger will run:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- This Action contains some **disabled** break points:

The Anatella-debugger has one small limitation: You can debug only ONE Action at the same time. The Action that will be debugged is the one with the **active** break points.

4.2.4. Actions being edited

When you double-click on an Action, you obtain a window that allows you to set the different properties of the Action. Inside the Anatella-Graph window, you can always visually check which action is currently being "edited": it's the action with a green border. For example:

- The properties of this action are been edited:
- This action is not currently edited:

4.2.5. The Output pin being inspected

When you click on one output "pin" of an Action in "Run mode", you directly see in the data-preview-window the data-table that is "outputted" on this pin. This is very useful when testing & debugging your transformation. Inside the Anatella-Graph window, you can always visually check which data-table you are viewing in the data-preview-window: it's the table that corresponds to the pin with a green border. For example:

- This green ouput pin is the pin that is currently shown inside the Main Anatella data-preview Window:

4.2.6. Hard Drive Cache / Gel Files

Let's assume that you are developing a very large transformation-graph containing dozens of Actions. One of these Actions requires extra-care because its parameters needs extra-tuning. You are forced to run several time this "difficult" Action before obtaining the correct values of its parameters. Let's assume that this "difficult-to-tune" Action occurs at the "end" of the transformation graph: i.e. each time you want to test a new set of parameters for this "difficult" Action, you are forced to wait for a very long time because there are many, many other actions to run before executing the "difficult" action with the new parameters to test. In this somewhat common situation, it would be nice to "Save on the Hard Drive" a backup of all the input tables of this "difficult" Action. In this way, the next time that you run the "difficult" Action (with another set of parameters to test), you can directly use the backed-up tables, instead of re-running the whole transformation graph from the start. This "Hard Drive Backup" is commonly named a "HD (Hard Drive) Cache" or "Anatella Gel file". An "Anatella Gel file" contains a complete "snapshot" of one table, ***frozen*** in time (...hence the name "Gel file" – gel means "freeze" in French).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The “Gel files” are extremely useful when developing new large and difficult transformations because it allows you to test very rapidly (nearly in real-time), many different parameters of your “difficult-to-tune” Actions.

Thanks to the unique automatic “HD cache” feature of Anatella, developing new complex transformation graphs is a “Child play”! The development time of large transformation-graph is usually divided by 10, compared to other ETL.

Inside the Anatella-Graph window, you can visually check which output pins have a corresponding “HD Cache”/”Gel File”: these are the output pins with an orange border: for example:

- This (orange) output pin possesses an “Anatella Gel File”:
- This (green) output pin possesses an “Anatella Gel File” and the content of this file is visible inside the data-preview-window:
- This (red) output pin does not accept any “Anatella Gel File” (any “HD Cache” files will be deleted as soon as the graph starts. This happens when using the advanced multi-threading execution techniques: see section 5.3.2.4):

When you are in “Run Mode” (i.e. when the button in the main toolbar is “checked”), you can:

- Create, for (nearly) any output pin, a new “Gel File”: Simply click on the required output pin.
- Click on any orange pin and instantaneously see the corresponding data-table in the data-preview-window. The preview is based on the available “Gel File” and appears instantaneously (it does not require any computation at all).

4.2.7. Action Status (OK/Error/Warning)

If an Action fails (because, for example, the Action parameters are invalid), then the background color of the action becomes light-red: for example:

- This action failed: (background color is light **red**)
- This action has some warnings: (background color is **yellow**)

You can check the “Execution log” to know the precise nature of the failure or warning.

4.3. A second example.

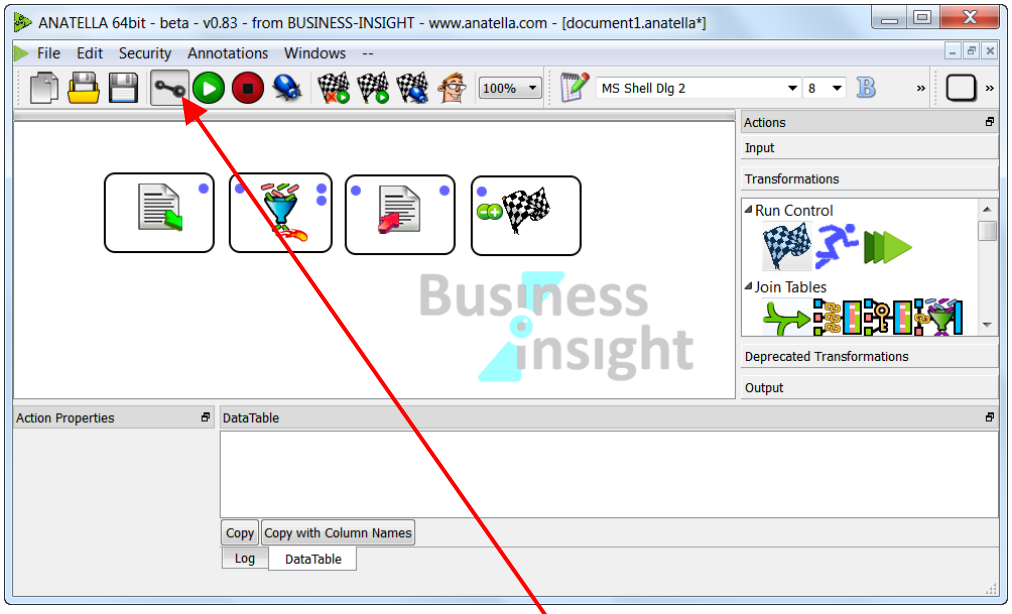
We will create an Anatella-Graph that reads a CSV file, filter some rows out of the CSV file and write back on the Hard Drive the “filtered” CSV file.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Add inside the Anatella-Graph, the following Actions:

- CSV file reader :
- Row filter:
- CSV file writer:
- Global Run Flag:

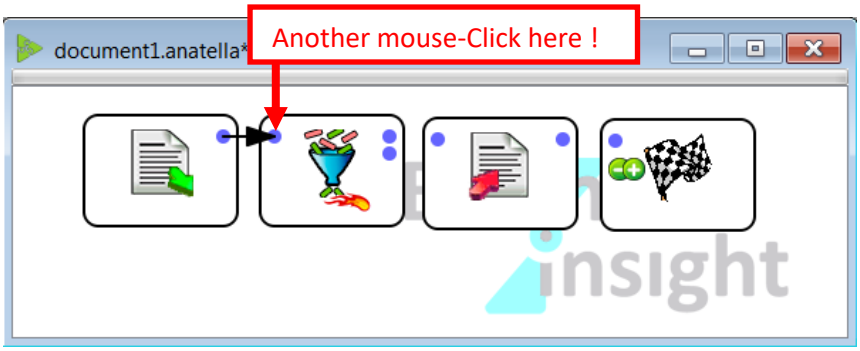
You should obtain something like this:



If required, click on the icon in the main toolbar: We are now in “Connect mode”: We will connect the different Actions together. Let’s do a simple left-click on the output pin of the “CSV file reader Action”: A red line appears:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Now, click on the input pin of the "Row filter Action":



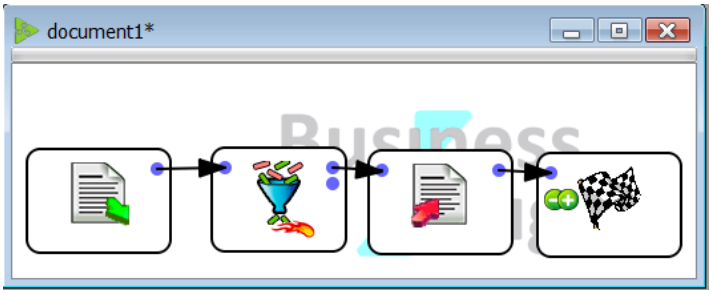
We just created a new arrow that means: "*The table that is coming from the CSV file will be injected into the Row filter Action.*"

TIP: You can use the "middle-mouse-click" to easily switch between "Connect mode" and "Run mode".

TIP: When you are drawing a new Arrow, you will notice that there is a "red line" that is following your mouse: This "red line" is a "preview" of the new Arrow that will soon be added to your Anatella-Transformation-Graph.

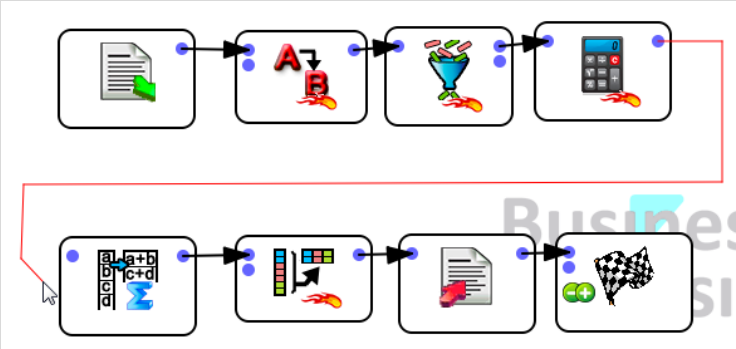
TIP: To cancel drawing a new arrow (to remove the "red line" that is always following your mouse): you can:
* Press the Escape Key.
* Right-click several times until the "red line" disappears.
* Middle-mouse-click (this will also switch to "Run mode").

Let's connect all the actions together. We obtain something like this:

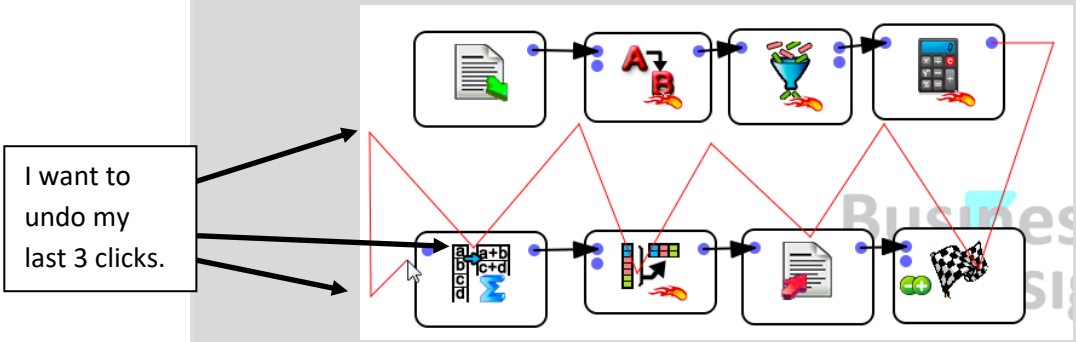


TIP: The arrows that are connecting the Anatella actions are not always "straight lines": You can give to these arrows very complex shapes: When the "red line" is visible, left-click anywhere inside the graph window to "shape" the arrow in any shape that you want. In the following small example we are in the process of creating a new arrow with a complex "S" shape:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

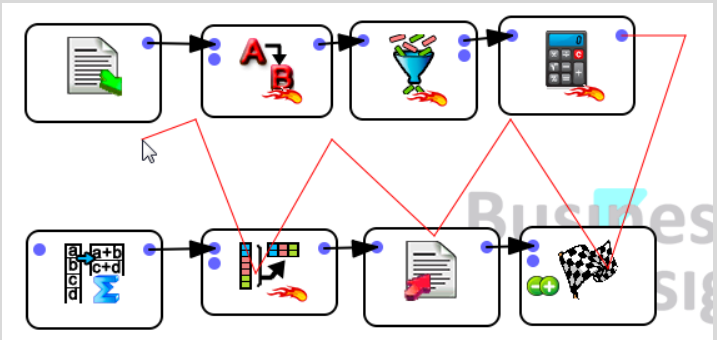


It can happen that sometime, you left-clicked several times at several wrong locations on the graph and you obtained a "wrong arrow shape": For example:



This is not a problem: you can always RIGHT-click to undo any LEFT-click that you made. Using the RIGHT-click mouse button, you can easily correct the current "arrow shape" to easily obtain the desired result.

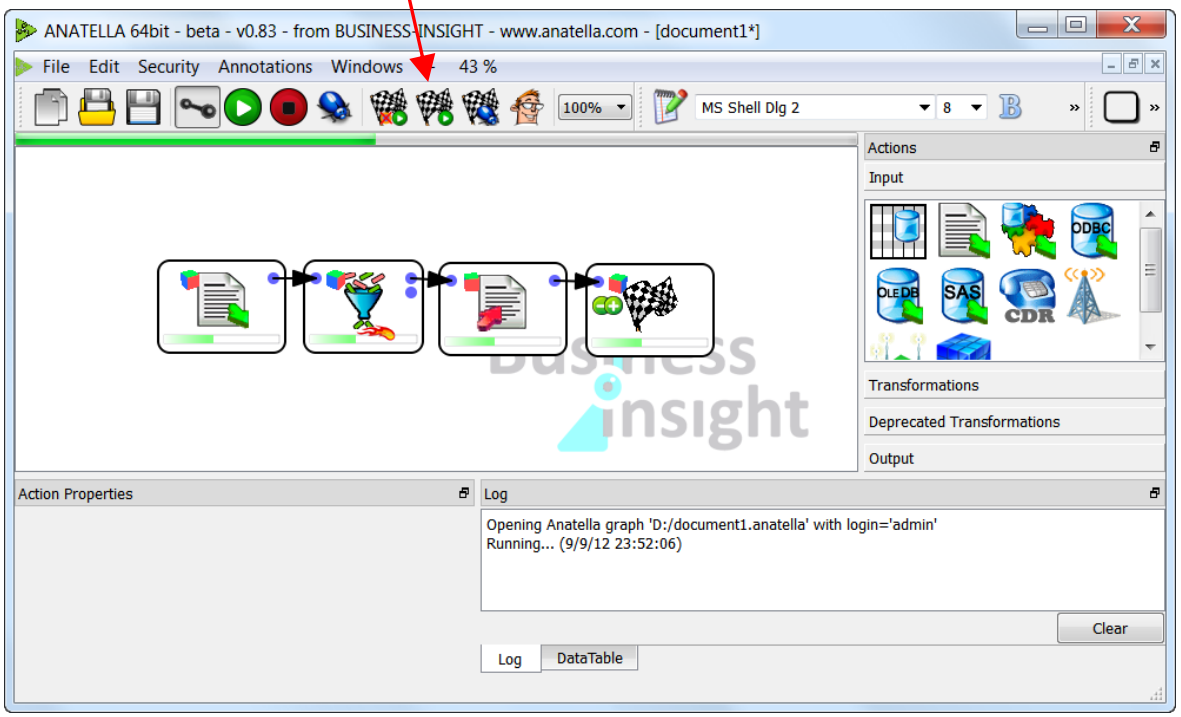
For example, if I RIGHT-click 3 times, then I undo the 3 last LEFT-clicks and I obtain:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You will find detailed explanations in the sections 5.2.1, 5.5.7 and 5.26.1 on how to configure the "CSV file reader Action", the "Row filter Action" and the "CSV file writer Action". See the next section to have more information about the last action used: the "Global Run Flag Action".

When you click on the button inside the main toolbar of Anatella, you execute the Anatella-graph: The below example creates a new CSV file that only contains "filtered rows". Here is a screenshot of an execution example:



4.4. Execution of Anatella-Graphs.

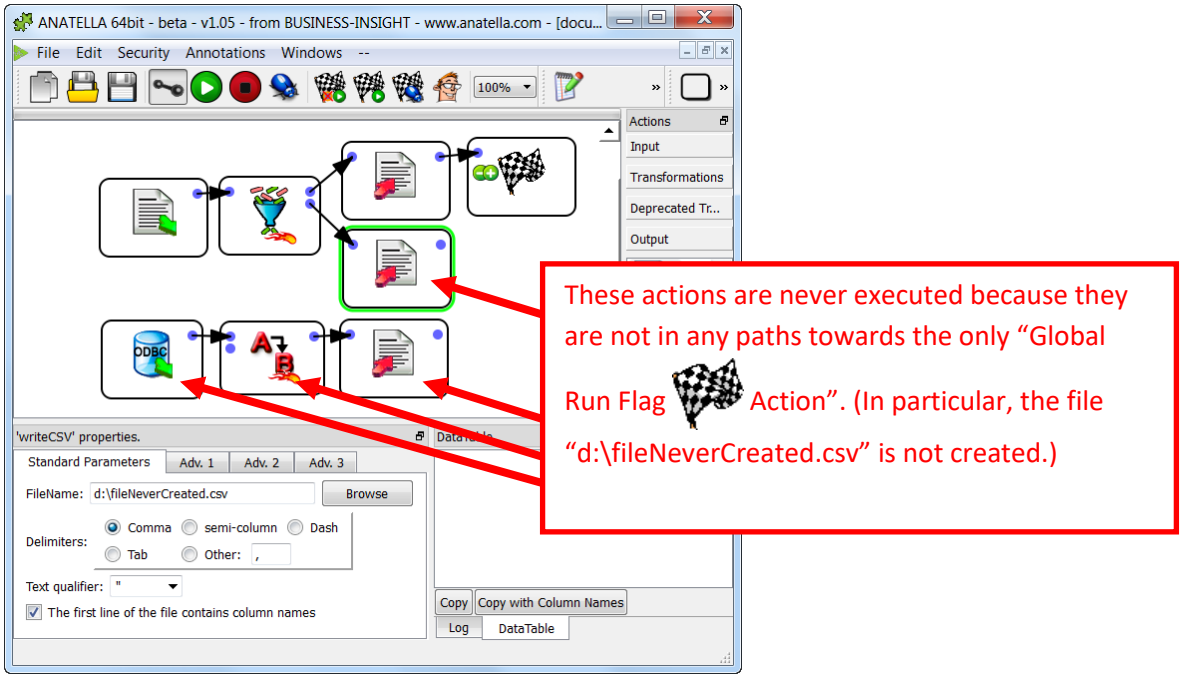
4.4.1. Execution of Anatella-Graphs: The "Global Run Flag Action"

WARNING! This section contains vital information for the correct usage of Anatella.

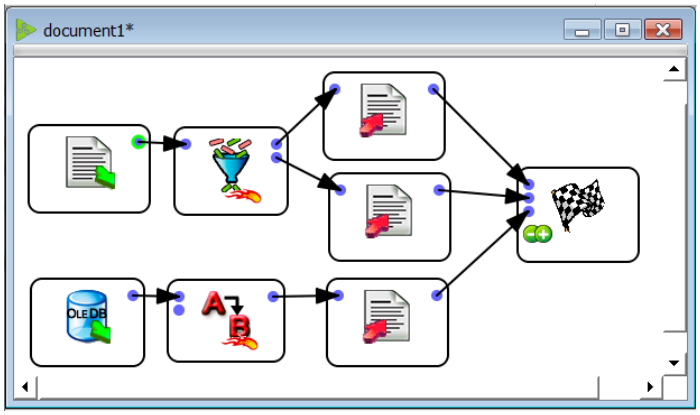
The "Global Run Flag Action" defines which part of the script will be executed when you click on the button inside the main toolbar of Anatella.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's give more details, using another small example:

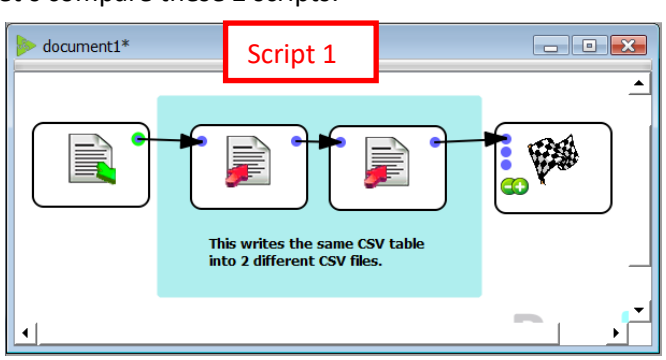


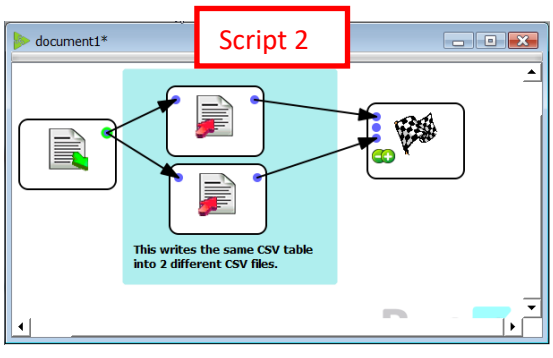
In order to run all the Actions of the above Anatella-Graph, each final node must be connected to the Global Run Flag, as detailed below:



One final remark about the optimization of the execution-speed of Anatella-graphs.

Let's compare these 2 scripts:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Inside “script 2” (on the right), the table available on the output pin of the “CSV file reader Action” needs to be buffered on the hard drive (because this table must be “sent” into several different Actions) before executing any of the other actions. This extra disk I/O (hard drive buffering/“Gel file” creation) is not performed in “script 1” and thus “script 1” is significantly faster than “script 2”. More information about this subject in section 5.6.

4.4.2. Execution of Anatella-Graphs: The “Run Mode”

When you are in “Run Mode” (i.e. when the button in the main toolbar is “checked”), each time you click on an output pin, Anatella runs the graph up to this point and also creates the corresponding “HD Cache” file.

All the “HD Cache” files have the file-extension “.gel_anatella”. Anatella always uses the “.gel_anatella” file format (and not the “.cgel_anatella” file format) to create the “HD Cache” files (because the “.gel_anatella” file format is a row-based file format that is optimized to read&write all the columns from a table. …And reading&writing all the columns is exactly what we are always doing when using “HD Cache” files).

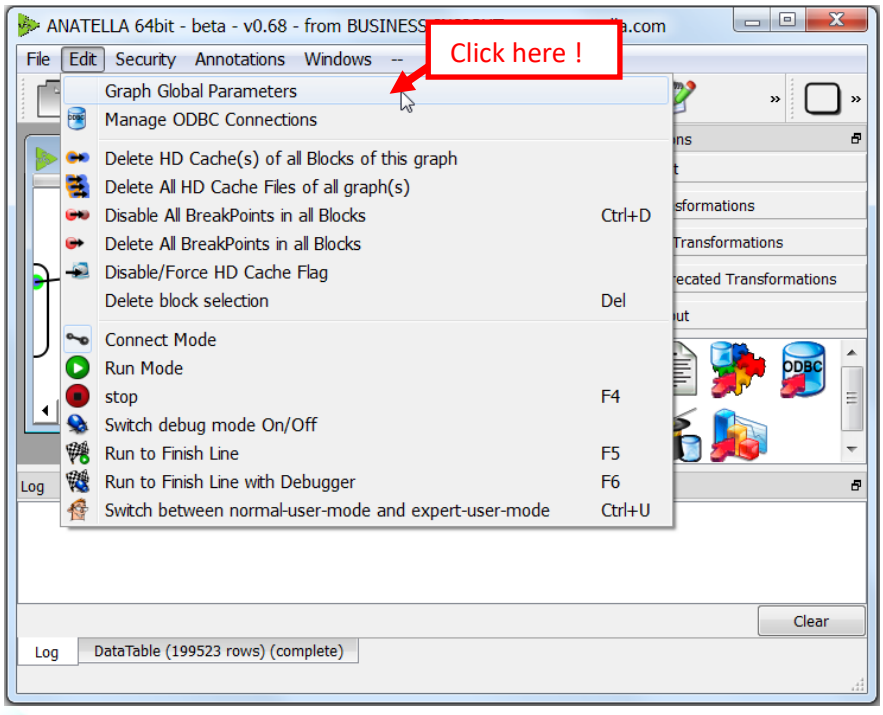
4.5. Working with the HD cache

Inside Anatella, the HD cache has 3 usages:

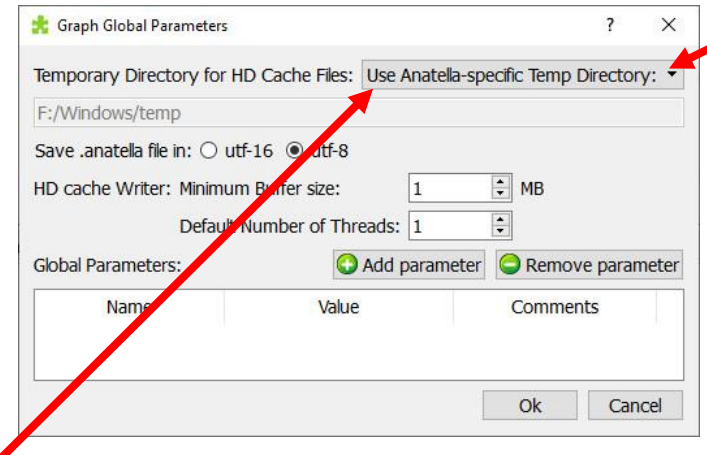
1. Improving execution speed when developing new Anatella-Graphs: more explanations at this subject in section 4.2.6.
2. A “HD cache”/ “Gel file” is required to be able to display the “preview” data-table inside the Anatella main windows.
3. In some situations, a “Gel file” is required for the “normal execution” of the Anatella graph. See for example the section 5.6 (or the “Script 2” in the section 4.4.1, hereabove).

4.5.1. Selecting the directory containing the “HD cache” / “Gel files”.

The directory that contains all the “Gel files” of the current Anatella-Graph can be specified inside the “*Graph Global Parameter*” window. To see this window, click here:



The directory containing all the HD cache files of the current Anatella-Graph is specified here:



You can change the directory containing all the HD cache files of the current Anatella-Graph by clicking here: . There are 3 options:

- **Use "Anatella-Specific Temp directory".**

  This is the default value.

  Each different user on the server can chose the location of its "Temp Directory for HD Cache file" in the "*Anatella Glogal Settings*" window: see section 7.1. for more details on this window.

  No administrative rights are required to change this value.

  By default, the location of the "*Anatella-Specific Temp directory*" parameter is initialized to be the same location as the location that was stored inside the "%TEMP%" MS-Windows Global Environmental Variable when you ran Anatella for the very first time.

- **Use "Global OS Temp Directory".**

  Use this option when you want Anatella to always use the value of the "%TEMP% Environmental variable as the directory containing all the HD cache files.

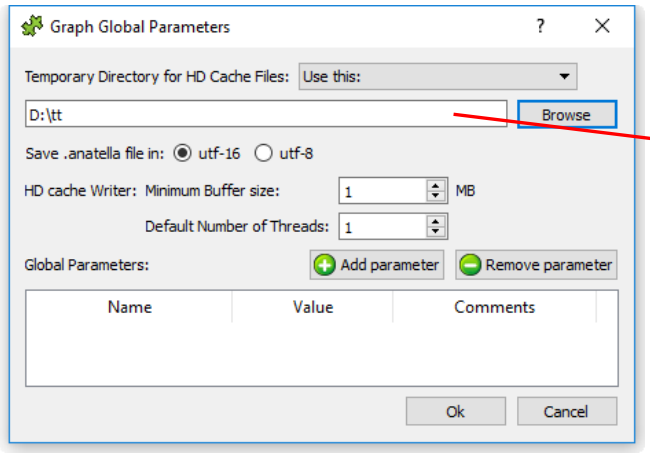
  You need administrative rights to define the value of the "%TEMP% Environmental variable. You should ask to your administrator to select carefully a nice location on your server. You need a location with a fast read/write access (an SSD is great).

  Actually, the "%TEMP% Environmental Variable contains the location of all the temporary files of the current user. It's always a good idea to set this location to a drive with plenty of space left (…and this is *always* a good idea to make this change even if you are not using Anatella). See the section 7.1.1. to know how to change the "%TEMP%" MS-Windows Environment Variable.

- **Use this:**

  Some very specific Anatella data transformation graphs are using so much disk space that you want a very specific location for the directory containing all the HD cache files, only for these very specific graphs. In such situation, you can manually edit the location of the "*Working directory for cache files*" to set it to a directory within a drive with plenty of space left. For example:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



The "*Working directory for cache files*" is set to "d:\tt" for the current graph **(and only for this one)**.

When you change the settings inside the "*Graph Global Parameter*" window, it only changes the "*Temporary Working directory for cache files*" for the currently edited graph: i.e. the value of this setting for of all the other graphs (newer of older) stays the same. The section 7.1. explains how to change the value of the "*Temporary Working directory for cache files*" setting for all the .anatella files inside the server.

4.5.2. Manual creation of "HD cache"/ "Gel files"

When you are in "Run Mode" (i.e. when the button in the main toolbar is "checked"), each time you click on an output pin of an Action, Anatella creates the corresponding "HD Cache file" (and, Anatella also displays the content of the "cached" table inside the "data-preview" table).

Regarding "HD Cache File" and the "data-preview" table, the output pin of the WriteGel Action has a special behaviour: See section 5.26.2. for more information about this subject.

4.5.3. Forcing the automatic creation of a "HD cache"/ "Gel files" at each run

By default, when you run an Anatella-Graph, only the minimum number of "HD cache files" is created.

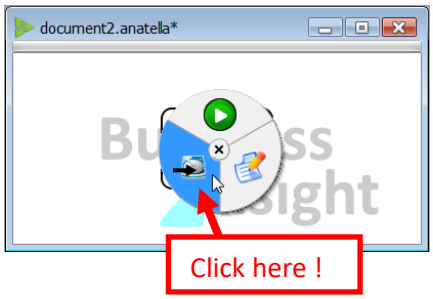
For debugging purposes, it could be interesting to force Anatella to always create, at every run, "Gel files" at specific pins inside the graph (even if these pins do not require any buffering for the graph to be properly executed). Thereafter, you can easily "inspect" all the pins that possess a "Gel file" by clicking on these pins (to "inspect" a pin, you must be in "Run Mode": i.e. the toolbar button must be checked).

To force Anatella to always create a "HD cache"/ "Gel file" for a specific Action:

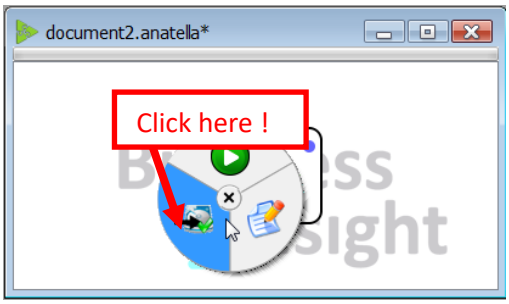
1. Right-click the Action: a circular context-menu appears.
2. Select the icon.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

See the illustration here:



To remove the “Force HD cache flag”, repeat the same procedure:



Please note that the clicked context-menu icon in the above two examples is not the same: this icon gives you the status of the “Force HD cache flag”:

- : This means that the “Force HD cache flag” is currently off. If we click this icon, we enable the “Force HD cache flag”.
- : The flag is ON. The small sign means that Anatella is currently always creating a “HD cache”/“Gel file” at each run of the Anatella-Graph. If we click this icon, we disable the “Force HD cache flag”.

4.5.4. Automatic deletion of “HD cache”/“Gel files”

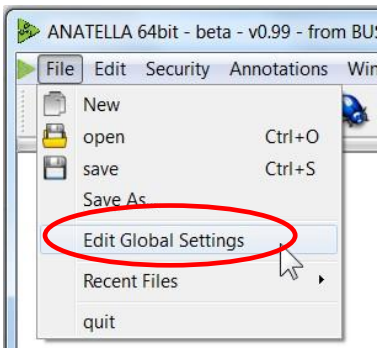
All “HD caches”/“Gel files” are the result of the execution of an Anatella-Graph and are similar in essence to temporary files. As such, they depend on the (possibly many) parameters of the different Actions in the Anatella-graph. When you change the parameters of a Node/Action in the graph, Anatella automatically checks if this change invalidates any of the currently computed “Gel files”, and delete all temporary files that no longer hold current data specifications. Basically, any change “above the stream” will result in the deletion of temporary files “below the stream”.

As explained in the previous paragraph, a “HD cache”/“Gel file” is closely linked to the set of parameters used inside the corresponding Anatalla-Graph. Let’s take two small examples:

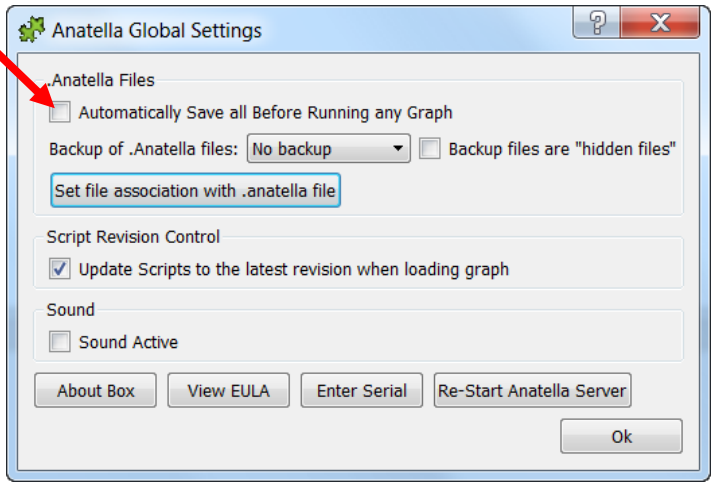
- First example:
  - At time T=1, you create some “HD cache”/“Gel files”.
  - At time T=2, you save your Anatella Graph and close the Anatella application.
  - At time T=3, you re-open your Anatella Graph. Anatella is able to retrieve the “HD cache”/“Gel files” that are matching the parameters included inside the loaded Anatella-Graph. Everything works smoothly: all the “Gel files” previously computed are available for direct usage and you can start working immediately.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Second example:
  - At time T=1, you de-activate the option "*Automatically Save all before running any graph*". This option is inside the Anatella "Global Settings" window. To get this window, select the "*Edit Global Settings*" option inside the "*File*" drop-down menu:



…and click here:



  - At time T=2, you save your Anatella Graph.
  - At time T=3, you change some parameters inside your Anatella Graph.
  - At time T=4, you create some "HD cache"/ "Gel files" (i.e. you run your graph). Since the option "*Automatically Save all before running any graph*" is disabled, Anatella did NOT save the latest version of your graph when you started running your graph.
  - At time T=5, you close the Anatella application.
  - At time T=6, you re-open your Anatella Graph. Anatella is NOT able to find any "HD cache files" that are matching the parameters included inside the loaded Anatella-Graph (because you still changed some parameters at T=3 **after** saving the graph at T=2). Anatella must re-compute all the required "Gel files", consuming time and CPU resources.

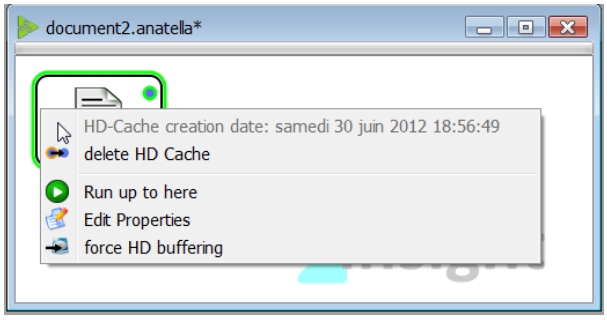
To summarize: Once you have computed some time-consuming "HD cache"/"Gel files", save your graph (i.e. press "ctrl-s"). In this way, when you re-open later your Anatella-graph, Anatella will be able to retrieve all your precious "Gel files" and you will be able to start working immediately.

If you made a mistake, don't run your graph (because this will save it), **close without saving** and re-open the latest saved file.

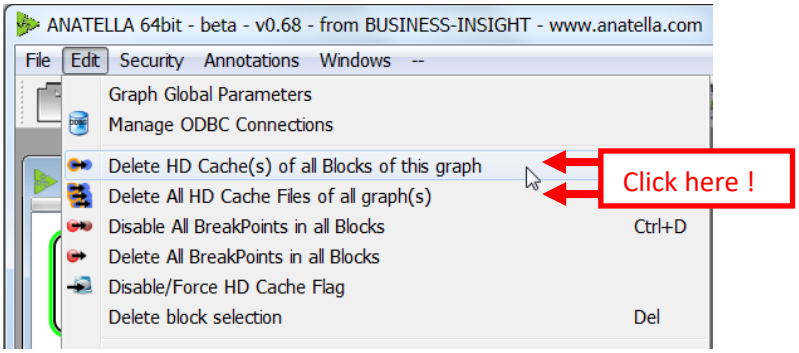
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.5.5. Manual deletion of "HD cache"/ "Gel files"

To delete an "HD cache"/"Gel file" for a specific Action, right-click the Action and select the "*delete HD cache*" option inside the context-menu, as illustrated here:



You can also use one of the two options available inside the "*edit*" drop-down menu: See the following illustration:



The option "*Delete All HD Cache Files of All graphs*" deletes all the HD Caches that are inside the "*Working directory for cache files*" (see section 4.5.1. for more information about this parameter). If all the users on the server are using the same "*Working directory for cache files*" (i.e. they have the same %TEMP% MS-Windows Environment Variable), then you should avoid using the option "*Delete All HD Cache Files of All graphs*" because this might delete the HD Caches of the other users on the server (and they won't like it!).

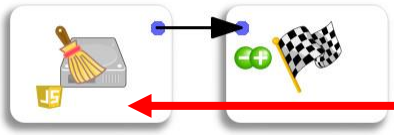
4.5.6. Cleaning-up old and un-used "HD cache"/ "Gel files"

The "HD cache"/"Gel file" that are automatically generated during a batch execution (see section 4.7 about "batch execution") are always destroyed at the end of the execution of the graph. In this way, a "production server" that is running many .anatella files in batch always stays "clean".

The "HD caches"/"Gel files" that are created "on request" (i.e. by clicking on an output pin of an Action in "Run Mode") can be destroyed "in bulk" using the option "*Delete All HD Cache Files of All graphs*". However you should avoid using this option on a server with many different concurrent users (see previous section about this subject).

To prevent the users on the server to leave many old cache files (that are no longer of any utility), we can run in batch, every week, this simple .anatella graph (see section 4.8. to learn how to schedule a graph to run it at regular intervals):

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



This is the “cleanTempDir” Action available inside the “System Tools” category (See section 5.20.2. for more information about this Action).

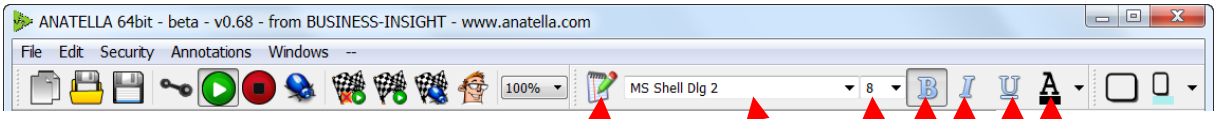
The above graph will to destroy all the really old (and thus certainly un-used) HD cache files.

If you are using different “*Working directory for cache files*” parameters for each different user (i.e. you have a different %TEMP% MS-Windows Environment Variable for each different user), you need to parametrize the deleteGelFiles Action to run it on the different “*Working directory for cache files*” of the different users.

4.6. Working with Anatella-Graphs

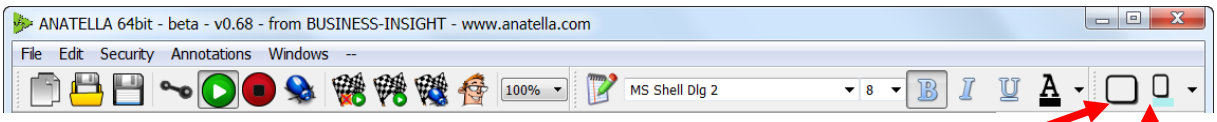
4.6.1. Graph annotations

You can add some annotations on your Anatella-graphs. These annotations are very helpful when sharing an Anatella-transformation-graph with some colleagues.



To add a text-comment : click here and then, just after, click in the graph at the position where you want to insert the comment.

You can change the position, the font type, font size, bold, italic, underline and font color of the text comments.

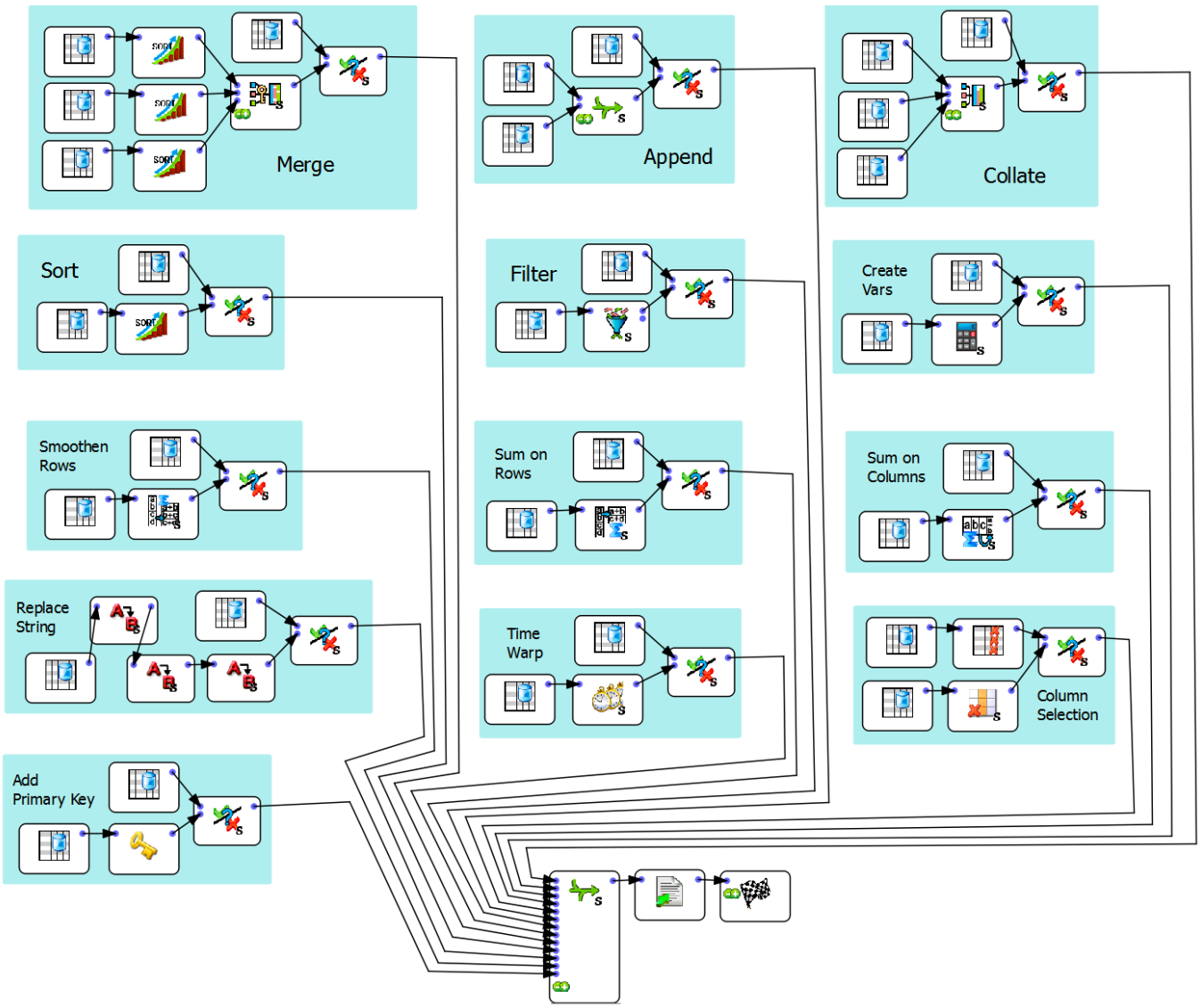


To add a group-box: click here and, then, just after: click two times (at two different positions) in the graph to define the position of the group-box.

You can change the position and background color of the group-boxes

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is an example of a nicely annotated Anatella-graph:



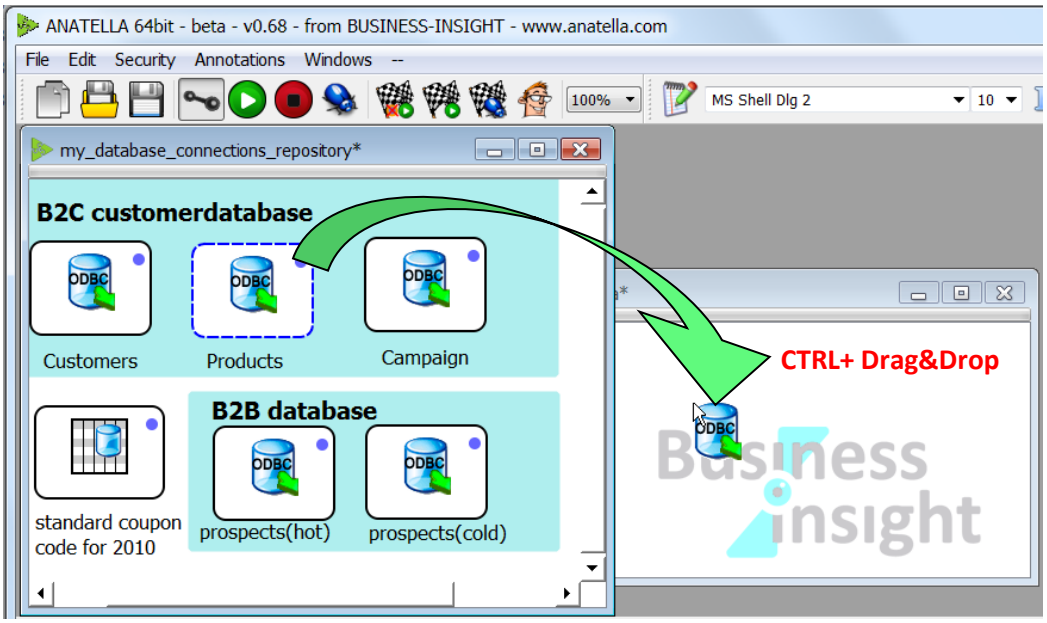
4.6.2. Duplicating Actions

By default, when you drag an Action with your mouse, it simply moves the Action on the Graph. If you are maintaining the CTRL-key pressed and you start "dragging" an Action, you create a duplicate of the Action. You can drop your duplicated Action into another Anatella-Graph: This allows you to transfer Actions from one Anatella-Graph to another.

This system makes it easy to store all your database connections settings. You can have a "reference" Anatella-Graph that contains one different Action per each different database connection that you regularly use. When you need a connector to a specific database, you simply "transfer" the associated Action from your "reference" Anatella-Graph.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

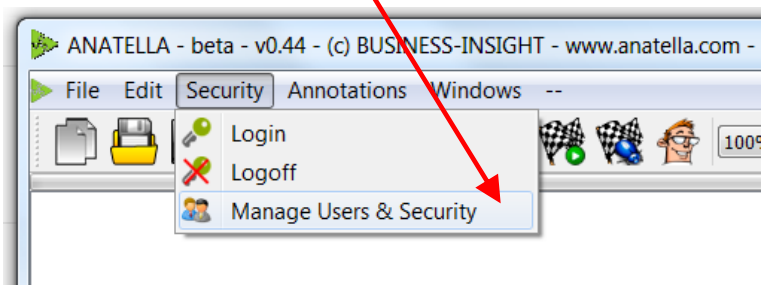
For example, consider the following database repository:



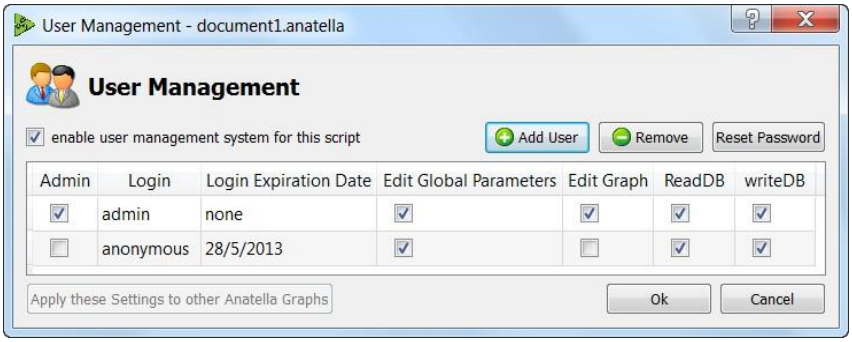
4.6.3. Security – User management system – Graph Encryption

The "Server" version of the full "TIMi Suite" contains a complete AMS (i.e. an "*Account Management System*") that allows you to manage in a central way all your user's credentials.

The Anatella software also contains, in addition to the "*TIMi Suite AMS*", a simple security management system. This system is not centralized: Each Anatella file contains its own set of users and credentials. To access the user management screen of a specific Anatella graph, click the option "Manage Users & Security" and the drop-down menu "Security":



The "User Management" window appears. Click on the "*Enable user Management for this script*" checkbox, you should obtain:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

By default, any "anonymous" user:

- Can run (from the command-line) this Anatella transformation graph provided that the current date is not after the expiration date of the "anonymous" user: on the above screenshot: the expiration date is "21/6/2012"). Please refer to the section 4.7. to learn how to run Anatella graphs from the command-line.
- Cannot edit (nor see the Anatella graph).
- Cannot change the security setting (i.e. he has no "administrator" privileges)

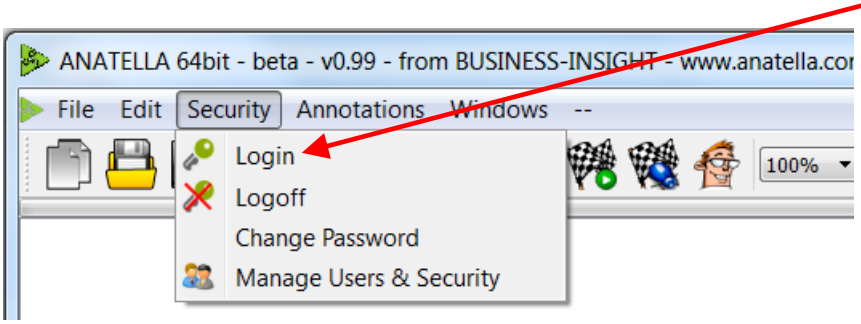
Optionally, you can also add new users (of this graph) with different user rights.

Once you have modified the security settings of an Anatella graph, don't forget to save your Anatella graph (press CTRL+S), otherwise your modification will be lost.

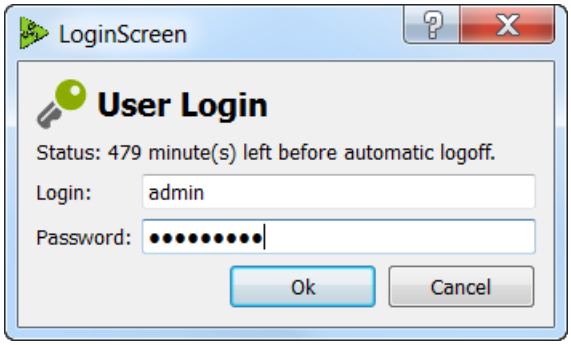
The "anonymous" user is a special case because it can't have any password. This means that anybody can run a graph as an "Anonymous" user (as long as the expiration-date is not overdue). However, editing a graph anonymously (i.e. seeing how the graph is made) is only possible if the "Edit Graph" check box is enabled.

When the "*user management system*" is active, the Anatella files are not simple human-readable XML files anymore: The Anatella files become 100% encrypted so that only the users with "*edit graph*" privileges can "*see the content*" of the Anatella-transformation-graph. You can revert to simple XML file by temporarily de-activating the "*user management system*" (when re-activating the "*user management system"*, all your security settings will be restored).

Use the "Login screen", to identify yourself to obtain the rights to administrate, edit & run your Anatella transformation graphs: click the "Login" option in the "Security" drop-down menu:



You should see:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When you select the "*Change Password*" option inside the "Security" drop-down menu, you will change the password of some user for the currently edited graph only. If no graph is currently begin edited, this option is disabled.

4.6.3.1. Why encrypt .anatella graphs?

There are several reasons to encrypt .anatella graphs:

1. Let's assume that you are an independent consulting company that is selling a very expensive "Business-Intelligence" solution. Typically, this solution extracts some data from your customer data warehouse, transform it and give some precious & valuable results to your customers. Let's now assume that you developed your solution using SQL scripts to do all the data transformations.

   In such a case, it's very easy to illegally make a copy of your solution: A thief only needs to look inside the log file of the database and copy/paste all the SQL scripts that were executed. Basically, anybody with a little bit of experience in database technology can steal your work (i.e. steal the SQL scripts that are composing your solution).

   In opposition, if you are using (encrypted) .anatella graphs to make all the required data transformations, your customers can only run your solution, but nobody can copy it. More precisely: you can copy the .anatella files but it will be practically impossible to "sell them" because you can't open the .anatella graph to:
   - change the name of the ODBC connections, change the input/output files, etc.
   - understand how it's working.

   To summarize: Anatella encryption allows you to protect your IP ("Intellectual Property").

2. The possibility to set different expiration dates for different users offers new Business opportunities for consulting companies. Consider the following scenario:
   - You developed a proof-of-concept (POC) for a client on the servers from the client (it's easy because you don't need any administrative privileges to run the portable version of Anatella).
   - After the POC, the client hesitates to buy your solution. In the meantime, your solution is still installed on their computer. Without using encrypted .anatella files, anybody (e.g. people from another consulting company) can "have a look" and steal your ideas.
   - The client finally decides to use (and buy!) your solution for one month (in the hope to be able to use it for a longer time, once everything is setup and running). When the contract expires, all the Anatella processes should stop working. To do that with Anatella, you can define a specific user (i.e. with a specific login/password) with a specific expiration date. Then, the client receives only the login/password of the time-limited user. You can even consider **renting** your Anatella-based solution: all you need to do is to define several users with different expiration dates.

3. In large companies, it's important to be able to prevent the new trainees to modify critical business processes. For additional security, such processes could be encrypted .anatella files. Only the people with the sufficient credits (i.e. knowing the login/password) can edit & change the .anatella graphs.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.6.3.2. Bulk Encryption of a large quantity of Anatella Files

For large projects that include hundreds of .anatella files, it's very cumbersome to manually change one-by-one, with your mouse, the security parameters of each of the .anatella files. In such situation, you can use the GraphEncrypt Action. With the GraphEncrypt Action, you can "in batch":

- Add new users of the graphs.
- Remove users of the graphs.
- Easily set the Expiration Dates of each different users in your graphs.
- Change the password of each different users in your graphs.
- Change the user-rights (i.e. Is the user an "admin"? Can he edit the graph? Can he change the Global parameters? Can he access the databases?) of each different users in your graphs.
- Change the "*Working directory for cache files*" parameter in the graphs.
- Change the encoding (utf-16 or utf-8) of the graphs.

4.7. Batch Execution / Anatella command-line

You can execute "in batch" (without any human interaction) any Anatella-graph.

Let's assume that the Anatella executables ("Anatella.exe" or "AnatellaConsole.exe") are inside a directory inside your "PATH" environmental variable (this is the default setting for a standard Anatella installation).
To run an Anatella-Graph "in batch", type:

```
AnatellaConsole.exe "<file_to_execute.anatella>"
```

…or (deprecated):

```
Anatella.exe –e "<file_to_execute.anatella>"
```

...or (deprecated):

```
Anatella.exe –r "<file_to_execute.anatella>"
```

The 2 parameters ("-e" and the <Anatella filename>) must be **the first 2 parameters** on the command line. The "-e" and "-r" parameters are equivalent ("e" stands for "execute" and "r" stands for "run"): A simple log-window will appear (the Anatella graph is thus not visible) (you can hide the log window with the "–s" option) and the graph is executed. The "AnatellaConsole.exe" executable will never create any new log-window: it will use the current shell (or console) window to display all the trace messages.

To run an Anatella-Graph, you can also type (but this is very uncommon):

```
Anatella.exe –i "<file_to_execute.anatella>"
```

The "-i" parameter opens the standard Anatella interface (that you typically use for interactively editing the Anatella graphs), load the graph and run it ("i" stands for "interactive" run). The Anatella graph is visible and you can "interact" with it. There exists very few Actions that allow some interactions while the graph is running (one such action is the "Assignment Solver") and, most of the time, you will rather use "AnatellaConsole.exe" to run your Anatella graphs in a batch, un-attended mode.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When a program terminates, it always returns back to the operating system (i.e. to MS-Windows) a number. This number is named the "error level" of the program. By convention, an "error level" that is non-zero indicates that there was an error during the execution of the program. Anatella extends slightly the convention: More precisely:

- Error Level=0: Everything went **ok**.
- Error Level=1: There were some **warnings** during the execution of the Anatella Graph.
- Otherwise: There were some **errors** during the execution of the Anatella Graph.

The Javascript engine included inside Anatella has some Anatella-Specific extensions. One of these extensions is the ProcessRunner class. This Javascript class allows you to run sequentially or in parallel:

- Any external software (such as curl, scp, cmd, etc.)
- Any Anatella graph (because you only need to run "Anatella.exe" with the proper command-line parameters).

Using the ProcessRunner class, you can access the "error level" of all the programs that have been launched. For example: it's very easy to implement a very complex logic based on the success ("error level"=0) or the failure ("error level">1) of the execution of your Anatella graphs. For a simple example of usage of the ProcessRunner class, see the RunProcesses Action (see section 5.20.3.)

The ParallelRun Action (see section 5.3.3.) also makes extensive use of the "error level" to detect errors or warnings during graph execution.

4.7.1. Re-Defining "Anatella Global Variables" on the command-line.

You can optionally redefine the value of several "Anatella Global Variables" on the command-line. See section 5.1.5. to know more about "Anatella Global Variables".

For example, if we want to define the "global variable" named "outputFile" as "d:\my data\outFile.gel_anatella", we will write (the quotes are required because of the white space character in the filename):

```
AnatellaConsole.exe "<file_to_execute.anatella>" "-DoutputFile=d:\my data\outFile.gel_anatella"
```

If the Anatella-Graph to execute already contains a "global variable" named "outputFile", then the value given on the command-line has a higher priority: Anatella discard the value saved inside the Anatella-Graph file and uses the value of the "global variable" as defined on the command-line.

Another example: If we want to define the "Anatella Global Variables" "A" as "123" and "B" as "456", we will write:

```
AnatellaConsole.exe "<file_to_execute.anatella>" -DA=123  -DB=465
```

4.7.2. Defining Login and Password

When the Anatella "*user management system*" is active, you might need to give your credentials (i.e. your login and your password) on the command-line to receive the rights to run your Anatella graphs. To specify your login and password on the command-line, use the command-line options "-l" (for login) and "-p" (for password).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example:

```
AnatellaConsole.exe "<file_to_execute.anatella>" -lScott -ptiger
```

… will run the given Anatella graph using the login “Scott” and the password “tiger”.

Please note that there are no spaces between “-l” and your login.
Please note that there are no spaces between “-p” and your password.

For increased security, you should only use a login/password without any privileges: i.e. without any “administrator” privilege and without any “Graph Editing” privilege: See section 4.6.3 about Anatella Graph’s Security.

For even greater security you can use a “Professionnal Job Scheduler” that is able to encrypt and hide the command-line (so that only authorized users can see the command-line that contains the login/password).

4.7.3. Silent mode & Quiet mode

You have two solutions to suppresses the creation of any Anatella Log-window (i.e. to run Anatella in “Silent Mode” without displaying any Window):

1. The first (and preferred) solution is to write:
   ```
   AnatellaConsole.exe  MyGraph.anatella
   ```
2. The second (legacy&deprecated) solution is to use “Anatella.exe” with the “-s” command-line option. For example, we’ll have:
   ```
   Anatella.exe –e MyGraph.anatella  -s -te
   ```

The “Quiet mode” is enabled using the “-q” command-line option. When running Anatella in “Quiet mode”, all **warning** messages are removed from the log window. For example, this command-line:

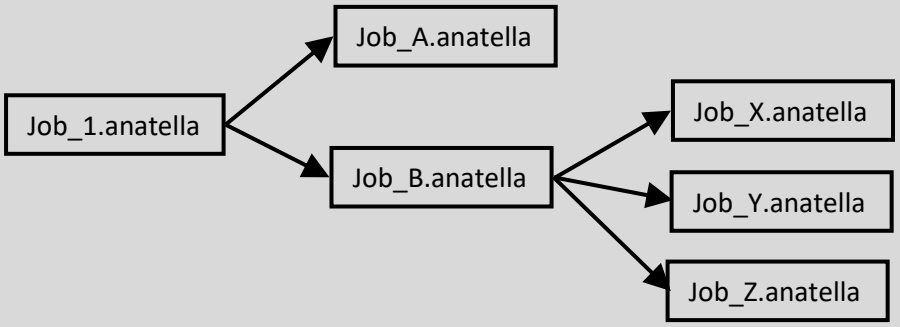
```
Anatella.exe –e mygraph.anatella  -q
```

…will execute the “mygraph.anatella” data-transformation-graph in “Quiet mode”.

The “AnatellaConsole” process is very useful when running Anatella Graphs from another language (i.e. from Python, Jenkins, Php, etc.): see the section 4.9. to know more about this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The "-s", "Silent mode" option "propagates down" to all the called subgraphs. To illustrate this propagation, let's look at the following example:



In the above example, the following command-line:

```
Anatella.exe -e Job_B.anatella  -s
```

…will execute the following data-transformation-graphs in "Silent mode": Job_B.anatella, Job_X.anatella, Job_Y.anatella, Job_Z.anatella.

4.7.4. Generating Trace (log) files

The easiest way to keep Trace/log files is to use Jenkins. When you execute AnatellaConsole from Jenkins the whole content of the log window is always directly saved inside Jenkins and you can always consult the logs easily later.

There are two different command-line options to create Trace files:

1. The "-tf" option.
   The "t" letter stands for "trace" and the "f" letter stands for "fixed name".

   This command-line:

   ```
   AnatellaConsole.exe mygraph.anatella  -tf
   ```

   …will create a trace file (also named log file) named "mygraph.trace" just "next to" the Anatella graph "mygraph.anatella": it will be in the same directory (but with another file extension).

   This command-line:

   ```
   AnatellaConsole.exe mygraph.anatella  -tfd:\mylog.trace
   ```

   …will create a trace file (also called log file) named "mylog.trace" in the directory "d:\"

Please note that there are no spaces between the "-tf" command-line option and the trace filename.

2. The "-ta" option
   The "t" letter stands for "trace" and the "a" letter stands for "auto" naming.

   This command-line:

   ```
   AnatellaConsole.exe mygraph.anatella  -ta
   ```

   …will create a trace file (also named log file) named:
   "***yyyyMMdd_hh***H***mm***_p***PPPP***_mygraph.trace"
   …that is just "next to" the Anatella graph "mygraph.anatella" (i.e. in the same directory).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In the filename, the letters "***yyyy***" will be replaced by the current year.
"***MM***" will be replaced by the current month.
"***dd***" will be replaced by the current day.
"***hh***" will be replaced by the current hour.
"***mm***" will be replaced by the current minute.
"***PPPP***" will be replaced by the Anatella process id.

For example, if we are in the 21th of March in 2012 at 17:03, the trace filename will be:
"***20120321_17***h***03***_p***1673***_mygraph.trace"

This command-line:

```
AnatellaConsole.exe mygraph.anatella  -tad:\mylog.trace
```

…will create a trace file (also called log file) named:
"d:/***yyyyMMdd_hh***H***mm***_p***PPPP***_mylog.trace"

For example, if we are in the 21th of March in 2012 at 17:03, the trace filename will be:
"d:/***20120321_17***h***03***_p***1673***_ mylog.trace"

Please note that there are no spaces between the "-ta" command-line option and the trace filename.

Alternatively to the above different trace file options, you can also save the content of the log window using the "-te" option (that redirects the content of the log window to the "stdErr"): See the section 4.9. to know more about this subject.

For example, these two command-lines are redirecting the StdErr towards a trace file named "d:\mylog.trace":

```
AnatellaConsole.exe mygraph.anatella  2> d:\mylog.trace
```

```
Anatella.exe -e mygraph.anatella -te 2> d:\mylog.trace
```

4.7.5. Changing the "*Working directory for Cache Files*" parameter

This command-line:

```
AnatellaConsole.exe mygraph.anatella  "-wd:/anatella cache"
```

…will change the "*Working directory for Cache Files*" parameter to the "d:/anatella cache" directory.

This command-line:

```
Anatella.exe –e mygraph.anatella  -w:/
```

…will change the "*Working directory for Cache Files*" parameter to same directory as the .anatella file.

The "-w" command-line parameter has priority over the value given inside the .anatella file.

4.7.6. Clearing the default value of all the "Anatella Global Variables"

This command-line:

```
Anatella.exe -e mygraph.anatella  -c -DA=123  -DB=465
```

…will reset all the "Anatella Global Variables" to the empty value and then set the "Anatella Global Variables" "A" as "123" and "B" as "456".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This option is useful to be sure to correctly re-define on the command-line the value of ***ALL*** the "Anatella Global Variables": i.e. if you forgot to re-define one variable, Anatella will use the "empty value" (and not the default value of the graph): This way you can easily "spot" the variables that you forgot to re-define.

4.8. Scheduling Anatella Graphs

To run different Anatella graphs at regular intervals, you can use any "scheduling tool": The most common "scheduling" or "continuous integration" tools are: **Jenkins**, the Microsoft Windows Task Scheduler, VisualCron, Z-Cron, TeamCity, WinScheduler, etc.

To schedule your Anatella graphs we advise you to use "**Jenkins**". You'll find a detailed list of the many advantages of Jenkins in the next section 4.8.1.

If you don't have administrative rights to install Jenkins on your machine, you can still also use the simpler "Microsoft Windows Task Scheduler": see the section (4.10.) for some explanation on how to use Anatella and the "Microsoft Windows Task Scheduler" together. The main disadvantage of the simple "Microsoft Windows Task Scheduler" is that, by default, it does not keep track of all the execution logs so that it's more difficult to find&fix the problems that might arise during scheduled executions (but you can correct that situation using the "-tf" flag: see the section 4.7.4. for more details).

4.8.1. Why Jenkins ?

The advantages of Jenkins over other simpler "Task Scheduler" are:

- Jenkins is directly integrated within Anatella: i.e. in a few mouse-click you can directly schedule any graph. More details in this youtube video:
  5 clicks to Schedule your Anatella Flows:
  https://www.youtube.com/watch?v=FjIGV2LkcHo

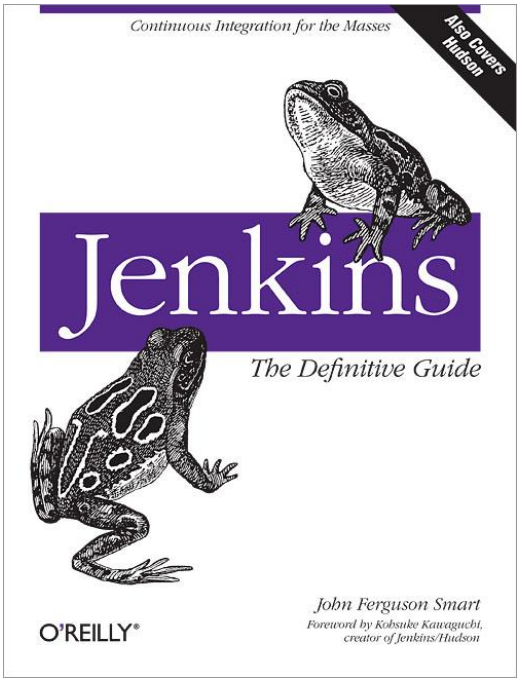
- Jenkins has a web interface. This simplifies a lot the administration of the jobs to run.
  For example: a specific user can consult the web interface to see if its own jobs executed properly and then he can ask to re-run the failed jobs with one mouse click inside the web interface.
- Jenkins can transparently manage a fleet of many computers (i.e. it manages many "nodes" in technical terms). When Jenkins needs to run a job, Jenkins can easily connect to an "idle" node and run the required job there (in technical term, this is called "distributed computation"). This gives to the final user/company a tremendous computing power: There are actually no limits to the delivered computing power: if you need more computing, simply add some more "nodes".
- The integration between Anatella and Jenkins has now been thoroughly tested for all the common tasks, such as jobs scheduling (see section 4.8.4.).
  Thanks to Jenkins, Anatella can also perform much more complex tasks such as running "on-demand" a large quantity of graphs (i.e. "on-demand, near real-time" distributed computation). For more information about this suject, see the sections 5.21.1., 5.21.2. and 5.21.3. (about the loopJenkins action, the waitJenkins action and the queryJenkins action).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Jenkins automatically saves the execution traces (i.e. the execution logs) of all the jobs that it executes. This allows backtracking at a specific date to see if everything executed as planned (typically, for debugging purposes).
- With Jenkins, you can define different users: Each user has its own set of rights.
  Specific users can see & execute specific jobs.
  This allows for everybody in your company to closely & easily track the execution of their own jobs without jeopardizing the security of other, more sensitive, jobs.
- Jenkins delivers some statistics about the number of successful & failed jobs over the last hours or days. These statistics &kpi's are displayed inside a nice graphical interface that allow easy tracking of the "sanity" of the scheduled jobs.
- Jenkins manages in a more intelligent way your ressources. For example, it can happen that several "heavy-workload-jobs" are scheduled to run at the same hour. When this situation happens, the "Microsoft Windows Task Scheduler" will simply runs all these "heavy-workload-jobs" simultaneously. This can lead to a complete crash of the scheduled jobs (e.g. when there aren't enough RAM memory avaible to properly execute the jobs). This won't happen with Jenkins: Jenkins automatically limits the number of jobs that are simultaneously executed.
- As all scheduler, Jenkins can run jobs at regular intervals (e.g. every week on Monday & Wednesday at 01:00 am) but Jenkins can also trigger the execution of jobs for many other reasons: For example: Jenkins can run a job when a specific file arrives, when a connection to a specific web-service occurs, when an email is received,etc. In the Jenkins vocabulary, these are named "build triggers". There are currently 59 different "build triggers" inside Jenkins plugins. A list of all "build triggers" is visible here:
  https://plugins.jenkins.io/ui/search?sort=relevance&labels=trigger&view=Tiles

- There are currently more than 1800 plugins inside Jenkins. Amongst these "plugins", you'll find different "build triggers" and many other things. A list of all plugins is here:
  https://plugins.jenkins.io/

- Jenkins is a well-known scheduler (also named "continuous integration software"): If you want more information about Jenkins, you can consult the book named "Jenkins: The Definitive Guide" published by O'Reilly:



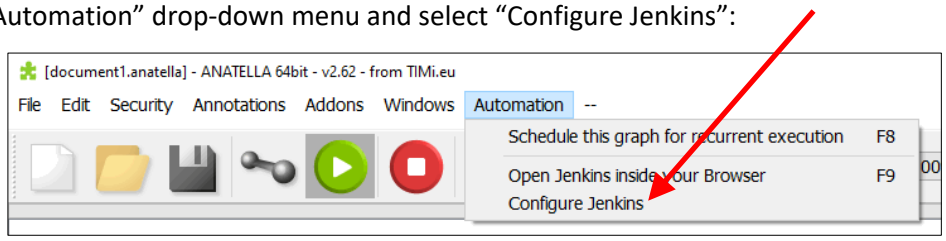
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.8.2. Jenkins installation procedure

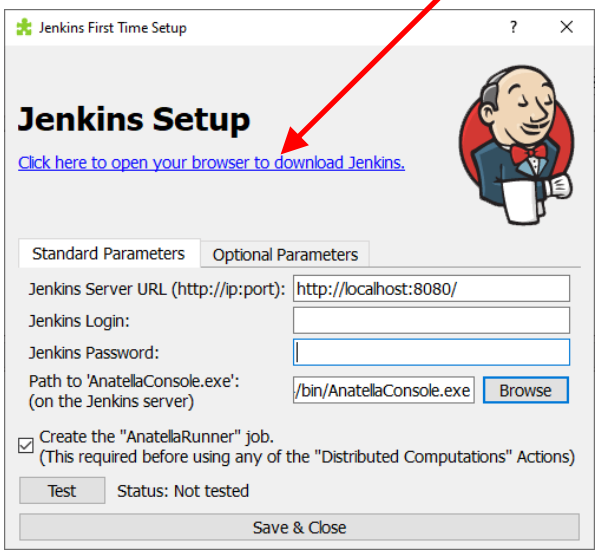
One pre-requisite to be able to run an Anatella graph inside Jenkins is to register your Anatella/TIMi licences as a "System-Wide" License: More precisely: You must follow the procedure given in section "7.4.4: Enter a Server-Wide Licence".

Here, we'll give instructions for the installation of Jenkins under Windows. The installation process under Linux is similar. A simplified Jenkins installation procedure is also documented using a youtube tutorial video available here: https://www.youtube.com/watch?v=MtuUpn2GskI
Follow these steps:

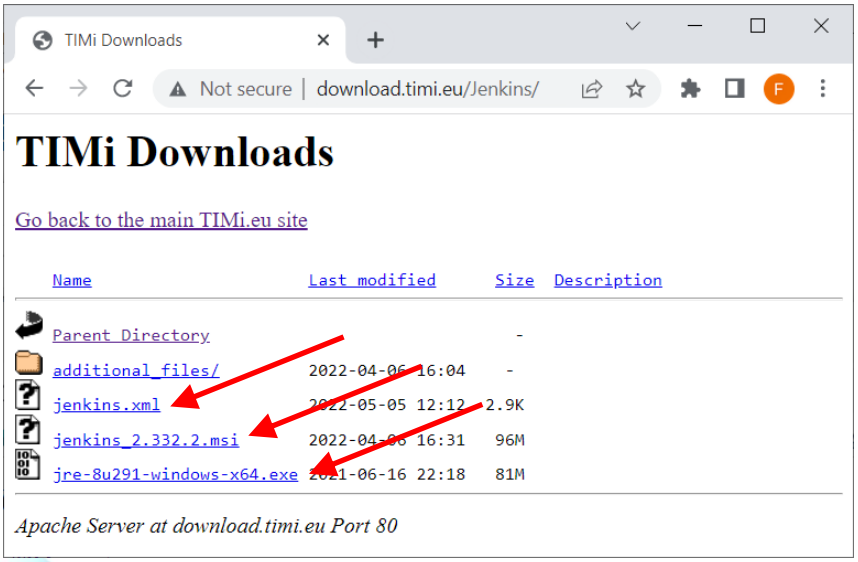
1. Inside Anatella, open the "Automation" drop-down menu and select "Configure Jenkins":



2. Inside the Jenkins configuration screen, click on the blue URL link:



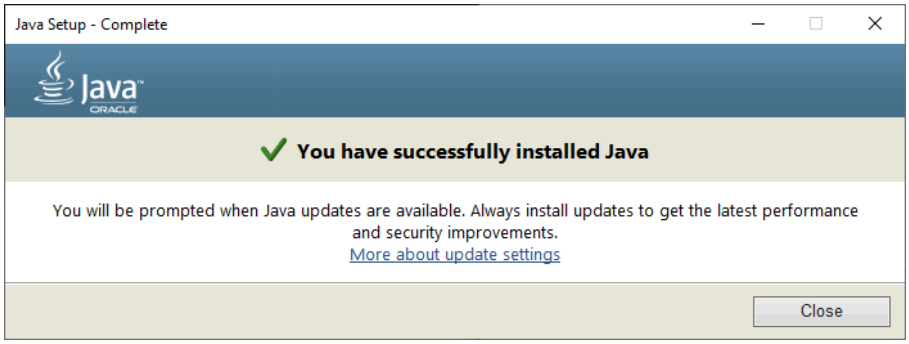
3. Your browser should now display the URL: http://download.timi.eu/Jenkins/
Download the 3 files: "jenkins.xml", "jenkins_xxx.msi" and "jre_xxx.exe":



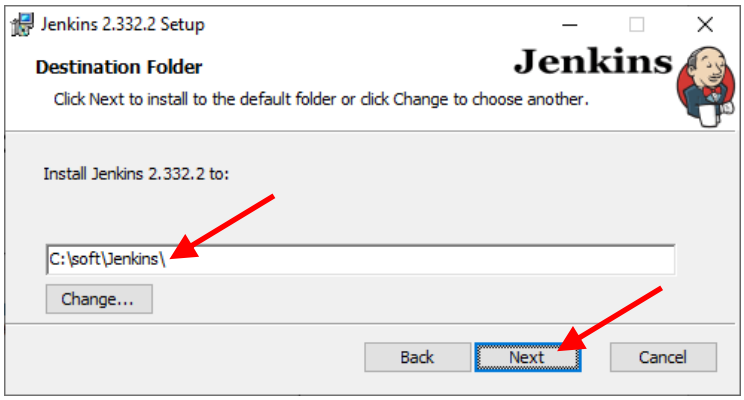
These files are placed there for your convenience.
You can also download them from their original location: These are:

- For the Jenkins installer ("Jenkins_xxx.msi"): https://www.jenkins.io/download/
- For the Java Virtual Machine ("jre_xxx.exe"):
  …either from here: https://www.oracle.com/java/technologies/downloads/
  …or from here: https://adoptium.net/temurin/releases/

4. Install the "Java Virtual Machine": Run the file "jre_xxx.exe". Click the "Install" button and wait until completion of the install process. For exemple, you should see:

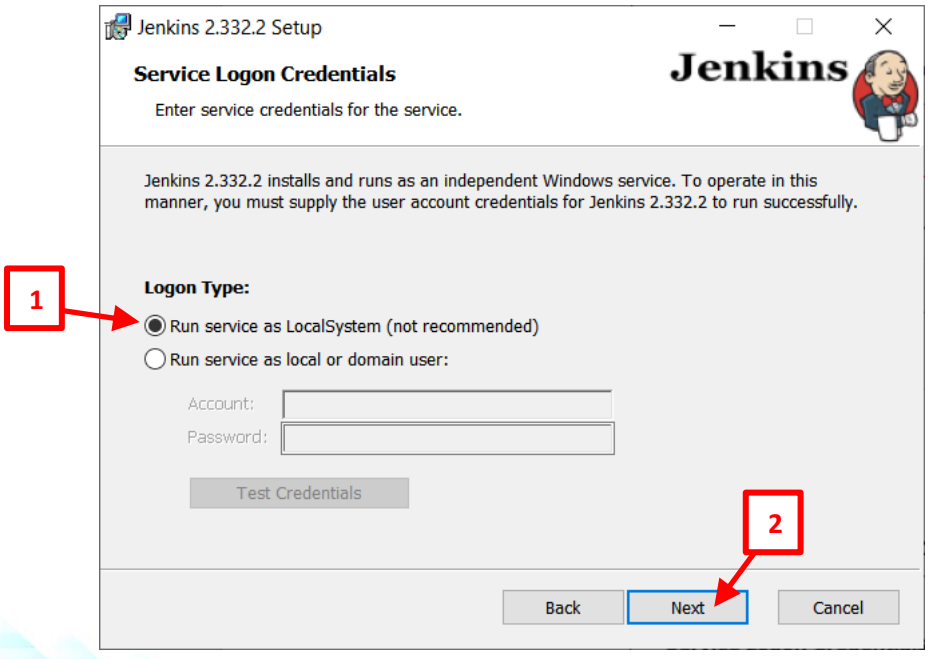


5. Run the file "jenkins_xxx.msi", click the "Next" button and select the Jenkins installation directory. I usually choose: "C:\soft\Jenkins\" like that:



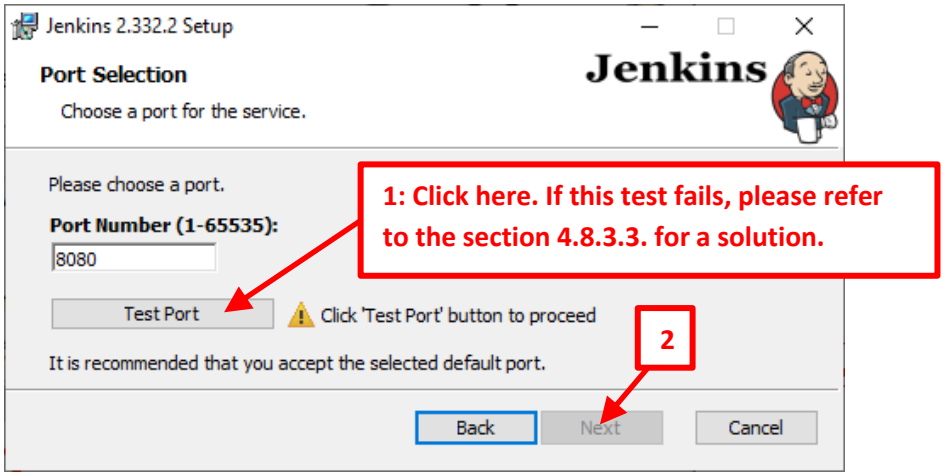
..and click "Next".

6. On the next configuration screen, select the radio button "*Run the service as LocalSystem (not recommended)*" and click the "*Next*" button:

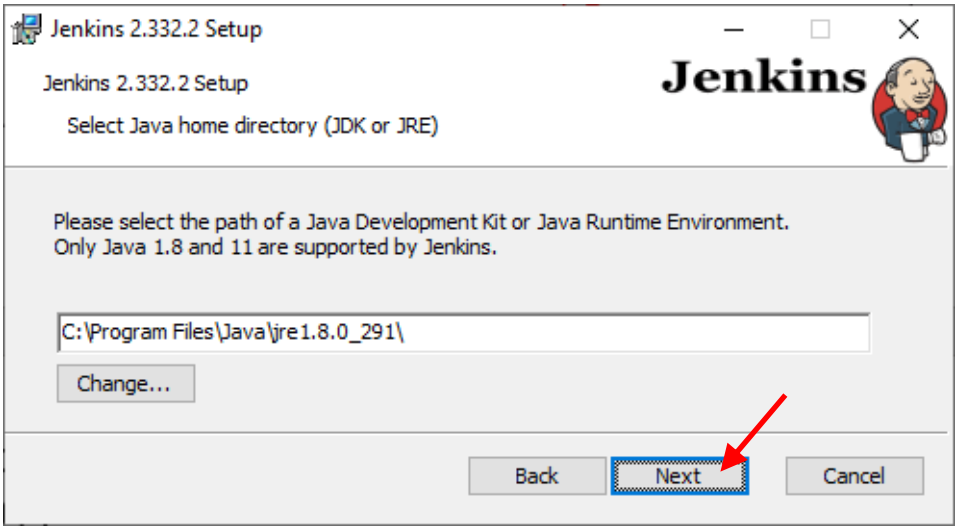


More Details: We really need to execute the Jenkins service "*as a local or domain user*" otherwise we'll run into many troubles (see section 4.8.3.2. for a list of all the bad things that can happen when running the Jenkins service as the "*LocalSystem*" account). …But, unfortunately, the Jenkins installer has a bug that prevents us to select the better option (i.e. we cannot select "*Run service as local or domain user*"), so we select instead here the worse option (i.e. we selected "*Run service as LocalSystem (not recommended)*") and we'll perform a few extra operations (during step 11 of this installation procedure) to ensure that the Jenkins Service is still running "*as local or domain user*".

7. Click on the "Test Port" button, click on the "Next" button:



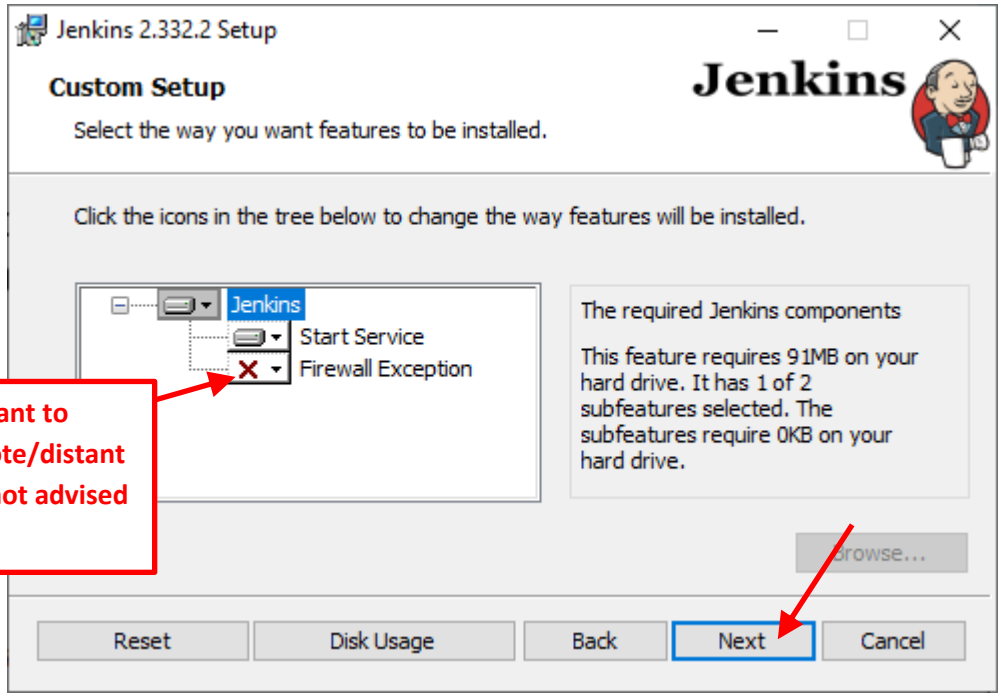
Select the "Java Virtual Machine" that will be used to run Jenkins. By default, it's:
C:\Program Files\Java\jre1.8.0_291\
…And click the "Next" Button:



You might get a warning message about installing "Java 11" instead of "Java 8". Disregard this warning.

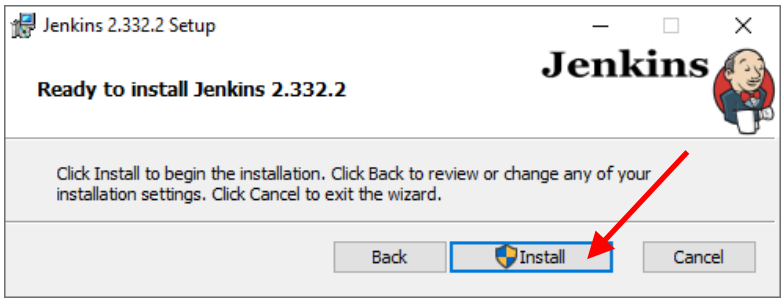
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

8. Click the “Next” button:

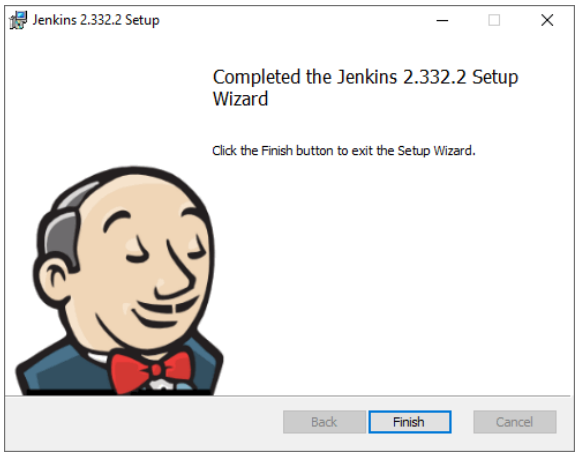


Enable this option if you want to access Jenkins from a remote/distant machine (Warning: this is not advised since it’s a security risk).

9. Click the “Install” button:



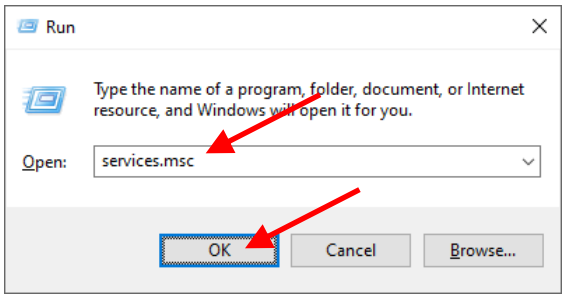
10. After a few minutes, you should see:



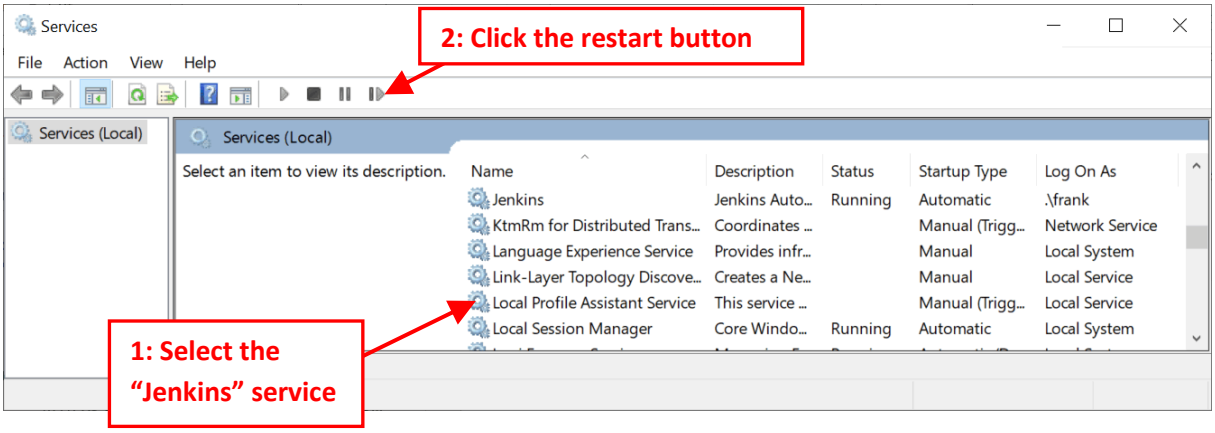
If you get a message about “impossible to start service”, you need to follow these two steps to open the “service manager” and manually start the “Jenkins” service:

- Open the “Service” Manager: Press simultaneously the keys [WIN]+[R] on your keyboard to open the “Run” window. Then write “services.msc” and click the “OK” button:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



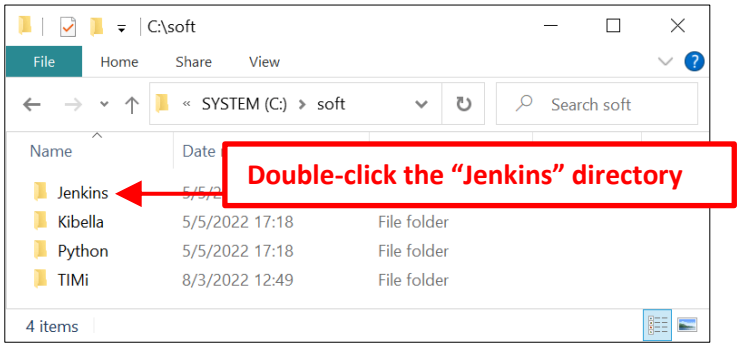
- Inside the "Service" Manager window:



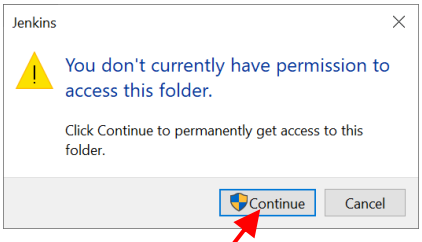
If Jenkins still does not start, please refer to the section 4.8.3.3. to know how to diagnostic & fix the error.

11. We really need to execute the Jenkins service "*as a local or domain user*" otherwise we'll run into many troubles. Unfortunately, up to this point, the Jenkins service is running "*as the LocalSystem account*". Let's correct that!

11.1. Navigate to the Jenkins installation directory (usually "c:\soft\jenkins"). By default, you don't have the rights to see the content of the "jenkins" directory. To get these rights, the easiest way is to double-click the "Jenkins" directory inside the MS-File-Explorer windows:



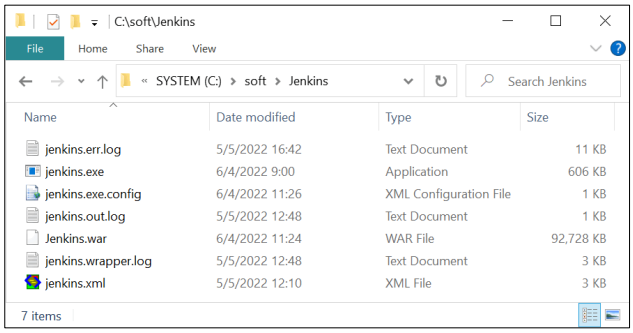
Then you see:



Just click on the "Continue" button:

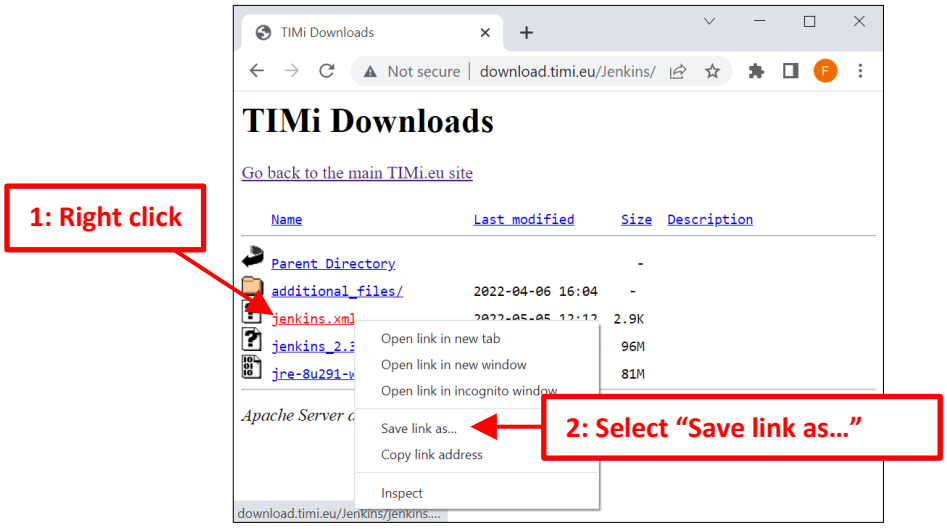
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You should now see:



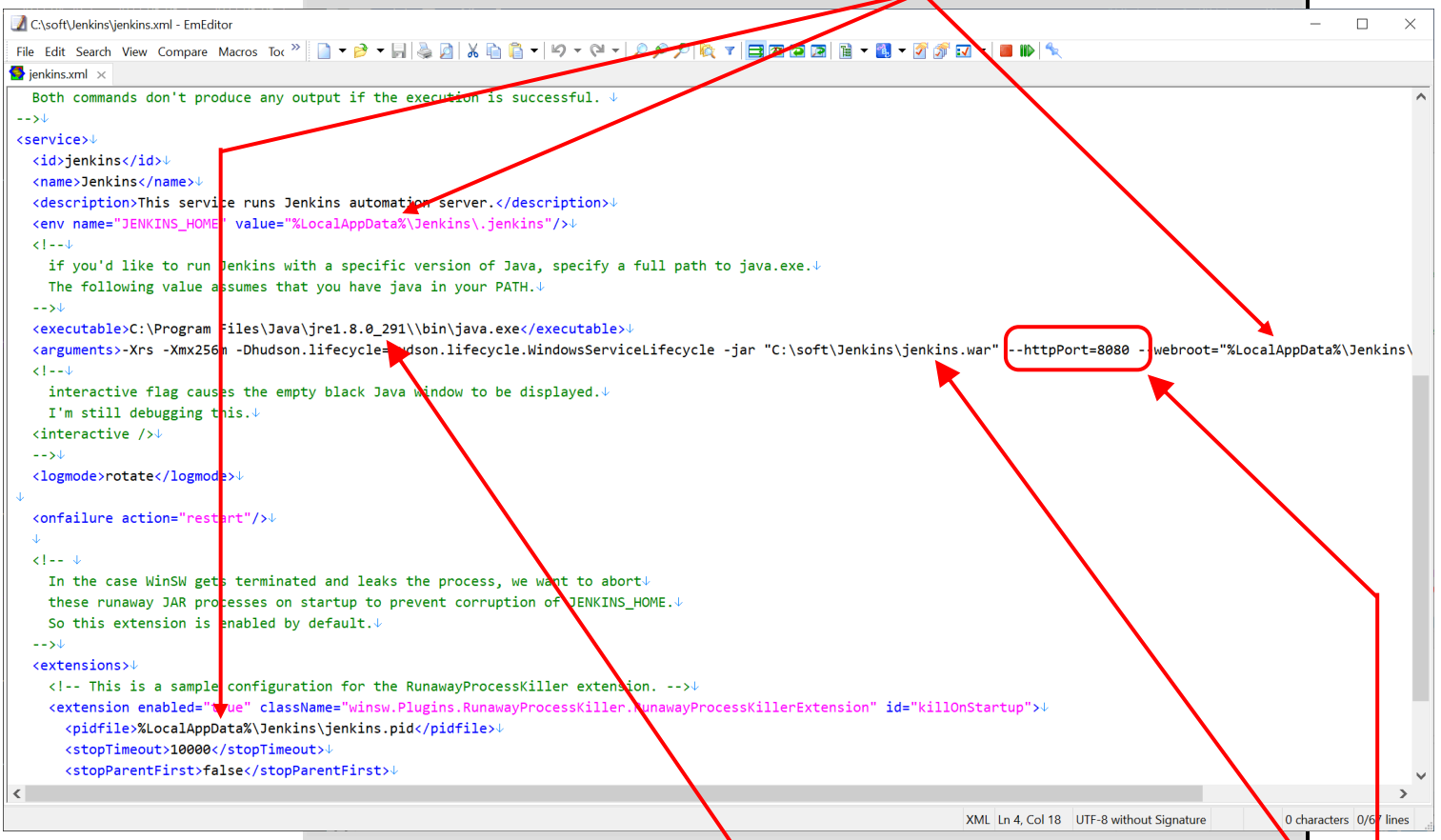
This step is important because your user MUST have the write access rights to the Jenkins directory (otherwise the Jenkins service won't start because it needs to be able to write LOG files inside this directory). If you installed Jenkins inside "c:\program files", you need to change the user access rights of the "c:\program files\jenkins" directory to allow your user to write inside this directory.

11.2. Replace the file "Jenkins.xml" inside the Jenkins installation directory with the one downloaded from: http://download.timi.eu/Jenkins/ . You should already have downloaded this file (during the step 3 of this installation procedure). If you don't have it yet, right-click the "jenkins.xml" link and select "Save link as…":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The difference between this new "Jenkins.xml" and the original "Jenkins.xml" is: We just replaced all occurrences of "%ProgramData%" with "%LocalAppData%" everywhere in the file (i.e. these three lines changed).

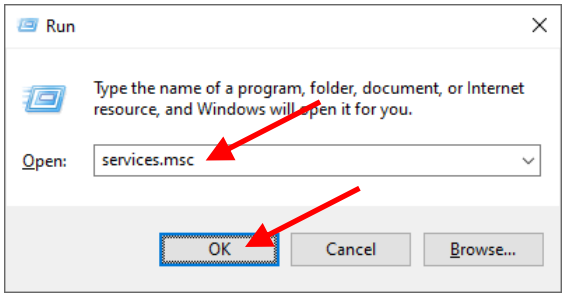


11.3. Jenkins is a tool coded in Java. Thus, to run Jenkins, we need a "Java Virtual Machine" (that we installed during the step 4 of this procedure). Please verify that this FilePath matches the location of a locally installed "Java Virtual Machine". If that's not the case (the FilePath is incorrect), you'll get this error message: "*The Jenkins service on Local Computer started and then stopped*" when starting the Jenkins service.

11.4. Please verify that this FilePath matches the location of your Jenkins installation directory:

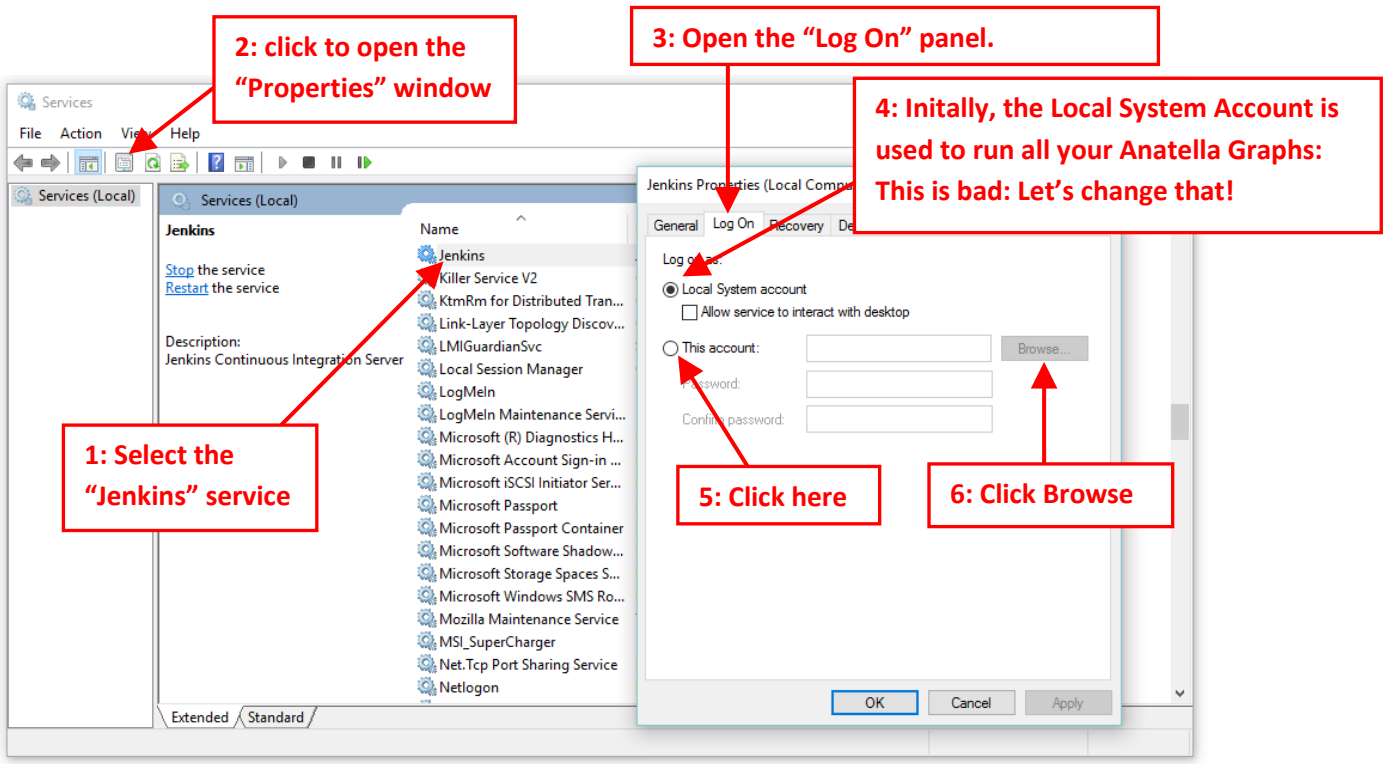
11.5. **Optional:** By default, Jenkins runs on port 8080. If you changed the value of the default port (during the step 8 of this procedure), you need to edit the "Jenkins.xml" (with "notepad") and write the new port value here:

11.6. Open the "Service" Manager: Press simultaneously the keys [WIN]+[R] on your keyboard to open the "Run" window. Then write "services.msc" and click the "OK" button:

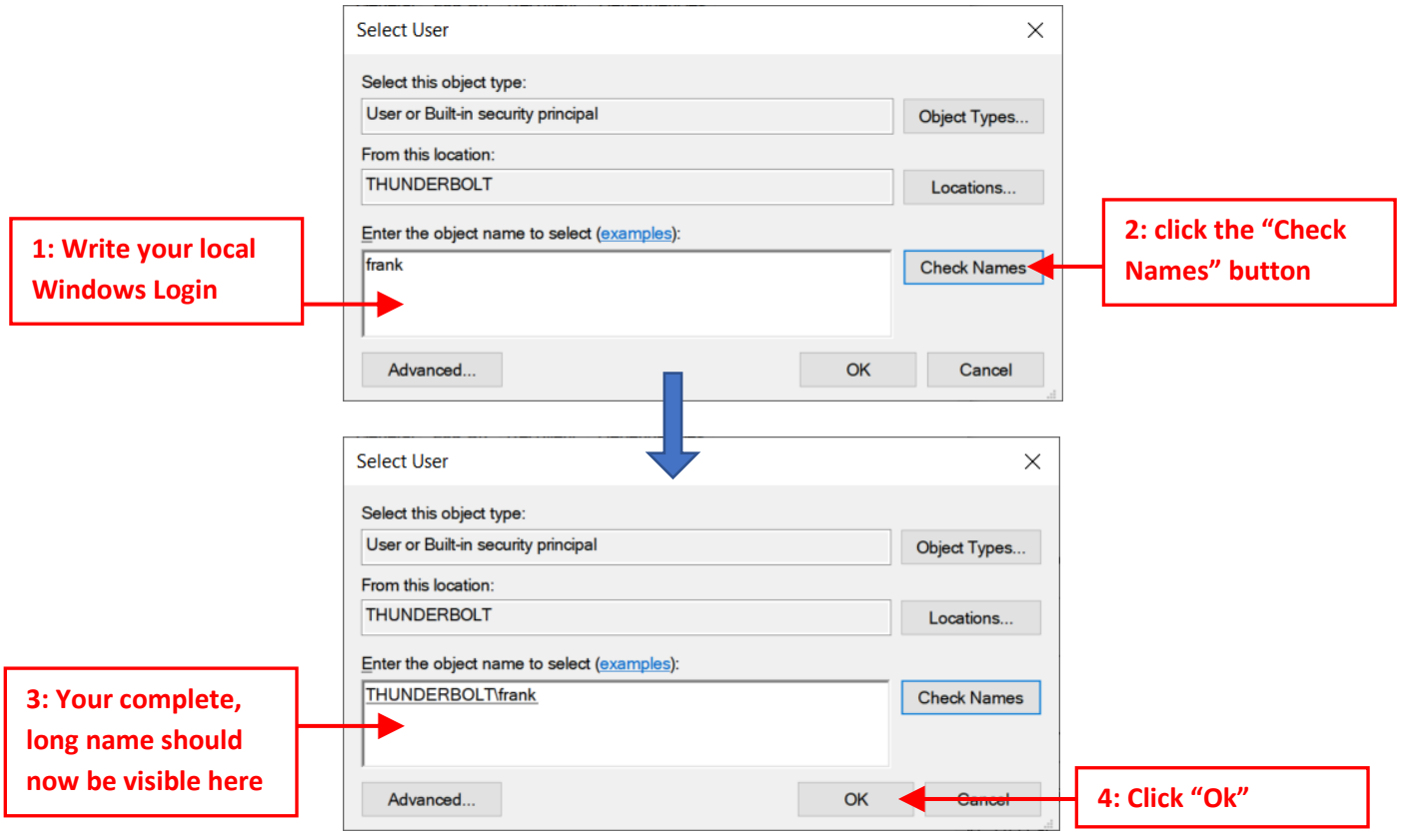


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

11.7. Inside the "Service" Manager window: We setup the Jenkins service to run under your local user:

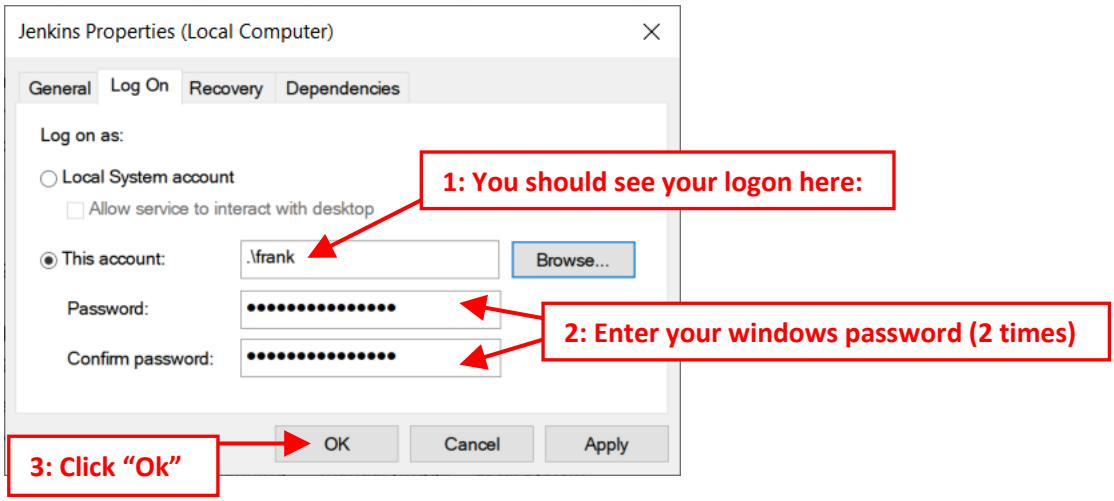


11.8. Let's select your Local Windows account:

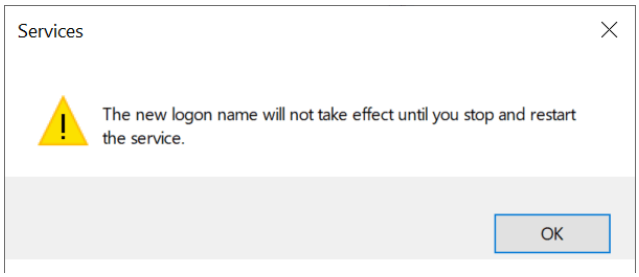


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

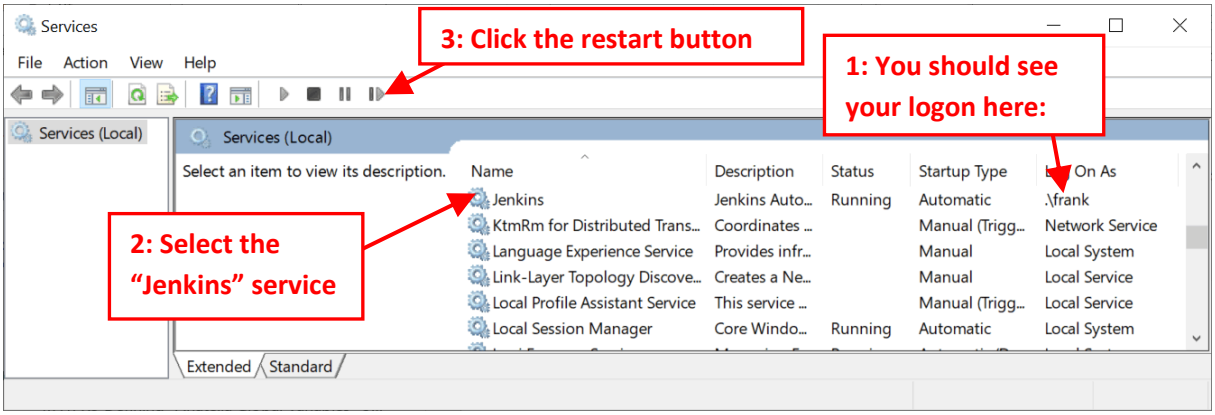
11.9. Enter your windows password (2 times) and press the “Ok” button:



11.10. You should see:

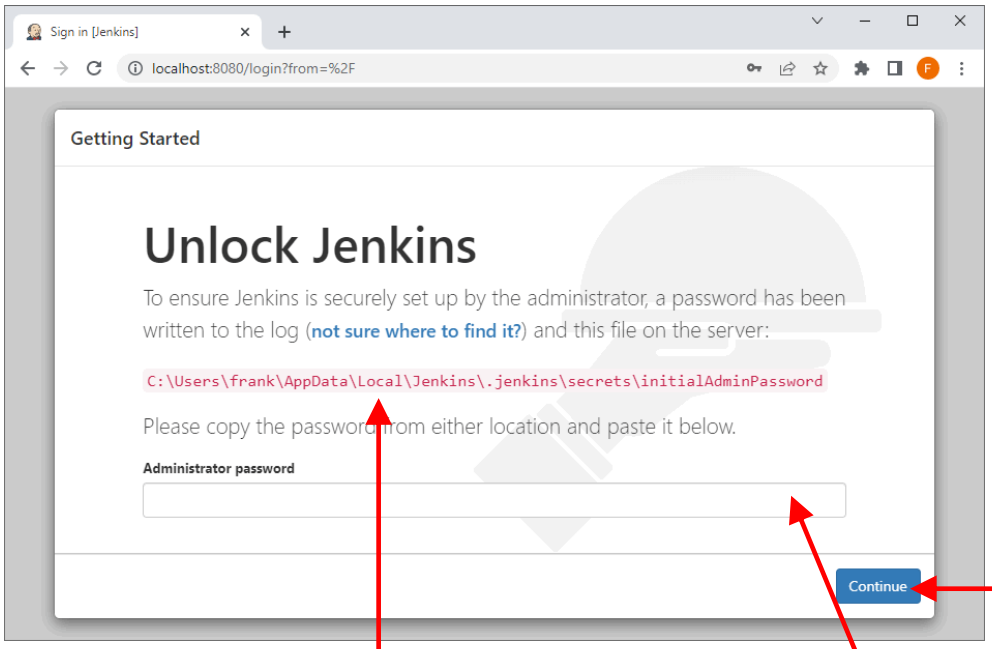


11.11. Restart the Jenkins service:



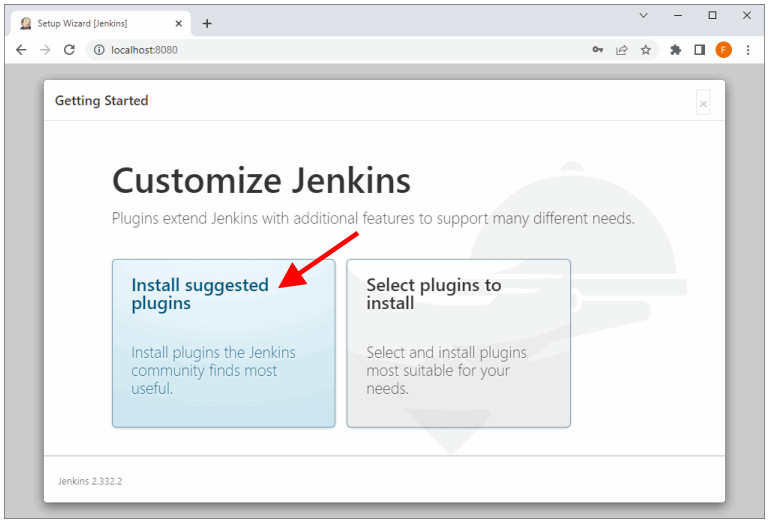
12. Open a browser and navigate to http://localhost:8080 (this is the default jenkins URL). You should see:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



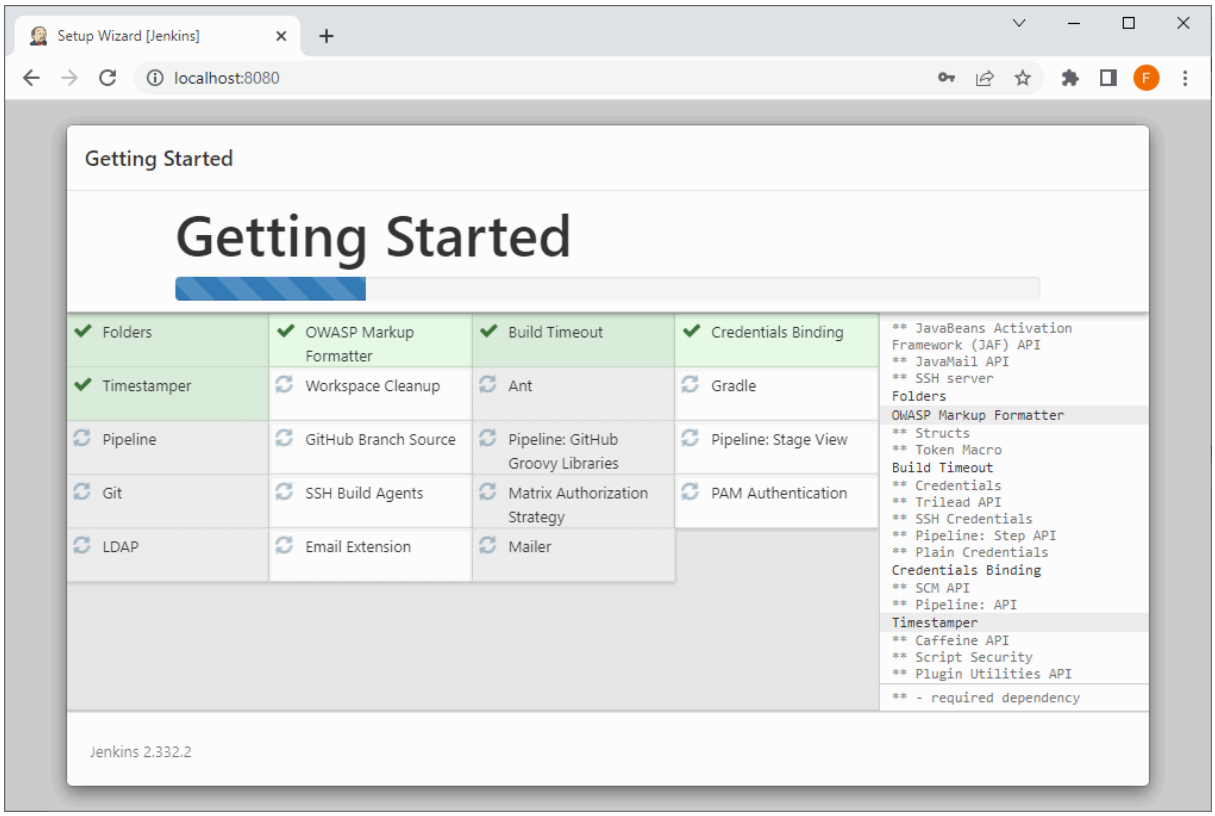
Open with “notepad” the file located here: ..and copy the content of this file here:
Click the “Continue” button.

13. Click the “Install suggested plugins” button (you’ll still be able to install more plugins later):

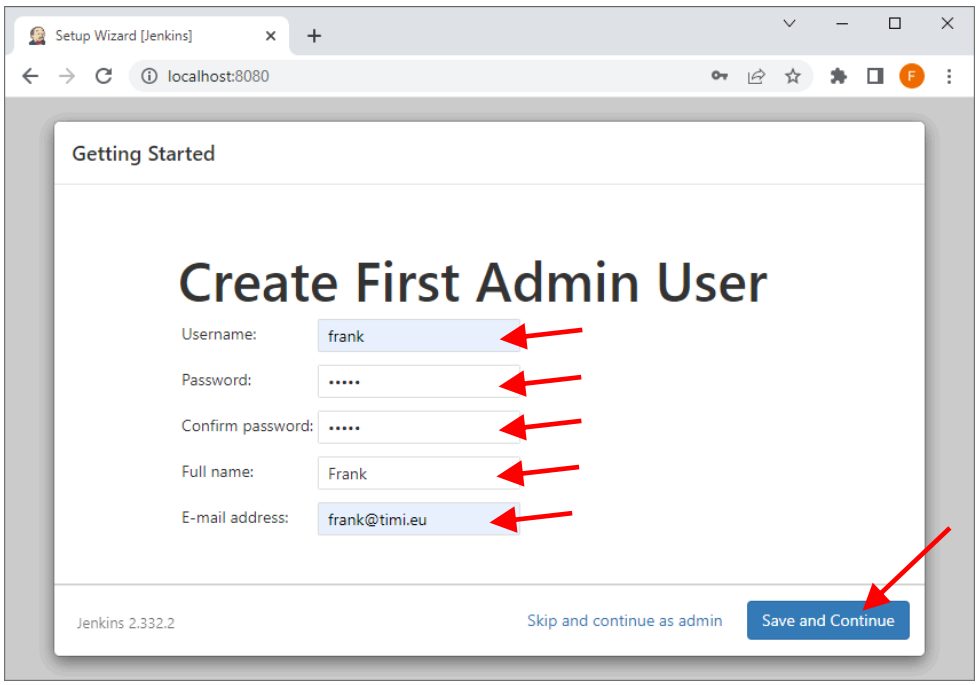


14. Please wait until the default Jenkins plugins are downloaded&installed. This might take some time, depending on your internet connection speed. Do not worry if some plugins do not install directly because:
    - you don’t need any of these optional plugins to use Jenkins with Anatella.
    - you can always install any of them later.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

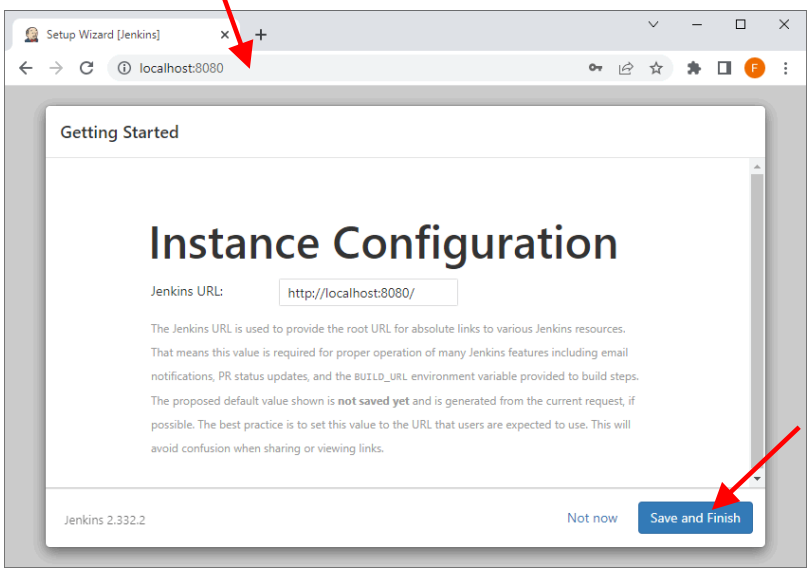


15. Create you first administrative user. You need to remember the login/password that you used here, on this step, because you'll need to enter it again later inside Anatella during step 18. Click on the "Save and Continue" button:

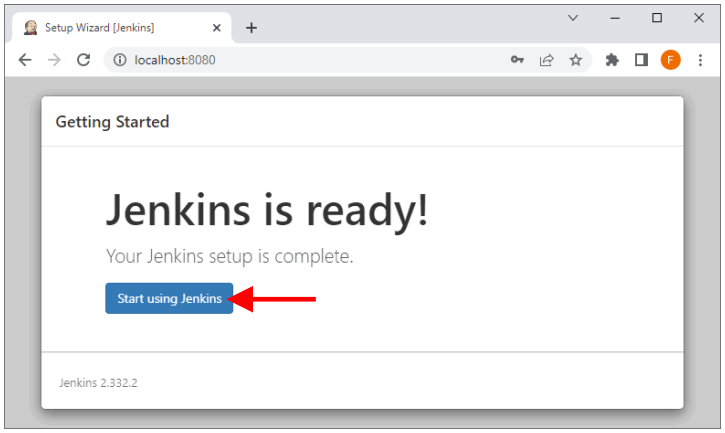


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

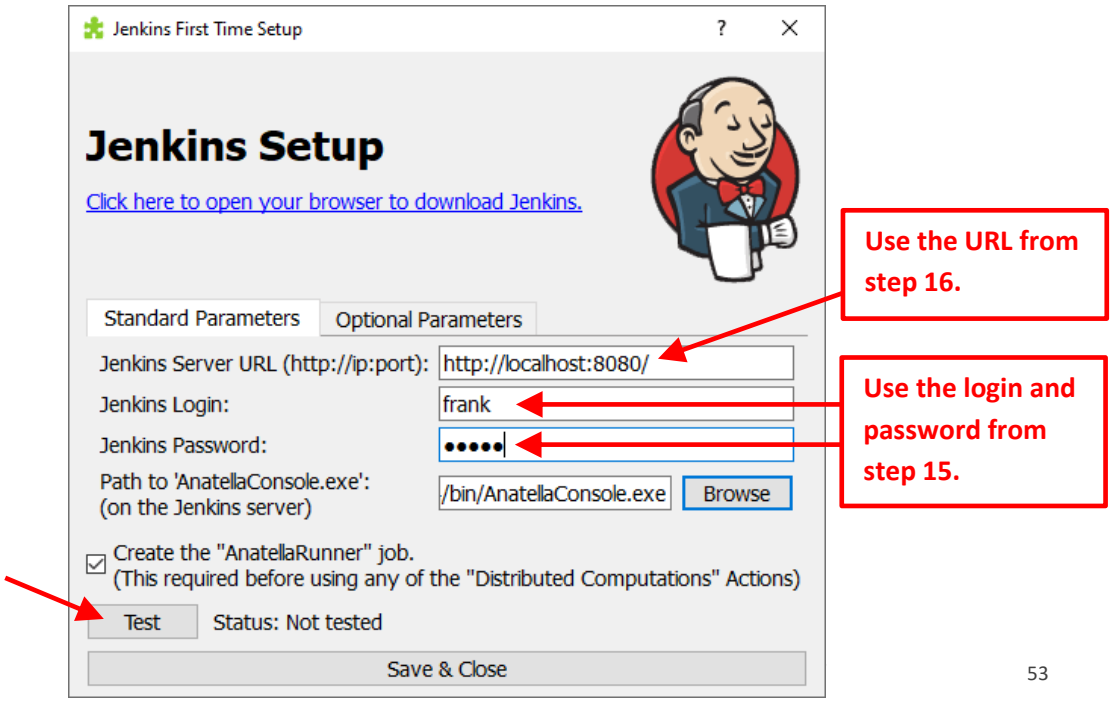
16. You should now see: Click on the “Save and Finish” button:



17. Click on the “Start using Jenkins” button:

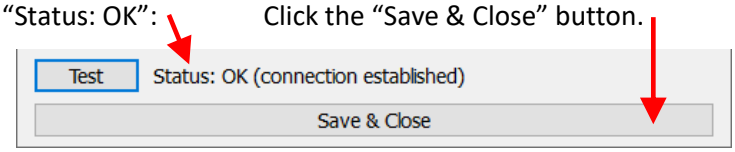


18. Inside Anatella, go back to the Jenkins configuration screen (i.e. open the “Automation” drop-down menu and select “Configure Jenkins” option). Enter in the Anatella window your Jenkins login and your Jenkins password that you used during step 15 and click the “Test” button:

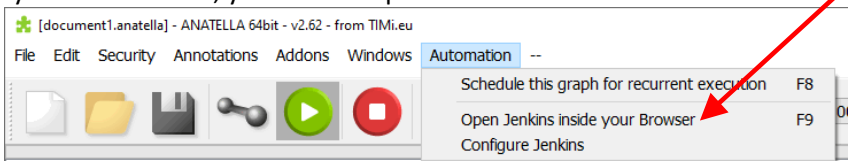


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

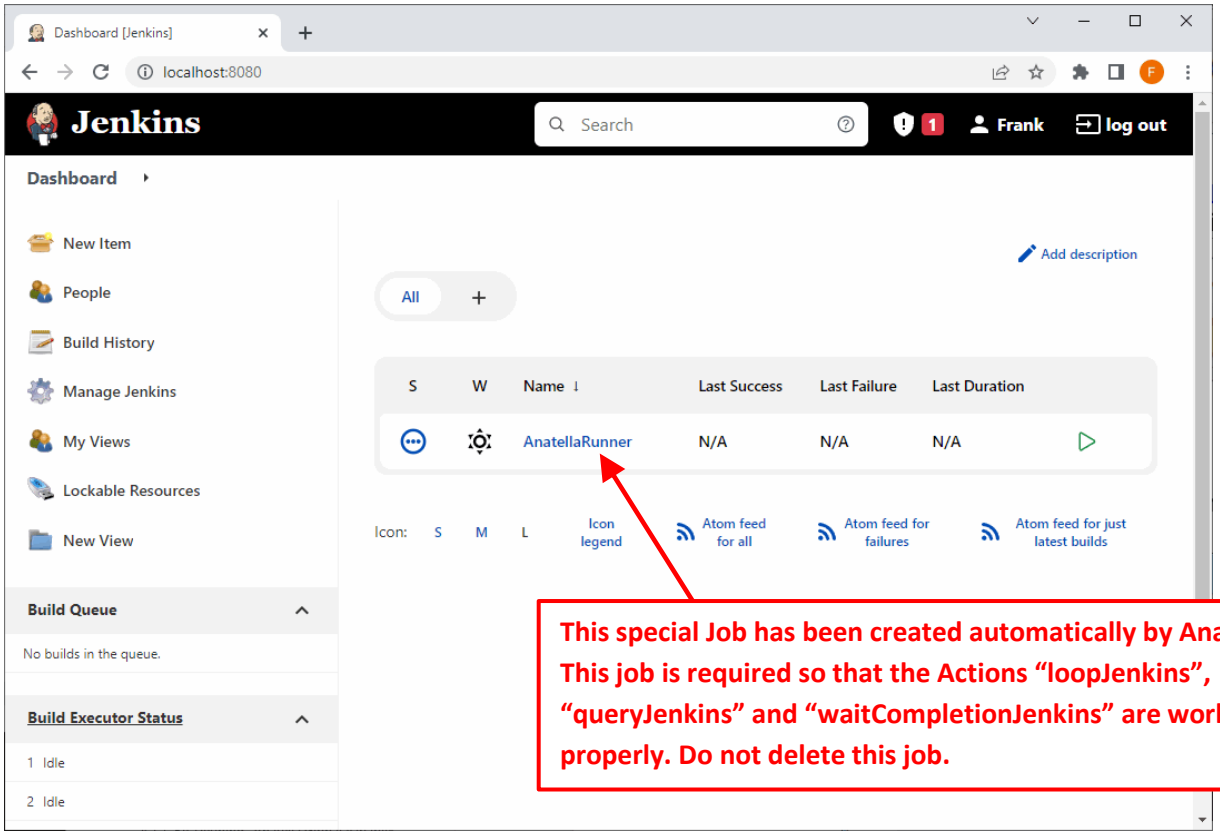
19. You should see “Status: OK”: Click the “Save & Close” button.



20. Open a browser and navigate to http://localhost:8080 (this is the default jenkins URL).
Alternatively, to navigate to your Jenkins URL, you can now press F9 inside Anatella or click here:



You should see:



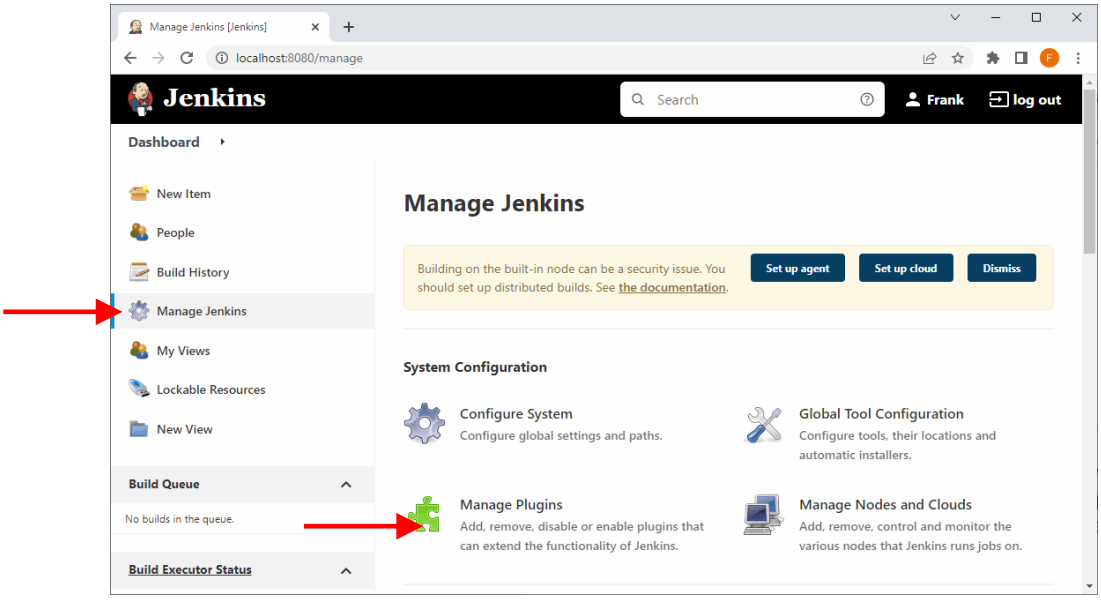
21. Jenkins is now properly configured to work with Anatella.
Congratulations!

22. **Optional step 1/2:** At this point, you might want to add a few optional Jenkins plugins that might be of interest to you. To do so:

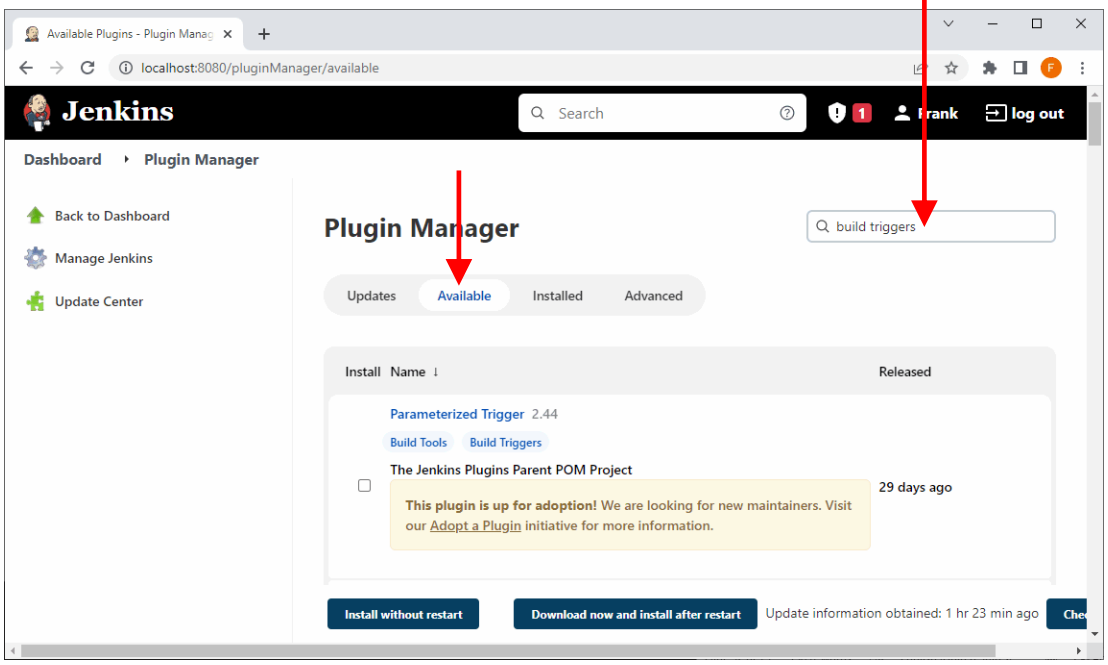
22.1. Open your Jenkins URL in your browser, usually it’s http://localhost:8080

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

22.2. Click on “Manage Jenkins”, then click on “Manage plugins”:



22.3. Click on “Available” and enter (for example) “build triggers” in the search field here:



22.4. Here are the names of a few plugins that might interest you:

Plugin name	Plugin function
Generic Webhook Trigger	Receive any HTTP request, extract any values from JSON or XML and trigger a job with those values available as variables. You can then pass these variables as “Global Parameters” to AnatellaConsole.
Naginator	Reschedules failed jobs.
Filesystem Trigger	makes it possible to monitor changes of a file or a set of files in a folder.
URLTrigger	makes it possible to poll changes of URLs.
Poll Mailbox Trigger	to poll an email inbox, and trigger jobs based on new emails.
Files Found Trigger	Schedules a job when certain files exist.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

23. Optional Step 2/2: "Custom Jenkins Initialization Scripts"

It happens very often that an Anatella graph must read some data files stored on a remote "central" File Server (typically these data files are stored on a "network" drive that is named "X:" or "Z:"). This situation must be handled in a special way when executing such an Anatella-Graph inside Jenkins.

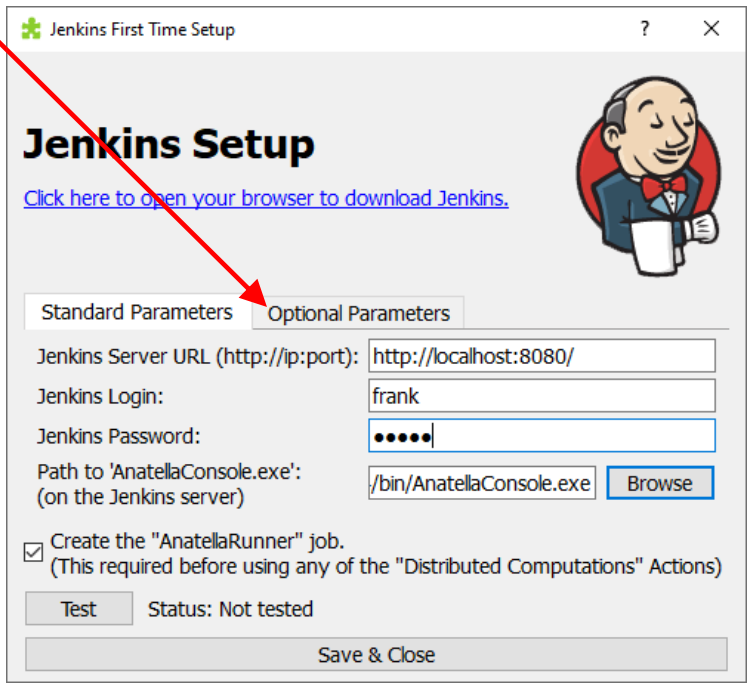
How are created (i.e. "mounted" in technical terms) the network drives? Usually, there exists a customized "ad-hoc" script (written by your IT department) that are "mounting" all the required network drives when you log-in on your computer. This initialization script might typically perform many different tasks, such as:

- It mounts various network drives on your computer (using the "`net use`" command).
- It initializes different Environmental Variables
- It configures different parameters inside various ODBC/OleDB drivers (such as the character encoding of the ODBC drivers).
- etc.

The Anatella graphs that are executed by Jenkins are each running inside their own "clean" windows session. When Jenkins creates a new "clean" session (to run a new graph), it does not execute any of the "custom initialization scripts" written by your IT department. As a consequence, this means that the network drives (i.e. the drives named "X:" or "Z:") are not (directly) available. Since some important network drives are missing, it's very likely that Anatella won't be able to execute successfully the required data-transformation-graphs.

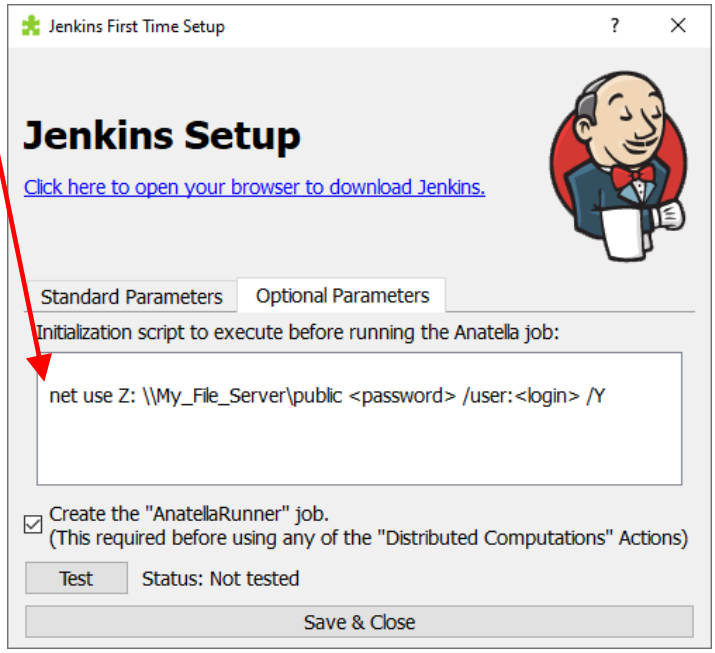
Hopefully, there is a solution to all these difficulties: i.e. We can define our own "custom initialization scripts" that are executed by Jenkins before running any Anatella graph inside Jenkins. In the simplest situation, you know exactly the content of the "custom initialization scripts" written by your IT department and you can simply copy/paste this script inside the "Jenkins Initialization Script Window" that is accessible here:

23.1. Inside Anatella, go back to the Jenkins configuration screen (i.e. open the "Automation" drop-down menu and select "Configure Jenkins" option) and click on the second panel that is named "Optional Parameters":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

23.2. Enter all the “custom initialization scripts” (e.g. all the commands required to “mount” the network drives) here:



In many situations, the “custom initialization scripts” written by your IT department will be overly complex and it might just be a better idea to use a much simpler “Custom Jenkins Initialization Scripts” that is composed of just a few lines with some “net use” commands such as these ones:

```
net use Z: \\My_File_Server\public /Y
```

…or:

```
net use Z: \\My_File_Server\public <password> /user:<login> /Y
```

For example, the above command “mounts” the drive “Z:” using your login so that this drive is accessible for Anatella inside Jenkins.

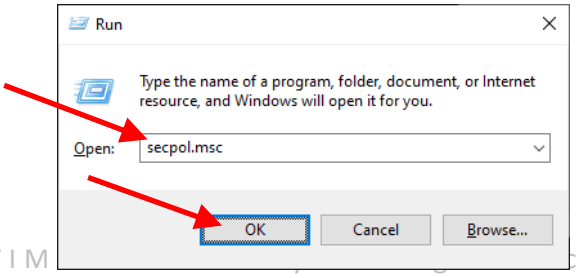
4.8.3. Solving Jenkins Installation Errors

This section contains some solutions to the most common Jenkins installation errors. If you already installed Jenkins successfully, you can safely skip this section.

4.8.3.1. Allow your “Windows User/Account” to execute as a service

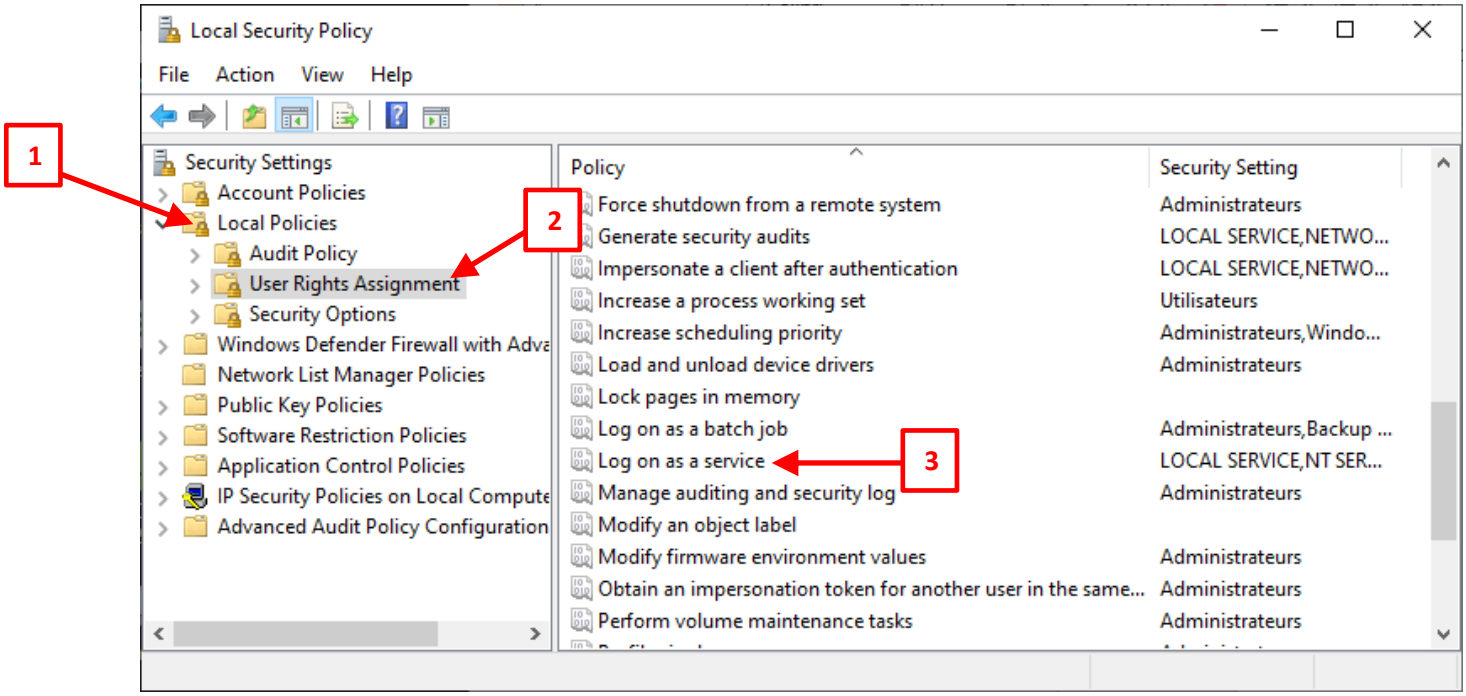
Let’s give to your user account the rights to execute as a service! If you are not authorized to get these rights, you can still run Jenkins under the “*LocalSystem*” account: See the section 4.8.3.2. for more details on this subject.

1. Press simultaneously the keys [WIN]+[R] on your keyboard to open the “Run” window. Then write “secpol.msc” and click the “OK” button:

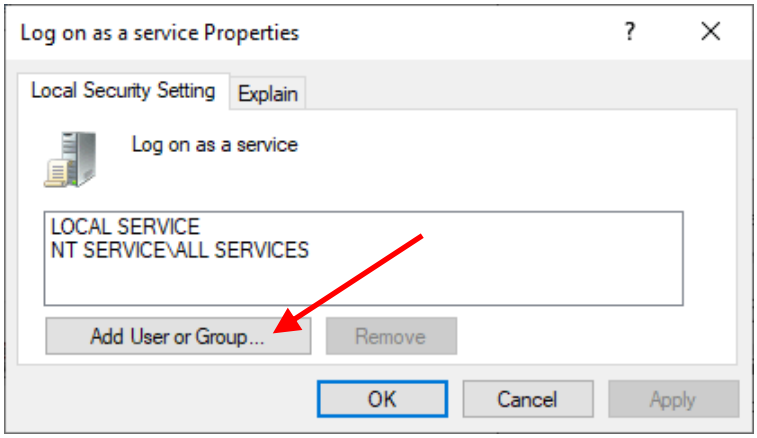


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

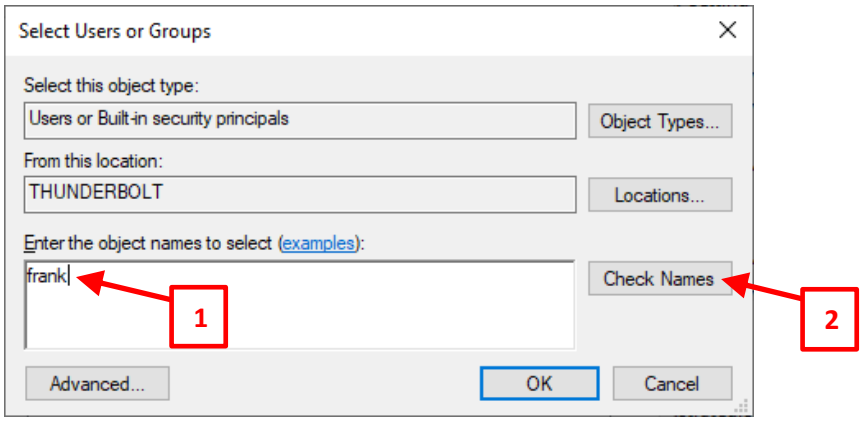
2. Inside the "Local Security Policy" window, inside the tree structure on the left, select "*Local Policies*" and, just below: "*User Rights Assignment*". Then, in the same window, in the right pane, double-click the line named "*Log on as a service*":



3. Click the "Add User or Group…" button:

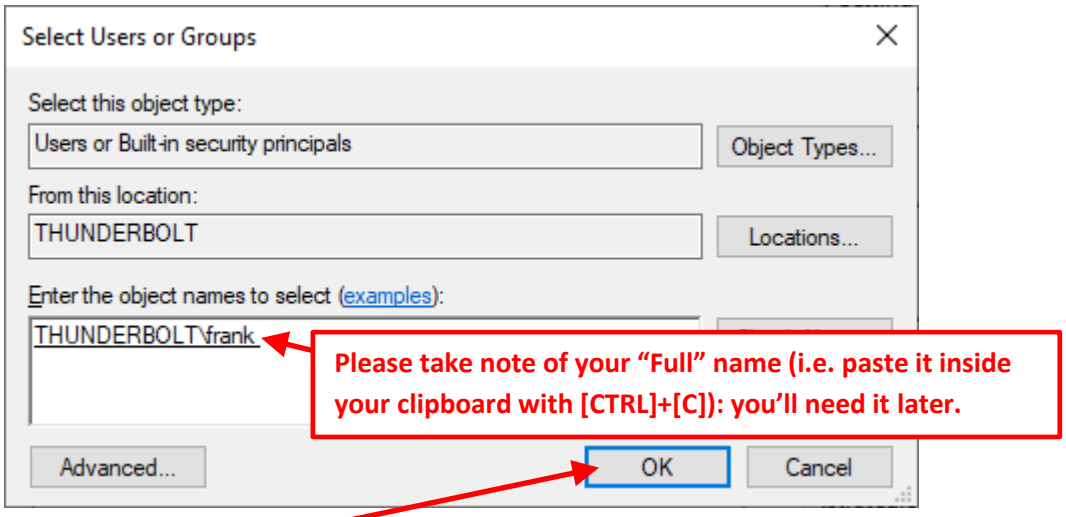


4. Enter your MS-Window Login and click the "Check Names" button:

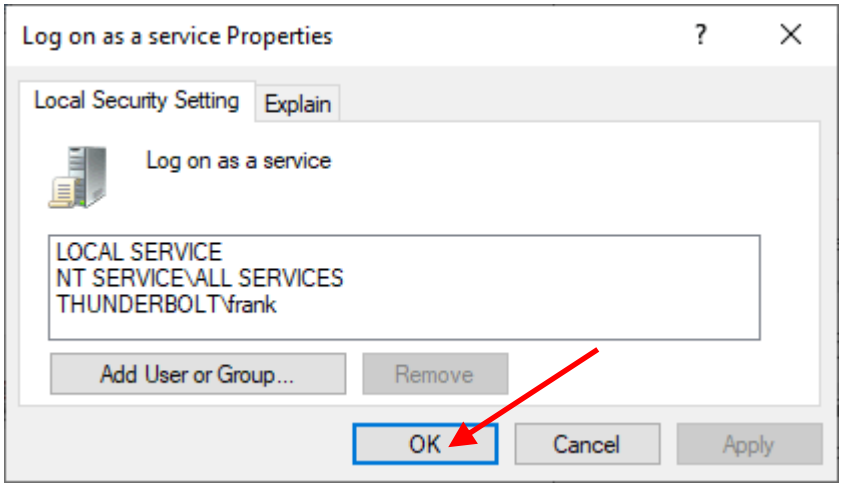


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Your short name is now verified and replaced by your "Full" name, including the "Windows Domain" prefix:



5. Click the "OK" button:



6. Close the "Local Security Policy" window.

4.8.3.2. Running Jenkins as the LocalSystem Account

Running Jenkins under your own account usually solves many different problems. For example, if you did not install Jenkins under your own account, you'll most certainly experience the following difficulties/errors:

1. The "shared drives" (i.e. the network drives: Typically "Z:") are not accessible to the "Local System Account": You must execute Jenkins under you own account to have access to the "shared drives".
2. If your Anatella Graphs are using a Type-1 ODBC connection, make sure that this ODBC connection (i.e. the "ODBC DSN" or ODBC link) is inside the category "System DSN" (and not "User DSN"), otherwise the Graphs executed by Jenkins won't have access to your (ODBC) connections: You'll find more information about the difference between "System DSN" and

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

"User DSN" inside the section 5.1.6.1 about Type-1 ODBC connections (you won't have any problem if you are using Type-2 ODBC connections).

3. Many functionnalities inside the R or Python Engine are not working when executed under the "Local System Account" (although they are working ok when executed under your own personal account).
4. You must enter your Anatella Serial Number as a "System-Wide License" (otherwise the Jenkins user won't have the required Serial Number to run Anatella). More precisely: You must follow the procedure given in section 7.4.4 to enter your Licence.

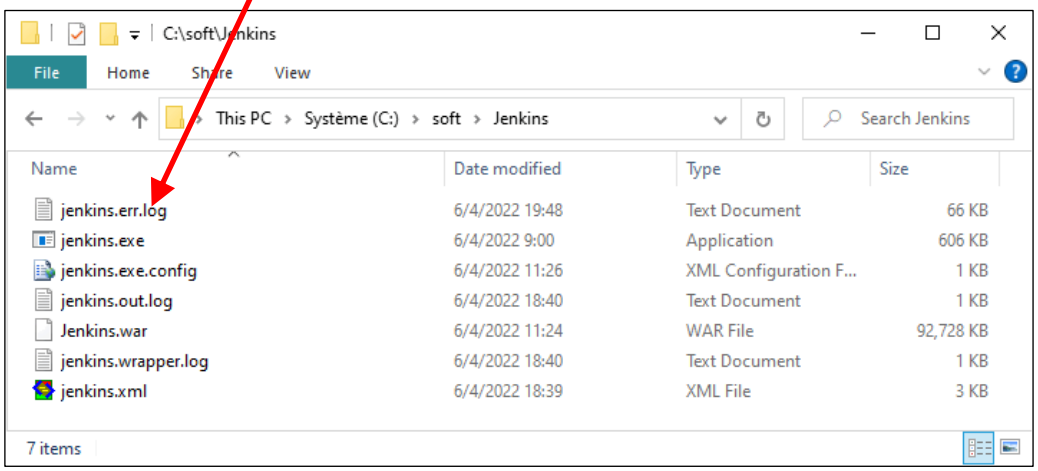
The good news is that you can always decide later to execute Jenkins under your own account rather than under the LocalSystem account. To change which user/account is executing Jenkins, follow the procedure given on step 11. of the section 4.8.2.

4.8.3.3. Diagnostic Jenkins Errors (e.g. port 8080 is already in use)

This section is interesting if:

1. The "Test Port" button used during step 8 of the Jenkins installation is reporting an error.
2. You can't see the Jenkins welcome webpage,
3. You cannot start the Jenkins service

You can get a description of the installation error inside the file named "jenkins.err.log" inside the Jenkins installation directory:



Here is an example of an installation failure (this is the content of the file "jenkins.err.log"):

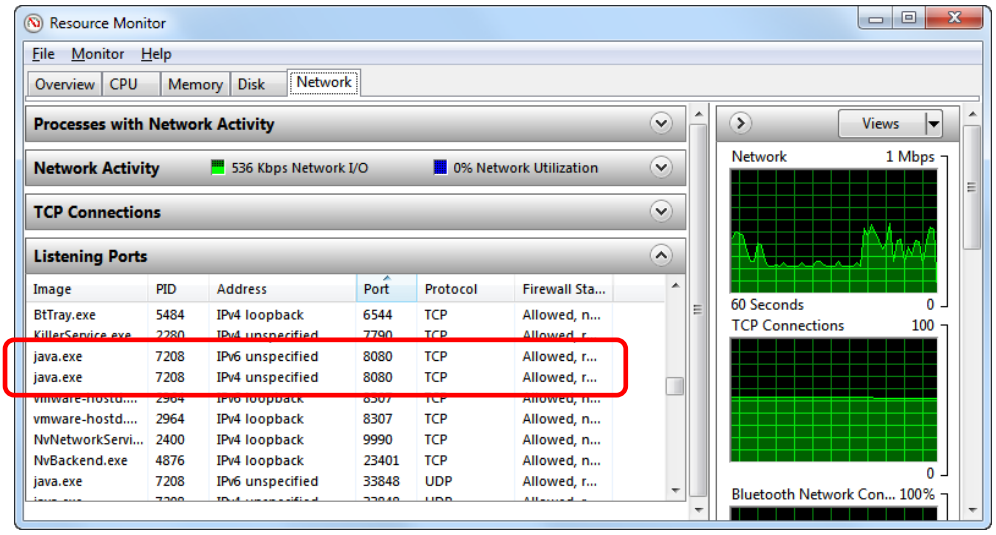
```
Nov 15, 2015 12:23:43 PM winstone.Logger logInternal
INFO: Beginning extraction from war file
Nov 15, 2015 12:23:44 PM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: jetty-winstone-2.8
Nov 15, 2015 12:23:46 PM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: NO JSP Support for , did not find org.apache.jasper.servlet.JspServlet
Nov 15, 2015 12:23:46 PM org.eclipse.jetty.util.log.JavaUtilLog warn
WARNING: FAILED SelectChannelConnector@0.0.0.0:8080: java.net.BindException: Address already in use: bind
java.net.BindException: Address already in use: bind
        at sun.nio.ch.Net.bind0(Native Method)
        at sun.nio.ch.Net.bind(Unknown Source)
```

If you got a "java.net.BindException" (such as in the example here above), it means that the default TCP/IP port that Jenkins is using (i.e. the port 8080) is already used by another application on your server.

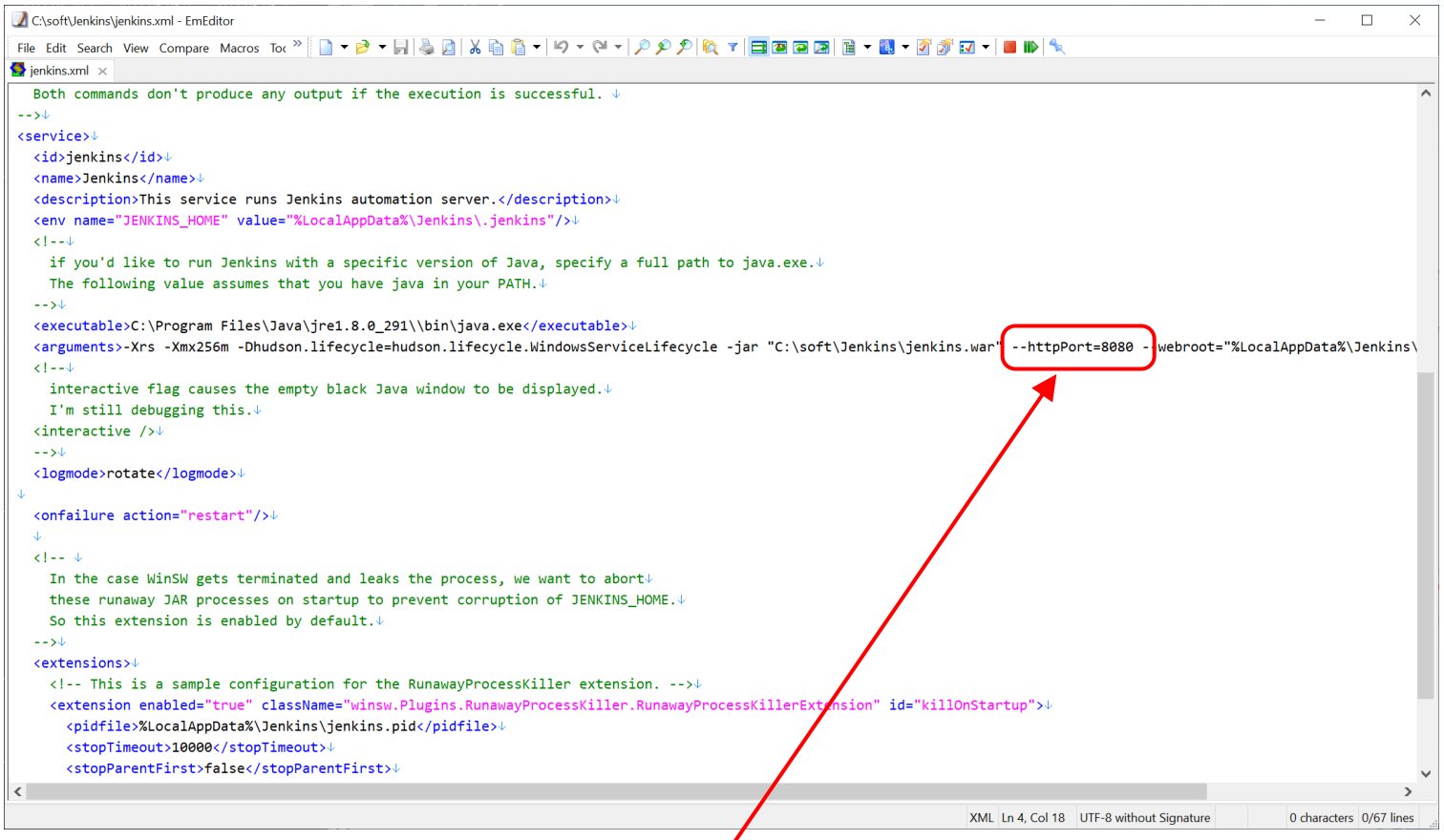
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When there is a “port” error, you have 2 choices:

1. You can **stop the other application** (that is currently using port 8080). To help you find the application using the port 8080: open a “Windows Task Manager” (i.e. right-click the “Task bar” and select “Task Manager” inside the context menu), go to the “Performance” tab, and click on the “Resource Monitor” button (on the bottom of the window). Inside the “Resource Monitor”, go to the “Network panel”, open the “Listening port” section and search for the application using the port 8080:



2. You can **configure Jenkins to use a different port**:
   2.1. Edit (with notepad) the file named “Jenkins.xml” located inside your Jenkins installation directory:

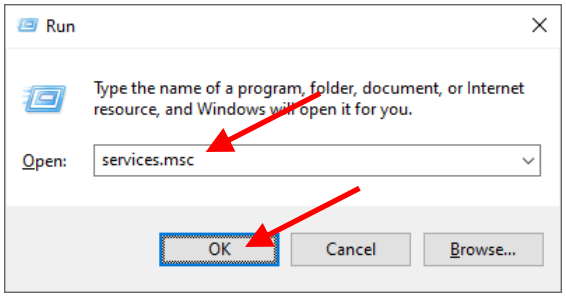


Write the new Jenkins port value here:

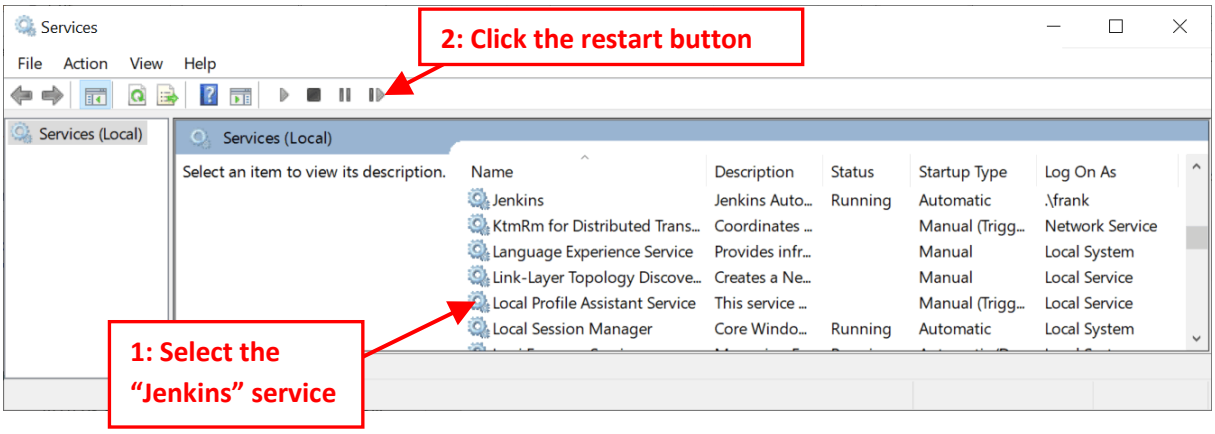
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2.2. **Optional (if Jenkins was already running):** Restart the Jenkins service:

- Open the "Service" Manager: Press simultaneously the keys [WIN]+[R] on your keyboard to open the "Run" window. Then write "services.msc" and click the "OK" button:



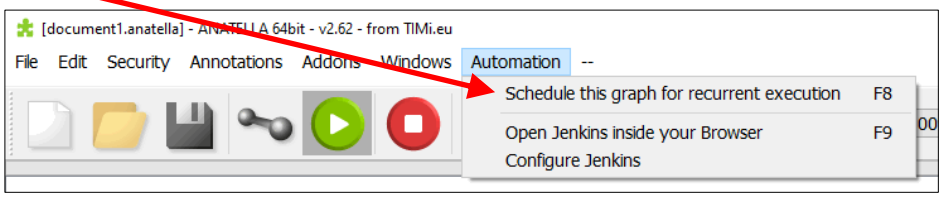
- Restart Jenkins inside the "Service" Manager window:



4.8.4. Scheduling your first Anatella graph with Jenkins.

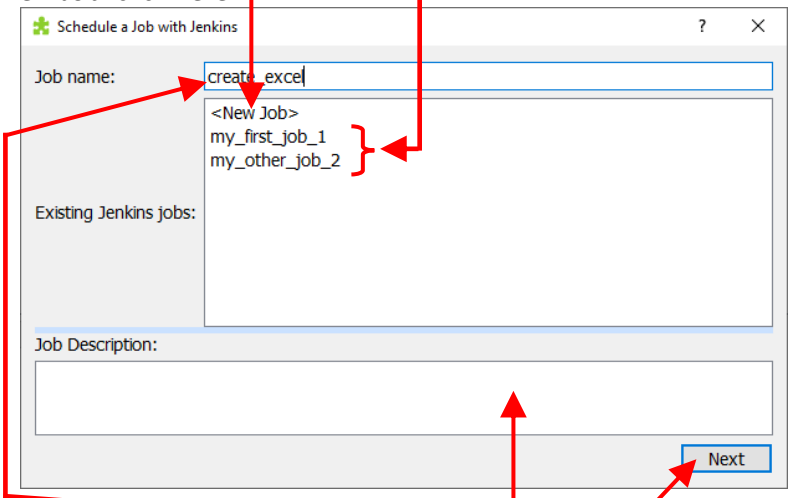
The easiest way to schedule an Anatella graph is to follow this procedure:

1. Open the Anatella graph to schedule inside Anatella.
2. Inside Anatella, open the "Automation" drop-down menu and select "Schedule this graph for recurrent execution":



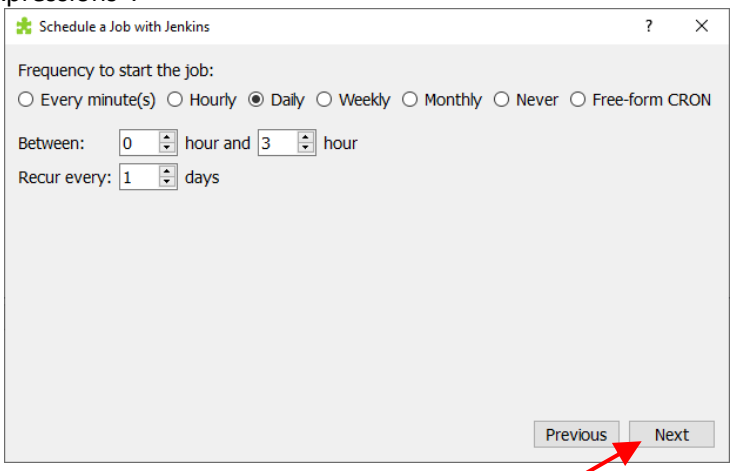
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. In the first configuration screen, you select if you want to:
   - **Update** an old Job: select it here:
   - **Create** a new Job: click here:



If we are creating a new job, we define its name here:
We can also edit the job description here:
Click the "Next" button to go to the next configuration screen:

4. In the second configuration screen, we select WHEN we want to run our graph: every minute, hourly, dayly, weekly, monthly, never. These settings are all self-explanatory. To have even more control on when the graph is running, we can also use here a free-form "CRON expression". Please refer to the section 4.8.6 for the explanation of the syntax used to create these "CRON expressions".



Click the "Next" button to go to the next configuration screen:

Actually, Jenkins supports many other ways to automatically starts an Anatella job (in addition to the "build periodically" option). For example, you can start an Anatella Job when: a specific email arrives, a connection to a specific URL happens, a file appears in a directory, etc. For more details on this subject, please refer to the "Jenkins build trigger plugins" listed here:

https://plugins.jenkins.io/ui/search?sort=relevance&labels=trigger&view=Tiles

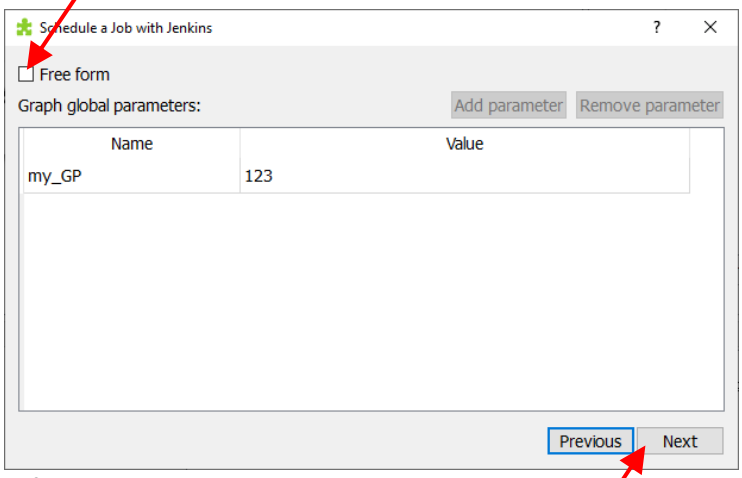
You must use the classical Jenkins web interface to parametrize these other special "triggers": see the step 23 inside the section 4.8.2. for more details on how to install new "triggers".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5. The third configuration screen only appears when, either:
   - You click the “Previous” button from the fourth&last configuration screen.
   - You have some “Global Parameters” defined inside your graph.

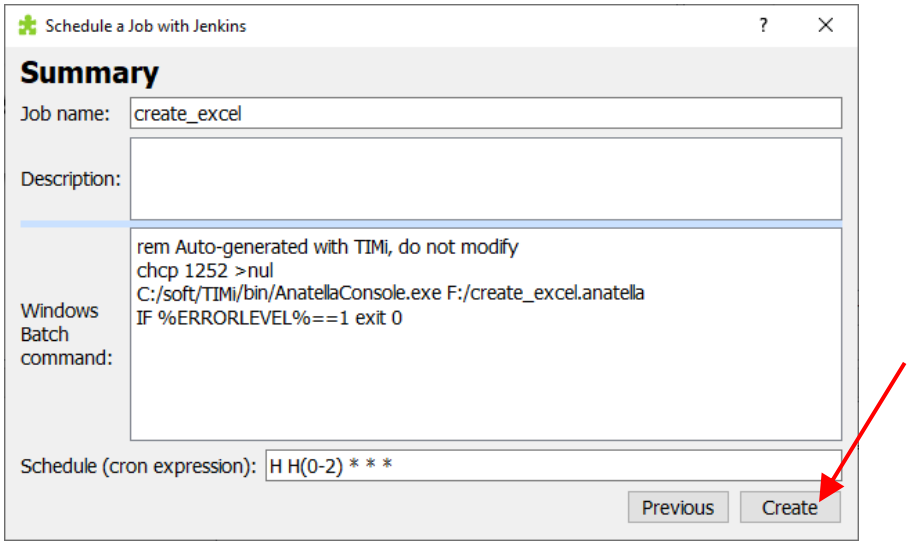
   This means that, most of the time, the third configuration screen will be totally skipped and invisible (i.e. you directly arrive to the fourth&last configuration screen).

   In the third configuration screen, you select if you want full control on the exact DOS-Batch Commands executed by Jenkins: In this case: you check this checkbox: . Warning: 99% of the time, you should let this checkbox unchecked.



   On this same configuration screen, you can also decide to change the value of some of the “Global Parameters” from you graph.
   Click the “Next” button to go to the next configuration screen:

6. The fourth and last configuration screen is a summary of all the settings entered in the previous screens. You cannot edit anything on this screen. Click the button in the bottom right corner to create (or update) your Anatella job inside Jenkins:



7. Congratulation! You just placed your first Anatella graph in production! Yes! 😊
   You should now go to the next section 4.8.5. to know how to TEST the graphs that are scheduled in production inside Jenkins.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The above procedure is the standard & minimal procedure to schedule the execution of an Anatella Graph using Jenkins. This is just a (very) simple example: Jenkins offers many more options (especially after installing the many extentions & plugins that are available). The standard web interface to Jenkins offers many more capabilities and settings. Do not hesitate to explore them!

4.8.4.1. A small explanation on the "Windows Batch commands" text field in the last configuration screen:

You can safely skip the remainder of this section if you are not interested.

The default windows batch commands are, typically:

```
Rem Auto-generated with TIMi, do not modify
chcp 1252 >nul
"c:/soft/TIMi/bin/AnatellaConsole" "<filepath_to_my_graph>"
IF %ERRORLEVEL%==1 exit 0
```

If you defined some "*Custom Jenkins Initialization Scripts*" (see the step 24 in the section 4.8.2. for more information about these scripts), you'll rather have something like this:

```
Rem Auto-generated with TIMi, do not modify
net use Z: \\My_File_Server\public /Y
chcp 1252 >nul
"c:/soft/TIMi/bin/AnatellaConsole" "d:/weeklyReport.anatella"
IF %ERRORLEVEL%==1 exit 0
```

…where we have:

- "`Rem Auto-generated with TIMi, do not modify`": is a marker that Anatella uses to detect if this set of batch commands are "free form" or auto-generated.

- "`chcp 1252 >nul`": is an optional command-line.

  This command allows you to use non-standard characters (e.g. accentuated characters such as é,è,ê,à,ù,etc.) inside the filePath of the Anatella graphs to run.

1252 is the most common code page (also called "character set") used by Java programs (Jenkins is a Java program). The 1252 code page stands for "Latin 1": it's the most common code page used in Europe/US.

- "`net use Z: \\My_File_Server\public /Y`" is an optional command-line to "mount" network drives (see the step 24 in the section 4.8.2.)

- "`c:/soft/TIMi/bin/AnatellaConsole`" is the location of the "Anatella Console" executable.

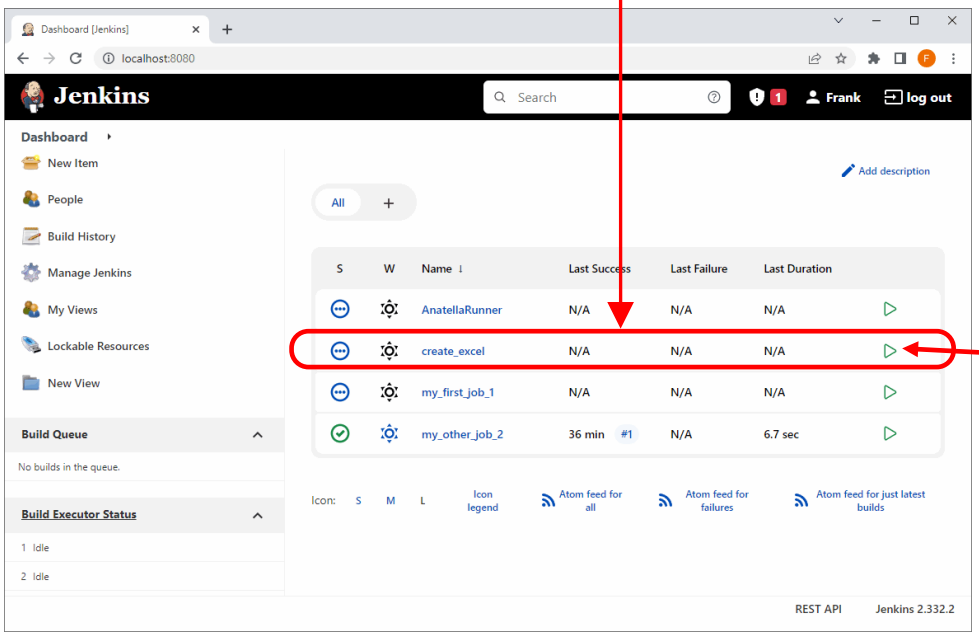
- "`d:/weeklyReport.anatella`" is the location of the Anatella graph to run.

- The last line (i.e. the line that contains "`IF %ERRORLEVEL%==1 exit 0`") is required to tell Jenkins that, when the "Error Level" returned by the Anatella engine (after the graph has finished running) is 1, it actually only means a simple "warning" and it does not mean a critical execution error of the Anatella Graph. Without this line, Jenkins will interpret all simple warnings as critical errors (and this is usually quite bad).

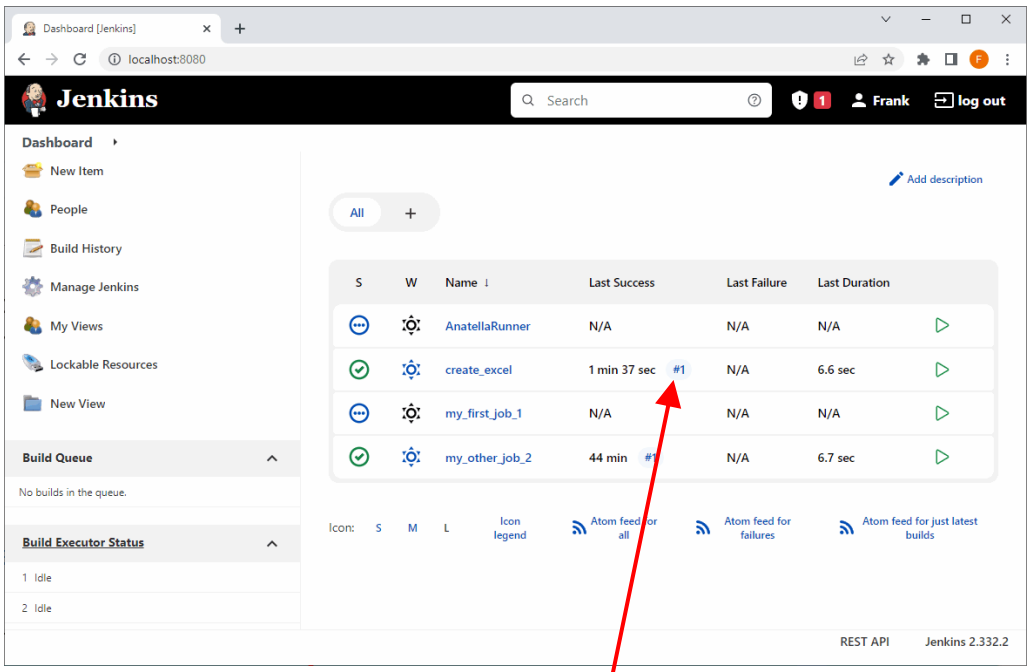
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.8.5. Testing your scheduled Graphs.

Once an Anatella graph is added inside Jenkins it appears inside the Jenkins web interface: For example, in the example from the section 4.8.4. here above, we just finished adding into Jenkins the Anatella graph named “create_excel”. It’s now visible here:



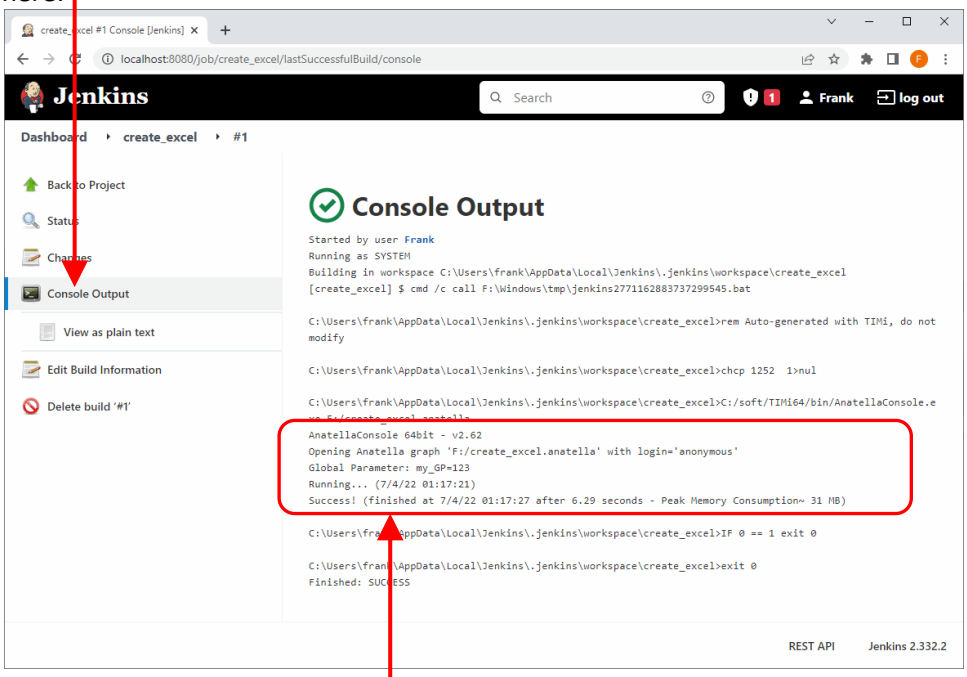
At this point, no need to wait for Jenkins to run the graph: We can execute it directly by clicking here: on the ▷ icon. After a few seconds, refresh the webpage (press F5 or click the ⟳ refresh button in your internet browser): You should see:



To see the content of the Anatella log window, click here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

..and then click here:

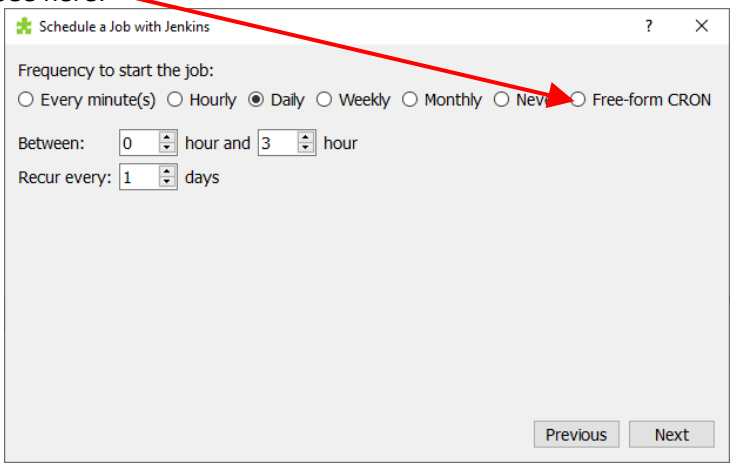


The content of the Anatella log window is here: . Check this log to see:

- .. if your graphs worked as expected.
- .. easily find & fix any errors in your graphs.

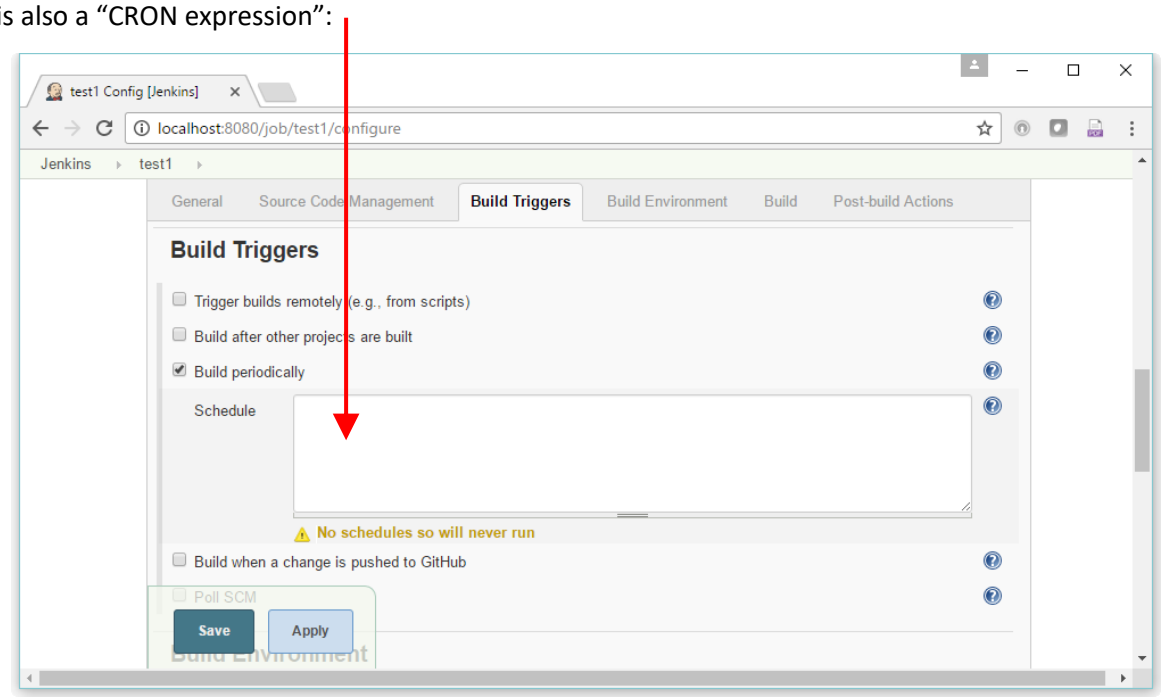
4.8.6. Defining when your graphs are running (CRON expression)

When we use the Anatella interface to schedule the excecution of our graphs inside Jenkins, in the second configuration screen, we can use a free-form “CRON expression” to define precisely when the graph will run. See here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In the same way, using the classical Web interface to Jenkins, we can define when our Anatella graphs are running using the “Build periodically” option that takes a Free-form parameter named “Schedule” that is also a “CRON expression”:



These “CRON expressions” are strings that follow the syntax of the famous UNIX tool named “CRON” (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace. These 5 fields are “MINUTE HOUR DOM MONTH DOW”:

More precisely, we have:

1. MINUTE: Minutes within the hour (0–59)
2. HOUR: The hour of the day (0–23)
3. DOM: The day of the month (1–31)
4. MONTH: The month (1–12)
5. DOW: The day of the week (0–7) where 0 and 7 are Sunday.

To specify multiple values for one field, the following operators are available. In the order of precedence:

- ***** : specifies all valid values
- **M-N** :specifies a range of values
- ***/X** :specifies steps by intervals of X through the whole valid range
- **M-N/X** :specifies steps by intervals of X through the specified range
- **A,B,...,Z** :enumerates multiple values

To allow periodically scheduled tasks to produce even load on the system, the symbol **H** (for “hash”) should be used wherever possible. For example, using “**0 0 * * ***” for a dozen daily jobs will cause a large spike at midnight. In contrast, using “**H H * * ***” would still execute each job once a day, but not all at the same time, better using limited resources.

The **H** symbol can be used with a range. For example, “H H(0-7) * * *” means to launch a job between 00:00 AM (midnight) to 7:59 AM. You can also use step intervals with **H**, with or without ranges. The **H** symbol can be thought of as a random value over a range, but it actually is a hash of the job name, not a random function, so that the value remains stable for any given project.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Beware that for the day of month field, short cycles such as "***/3**" or "**H/3**" will not work consistently near the end of most months, due to variable month lengths. For example, "***/3**" will run on the 1st, 4th, …31st days of a long month, then again the next day of the next month. Hashes are always chosen in the 1-28 range, so "**H/3**" will produce a gap between runs of between 3 and 6 days at the end of a month. (Longer cycles will also have inconsistent lengths but the effect may be relatively less noticeable.)

Empty lines and lines that start with **#** will be ignored as comments.

In addition, the following codes:

@yearly @annually @monthly @weekly @daily @midnight @hourly

…are supported as convenient aliases. These use the hash system for automatic balancing. For example, the code "**@hourly**" is the same as "**H * * * ***" and could mean at any time during the hour. The code "**@midnight**" actually means some time between 00:00 AM (midnight) and 2:59 AM.

Some examples:

Description	CRON String
Every day between midnight and 2:59 AM	@midnight *or* H H(0-2) * * *
Every fifteen minutes (perhaps at :07, :22, :37, :52):	H/15 * * * *
Every ten minutes in the first half of every hour (three times, perhaps at :04, :14, :24)	H(0-29)/10 * * * *
Once every two hours at 45 minutes past the hour starting at 9:45 AM and finishing at 3:45 PM every weekday (excluding weekends).	45 9-16/2 * * 1-5
Once in every two hours slot between 9 AM and 5 PM every weekday (perhaps at 10:38 AM, 12:38 PM, 2:38 PM, 4:38 PM)	H H(9-16)/2 * * 1-5
Once a day on the 1st and 15th of every month except December	H H 1,15 1-11 *
Once a week, every sunday	H H * * 0

4.8.7. Backing up Jenkins

In addition to disaster recovery, Jenkins backups are useful insurance against accidental configuration changes, which might be discovered long after they were made. A regular backup system lets you go back in time to find the correct settings.

Manually Creating Jenkins Backup

Jenkins stores everything under the Jenkins Home directory, $JENKINS_HOME, so the easiest way to back it up is to simply backup the entire $JENKINS_HOME directory. Even if you have a distributed Jenkins setup, you do not need to backup anything on the build agent side.

To find the $JENKINS_HOME location, go to the Configure System menu.

Another backup planning issue is whether to do backup on live instances without taking Jenkins offline. Fortunately, Jenkins is designed so that doing a live backup works fine (configuration changes are atomic), so backups can be done without affecting a running instance.

The following directories contain bits that can be easily recreated, so you don't need to include these in the backup:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

* /war (exploded war)
* /cache (downloaded tools)
* /tools (extracted tools)

Optimization: Backup a Subset of $JENKINS_HOME

Although $JENKINS_HOME is the only directory you need to backup, there's a catch: This directory can become rather large. To save space, consider what parts of this directory you really need to backup and back them up selectively.

The bulk of your data, including your job configuration and past filed records, lives in the */jobs* directory. The */jobs* directory holds information pertaining to all the jobs you create in Jenkins. Its directory structure looks like this:

*/jobs/**
– *builds* (build records: you can usually delete this from backup)
– *builds/*/archive* (archived artifacts: delete this from backup)
– *workspace* (checked-out workspace from github: delete this from backup)

The */builds* directory stores past build records. So, if you're interested in configuration only, don't backup the builds. The workspace directory contains the files that you check out for the version control systems. Normally these directories can be safely thrown away.

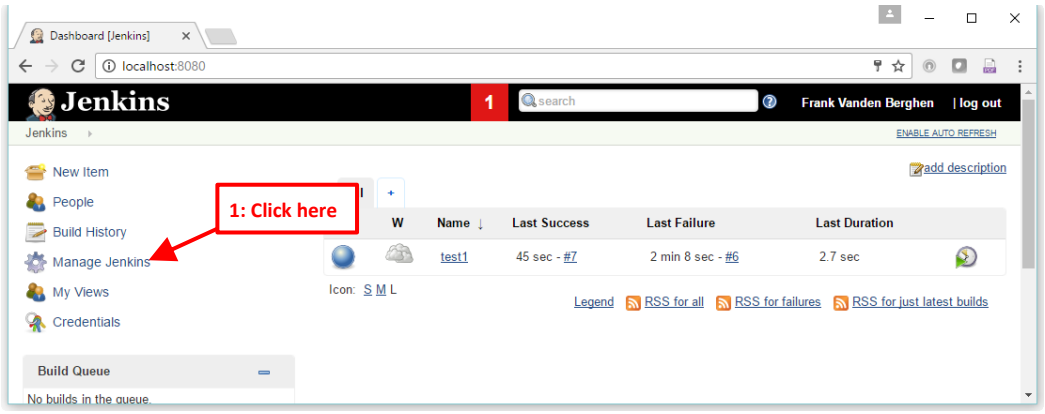
4.8.8. Configuring Jenkins to send email notifications in case of errors

To send notification emails, you must:

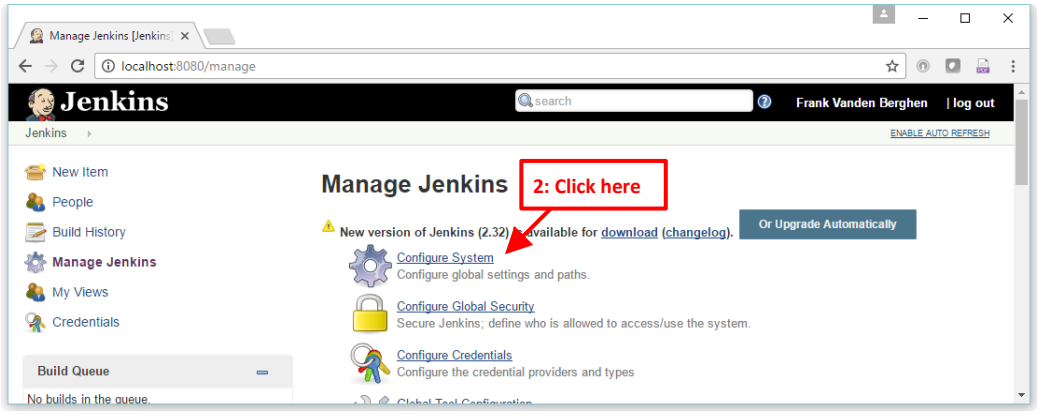
1. Configure the SMTP server used by Jenkins to send email: This is a one-time only setup (see next section 4.8.8.1 for more details).
2. Configure each Jenkins Job to send the email to the required persons. This is a "per-job" setup (see next section 4.8.8.2. for more details).

4.8.8.1. Configuration of the SMTP server used by Jenkins

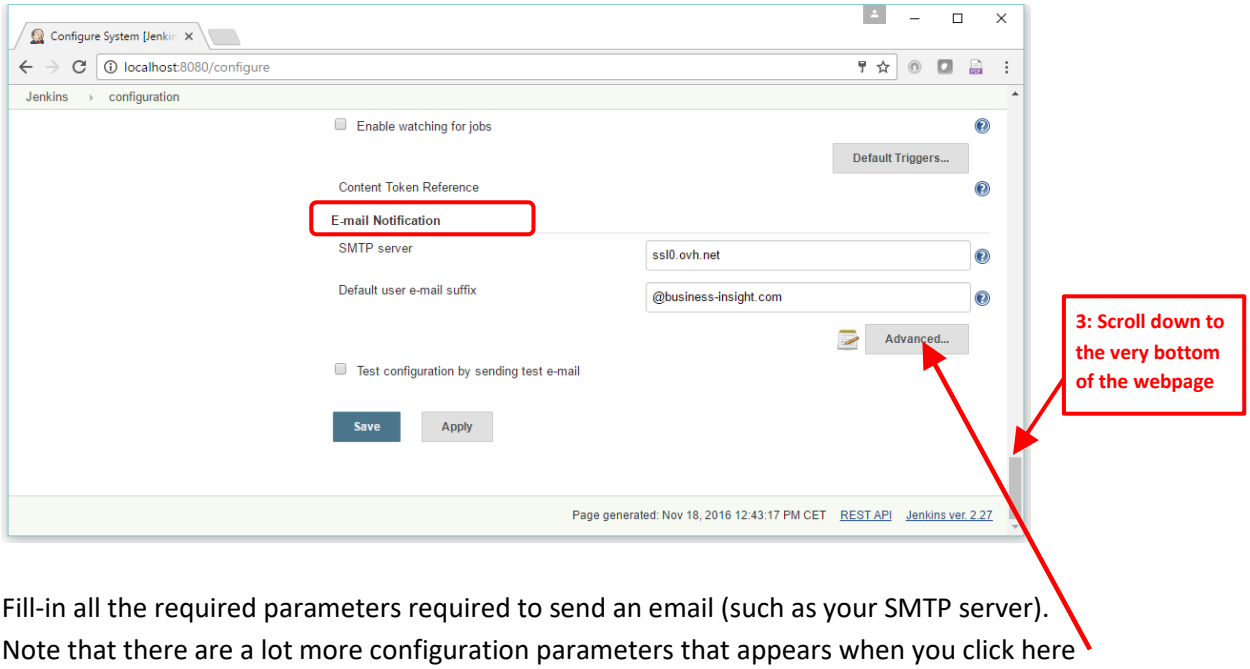
Go to the "Email notification" panel in three steps:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



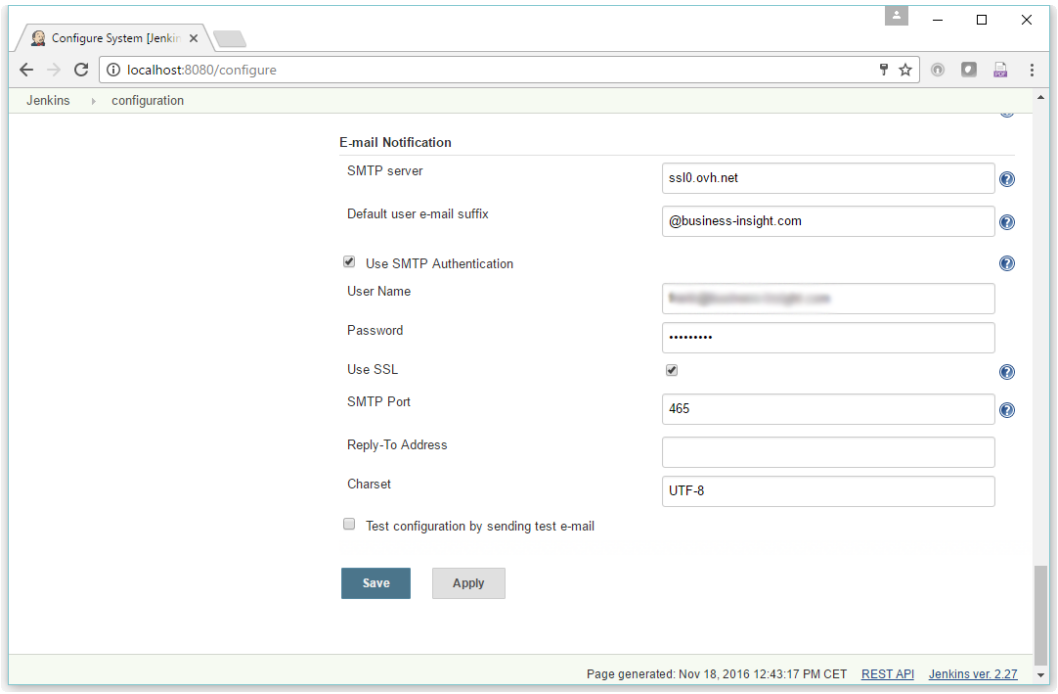
Scroll down to the bottom of the page: The panel to configure “Email notification” is there:



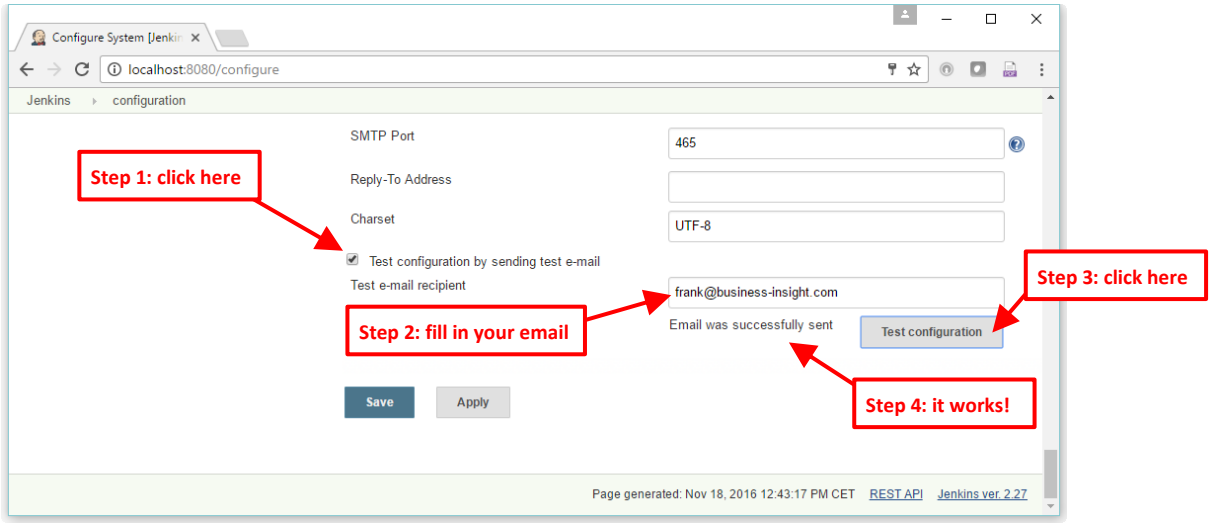
Fill-in all the required parameters required to send an email (such as your SMTP server).
Note that there are a lot more configuration parameters that appears when you click here

Here are the parameters required to use a SMTP server with SSH encryption (SSL) and authentication:
(These are typical parameters for OVH SMTP servers)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

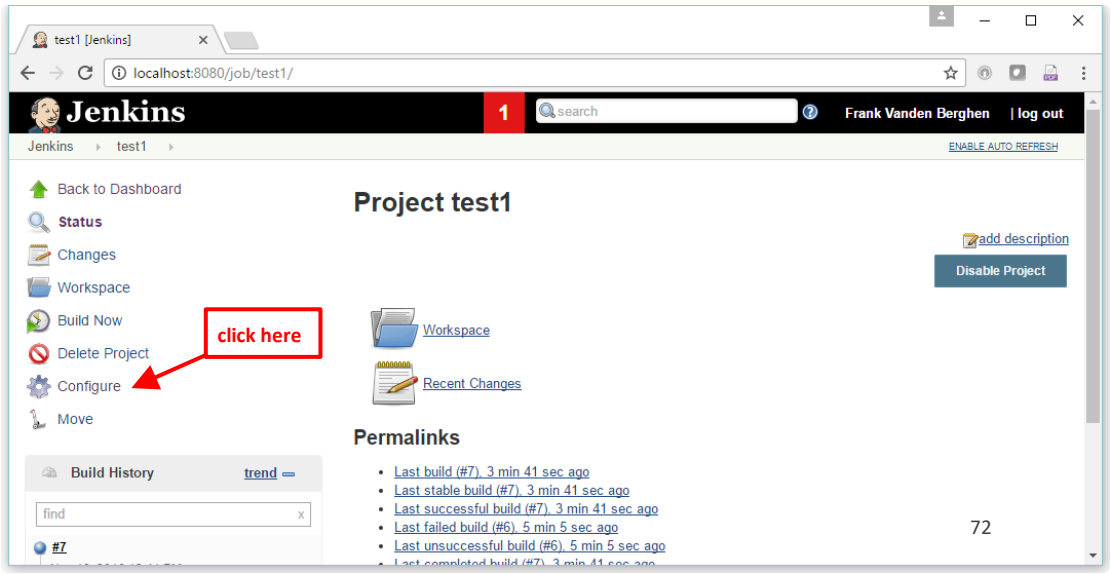


You can already test your settings:

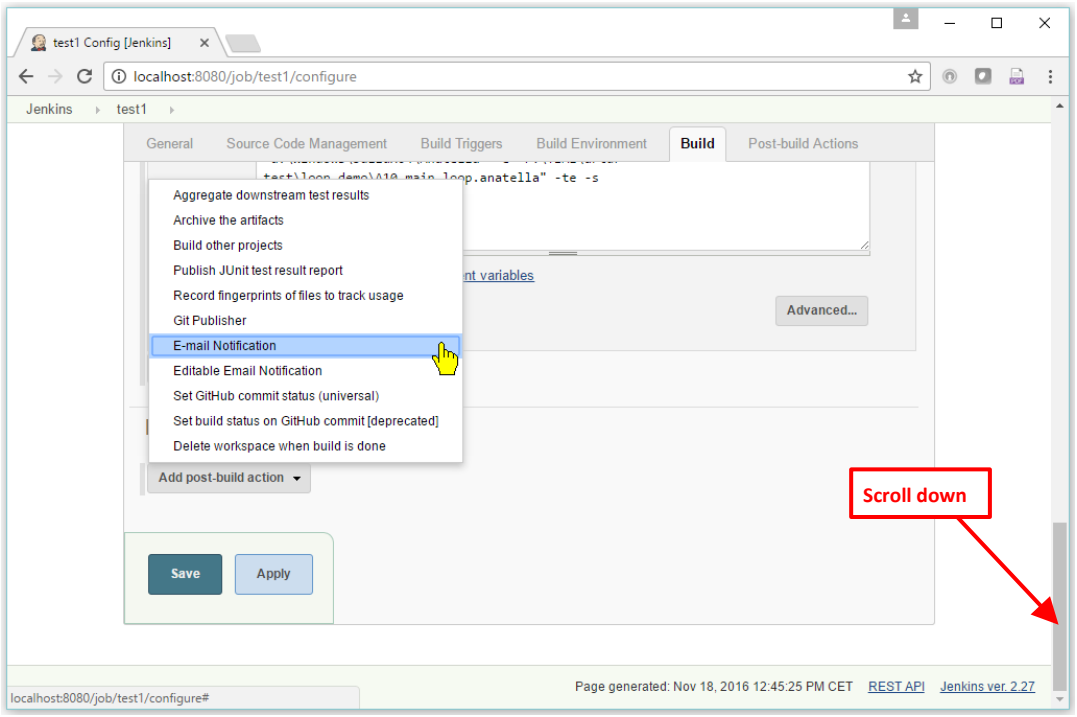


4.8.8.2. Per-Job configuration

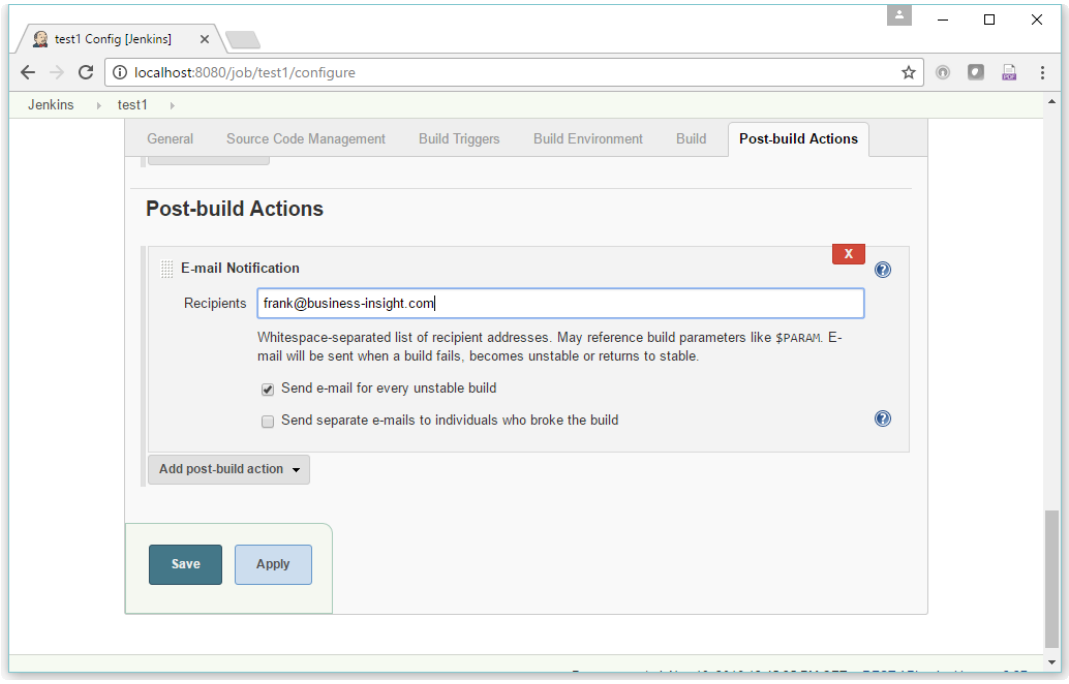
Go to the "configuration" page of your job:



Scroll down to the “Post-build action” section, and add the post-build action named “Email Notification”:

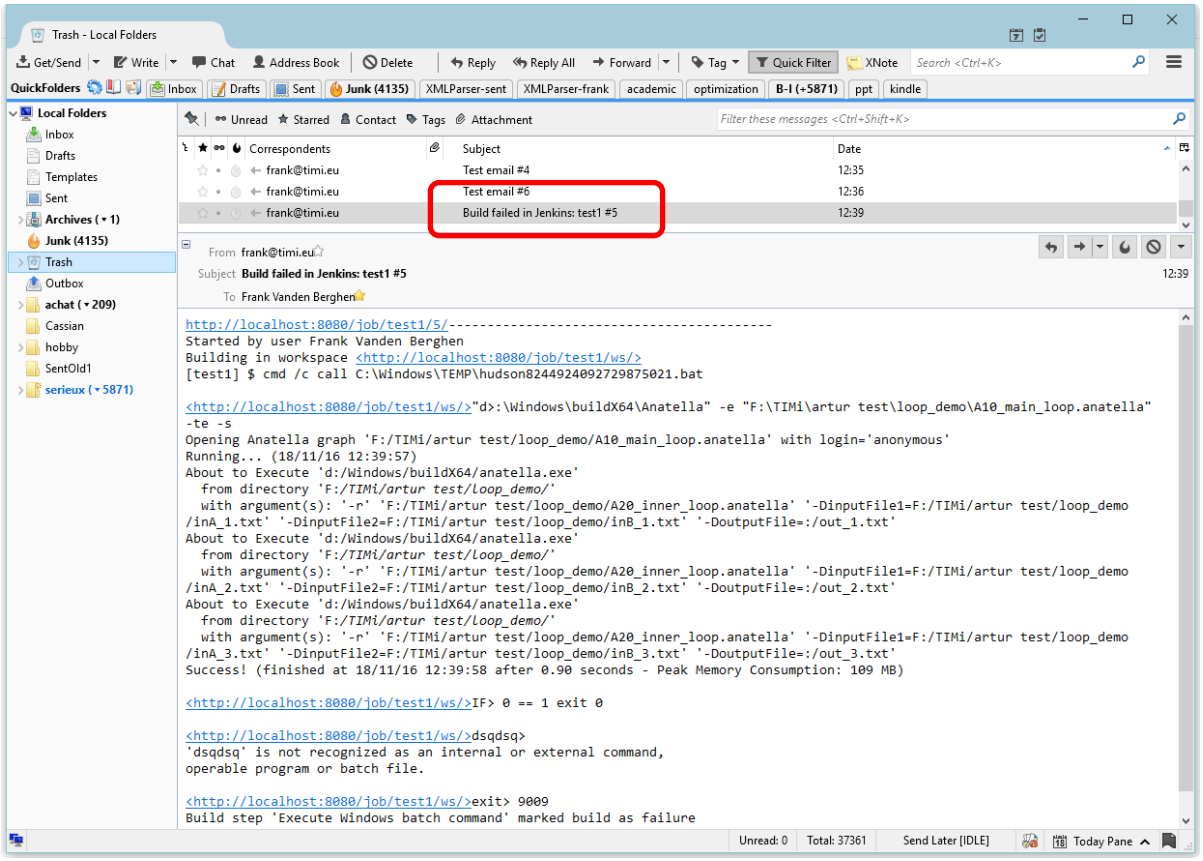


Just enter your email and click the “Save” button:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Now each time a job fails, you get an email: For example:



4.8.9. Configuring the Jenkins Execution Queue

The Jenkins "Execution Queue" contains all the graphs that are waiting to be executed at a given moment in time. There are, basically, two different ways to "fill-in" the Jenkins "Execution Queue":

1. **Through Jenkins:** For example, you can configure Jenkins to execute an Anatella graph X everyday at midnight (see section 4.8.4. for more information about "scheduling" graphs). This means that, everyday at midnight, Jenkins automatically adds the Anatella graph X inside the "Execution Queue" (so that it is executed as soon as possible).
2. **Through Anatella:** The "Loop Jenkins" Action (see section 5.21.1) instantaneously adds many Anatella graphs inside the Jenkins "Execution Queue".

All the Anatella graphs that are inside the Jenkins "Execution Queue" are executed as soon as possible using all the computing power available inside all the computers managed by Jenkins.

In technical terms, a server where Jenkins is installed & running (and that is thus able to execute some Anatella graphs) is called a "node".

Jenkins has several options that allows you to configure exactly what's the available "computing power". More precisely:

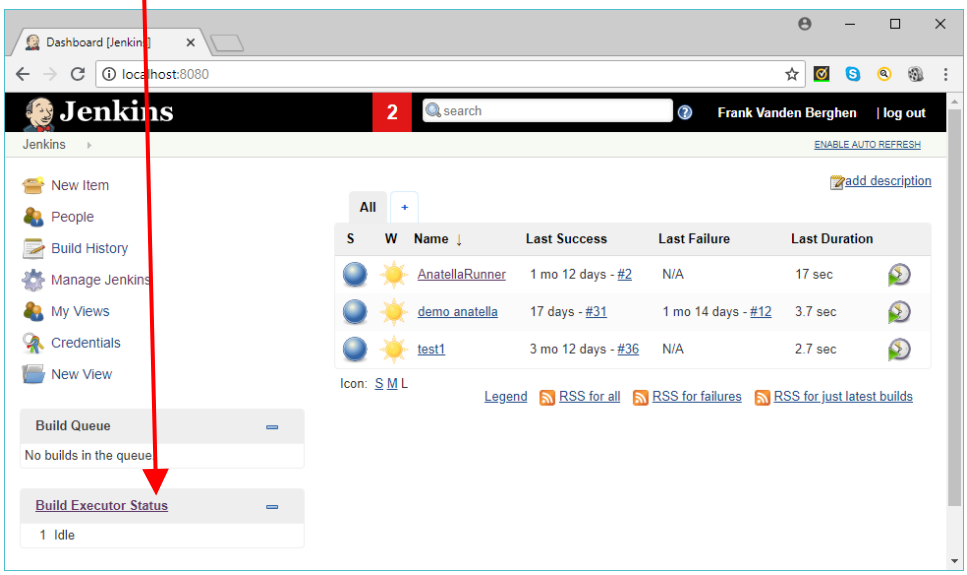
1. You can add/remove "nodes".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. For each “node”, you can decide the maximum number of graphs that can be executed simultaneously.

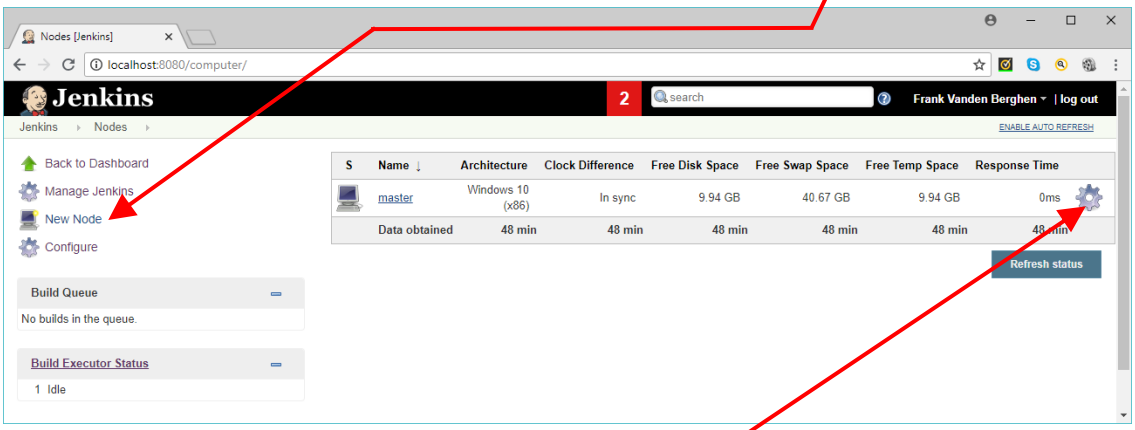
1. To Add/Remove “nodes”

1.1. Open a browser and click on “Build Executor Status” link in the bottom left corner of the “Jenkins” home page:



This opens a webpage where you can configure each node managed by Jenkins.

1.2. To add a new computing “Node”, click on the “New Node” link here:

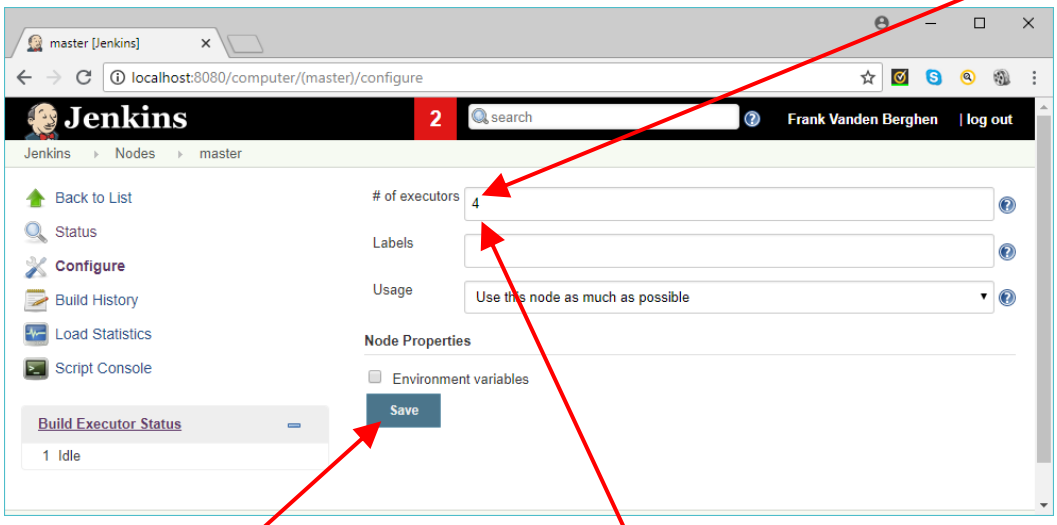


2. To set the maximum number of graphs that will be executed simultaneously on a node

2.1. Click on the node that you want to configure. For example, if you want to configure the “master” node, click in the gear icon, here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2.2. The maximum number of Anatella graphs that will be executed simultaneously on the "master" node (i.e. the "number of executors") is defined here (in the example, it's "4"):



2.3. Click on the "Save" button

A good value for the "number of executors" (here:) would be the number of CPU cores on the machine. Setting a higher value will cause each graph to take a longer time to execute, but could increase the overall throughput. For example, one graph might be CPU-bound, while a second graph running at the same time might be I/O-bound — so the second graph could take advantage of the spare I/O capacity at that moment.

Setting the number of executors to zero (for a node N) will prevent any graphs from being executed on the node N.

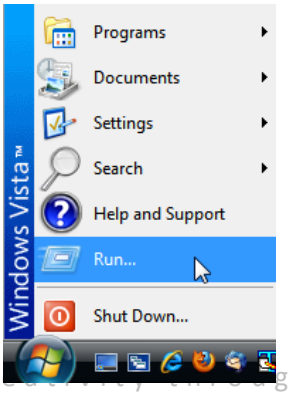
4.8.10. Using the Microsoft Windows Task Scheduler with Anatella

For very basic scheduling needs, you can use the "*Microsoft Task Scheduler*" instead of Jenkins.

The first step is to find & open the "*Microsoft Task Scheduler*". You can usually find the Microsoft "*Task Scheduler*" application inside the "Administrative tools" folder of your "Start Menu".

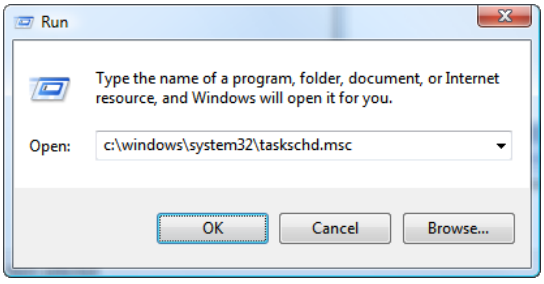
The icon of the Microsoft "*Task Scheduler*" application is: .

If you can't find the "*Task Scheduler*" application in your "Start Menu", click on the "Run" button in the start menu:

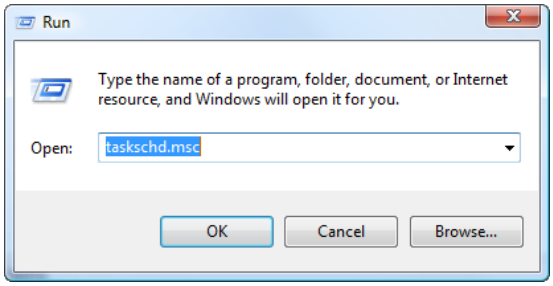


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

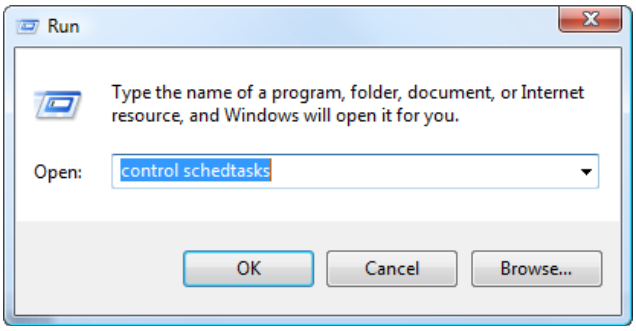
You can also enter: “c:\windows\system32\taskschd.msc” into the textbox and click the OK button:



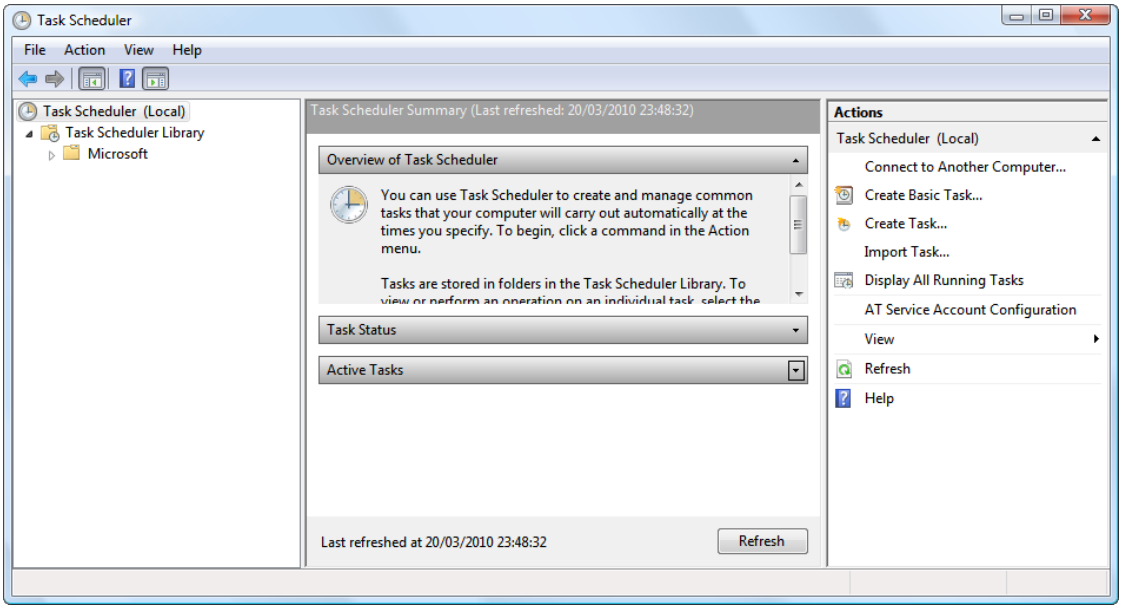
Alternatively, enter: “taskschd.msc” into the textbox and click the OK button.



Alternatively, enter: “control schedtasks” into the textbox and click the OK button.

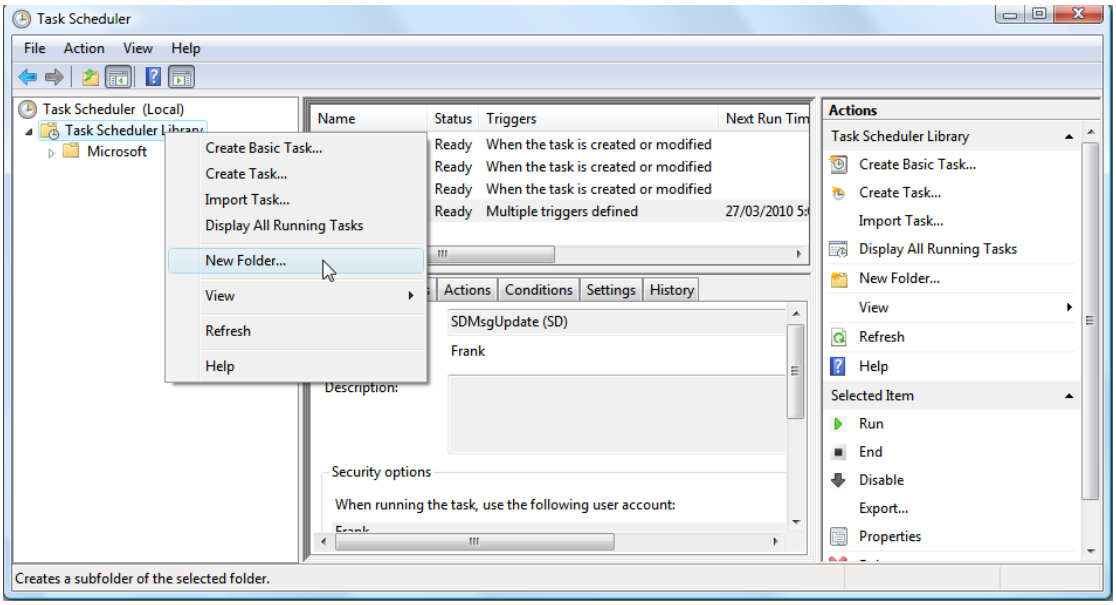


The Microsoft “*Task Scheduler*” window should now appear:

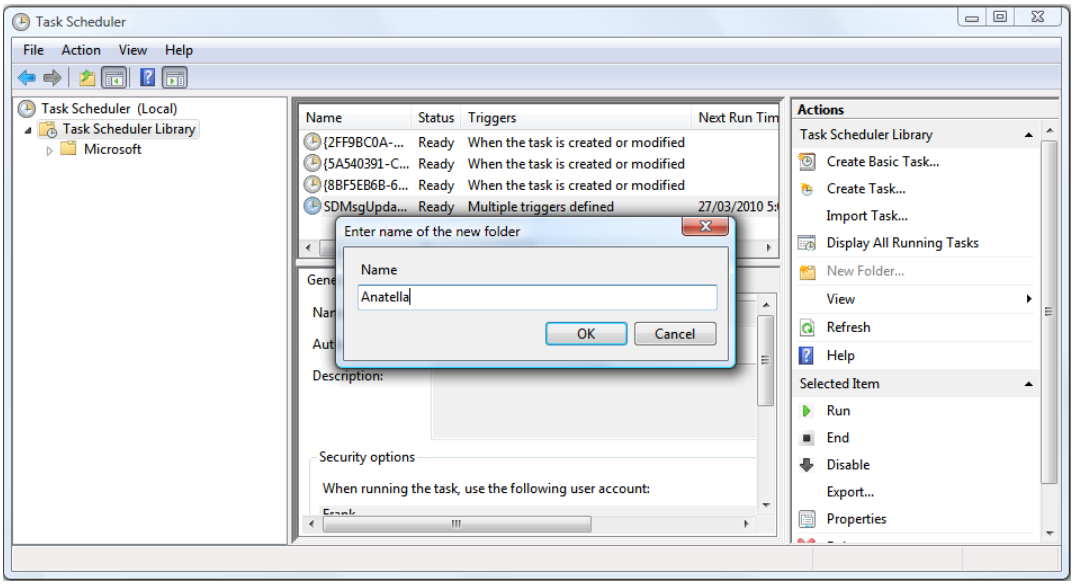


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Right-click on the "*Task scheduler Library*" folder to open a context-menu.
In this context-menu, select "*New folder*":

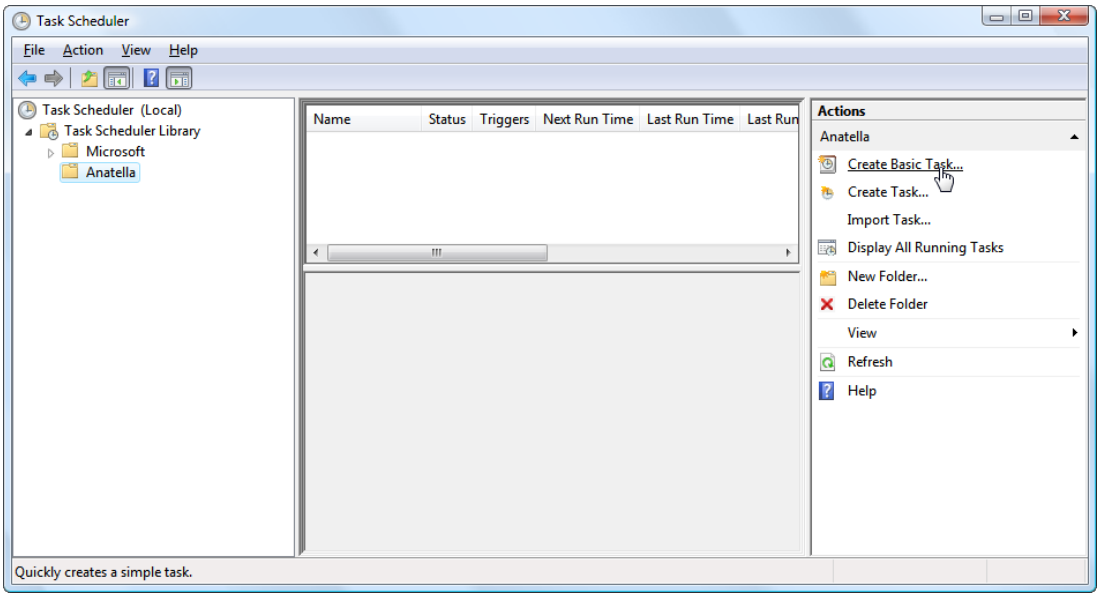


Enter "Anatella" as folder name and press enter:

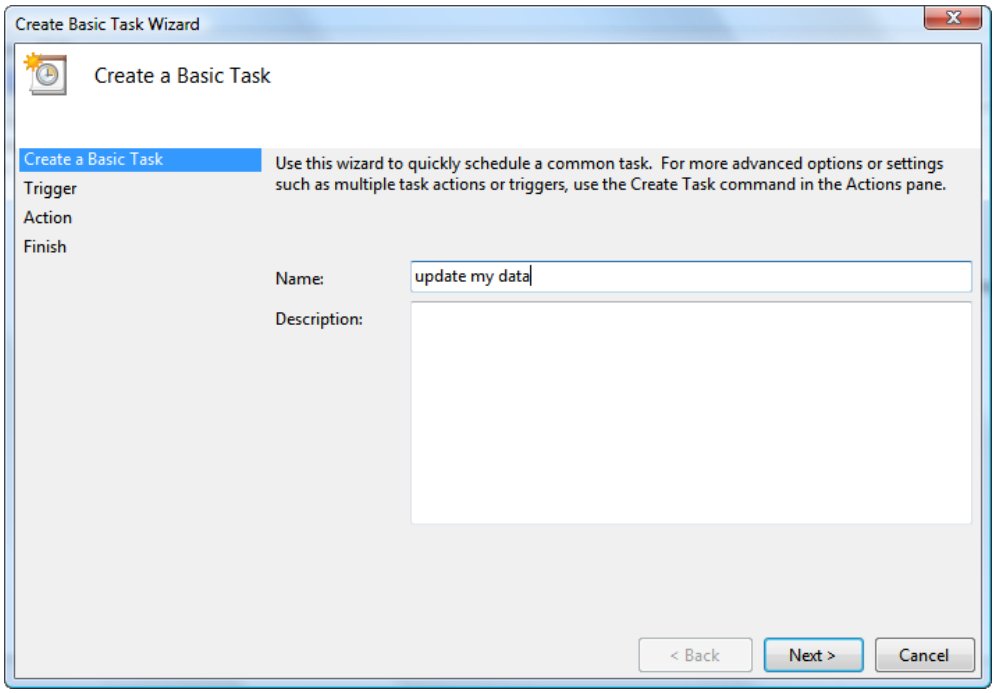


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

As an example, we will create a new task named “update my data”. This new task runs the Anatella graph “d:\my_graph.anatella” every Monday of every week. Select the “Anatella” folder and click on “Create Basic Task...”:

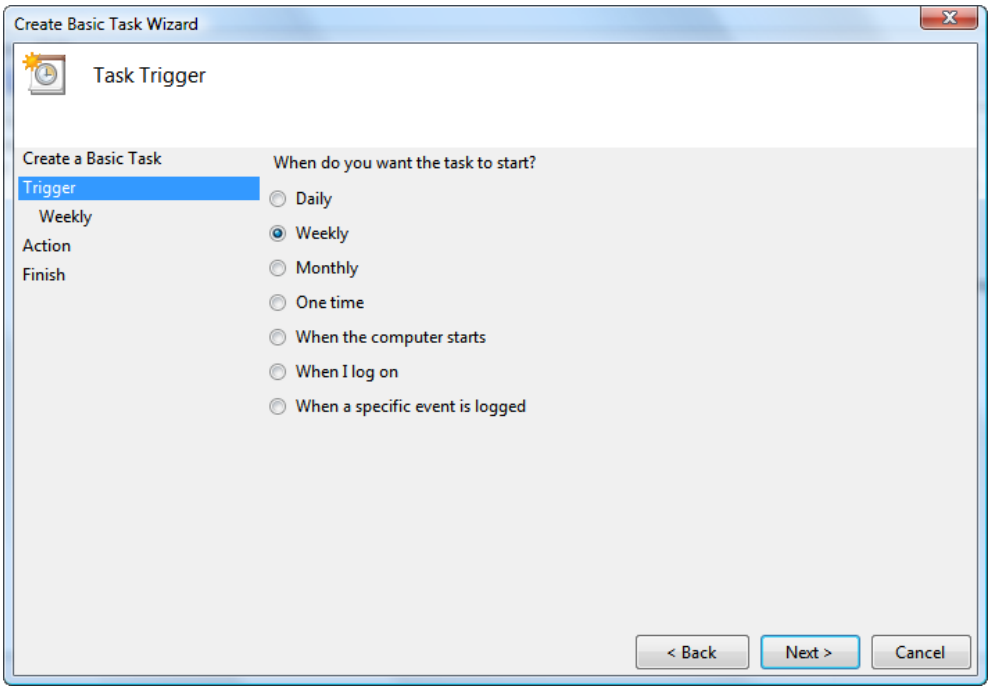


Enter “update my data” as name of the task and click the “Next” button:

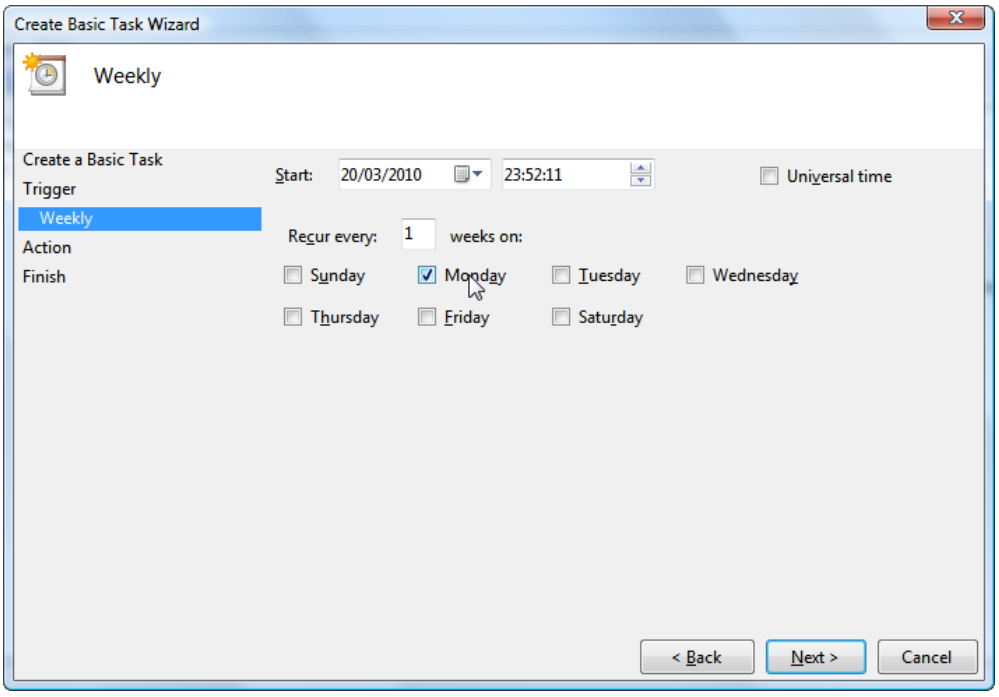


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Select “Weekly” and click the “Next” button:

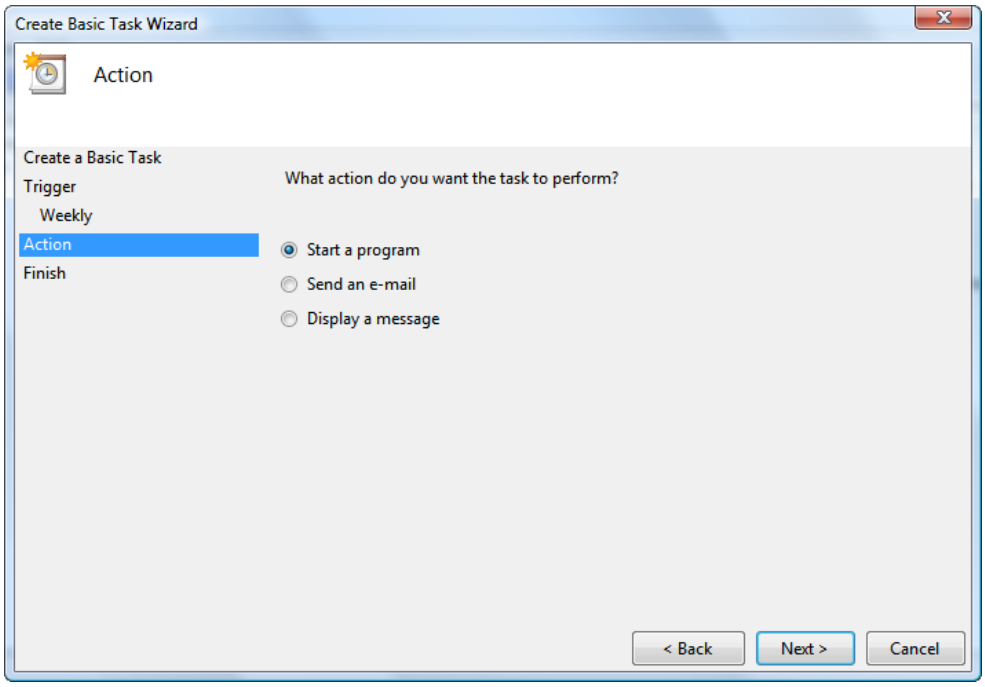


Select “Monday” and click the “Next” button:



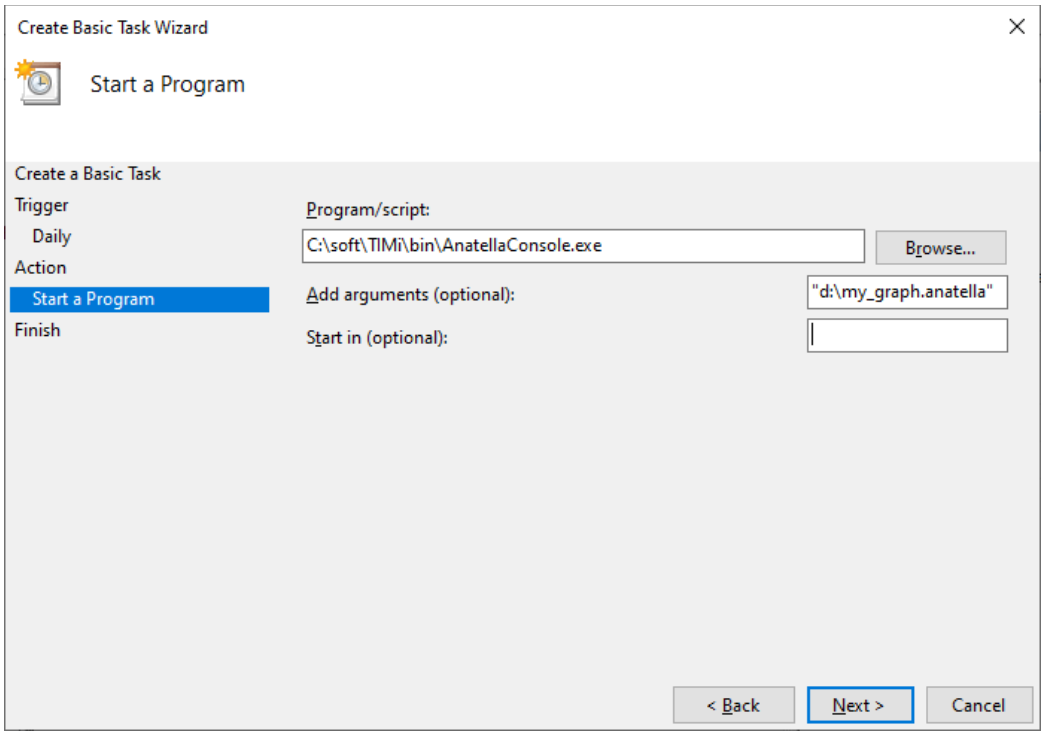
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Select “Start a program” (this is the default parameter) and click the “Next” button:



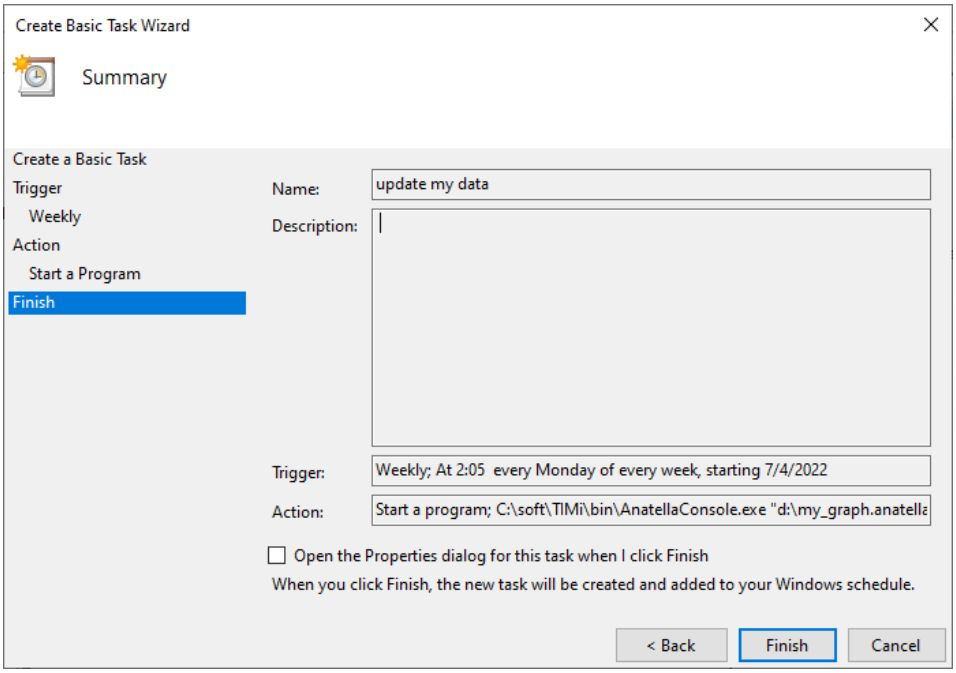
Enter (between quotes) the full path to the AnatellaConsole executable inside the “Program/Script” textbox. In the “argument” textbox, enter the full path (between quotes) of the Anatella-graph to execute. Once the 2 fields are setup correctly, click the “Next” button.

You should have something like this:

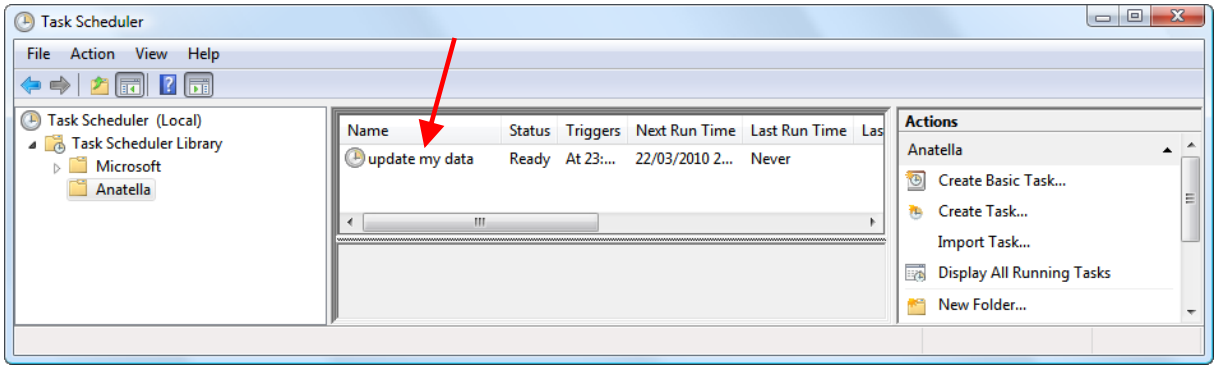


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Click the “Finish” button:



You can see the new Anatella Tasks named “update my data”. This task has been added to the list of tasks to perform. From now-on, your Anatella-graph will be executed every Monday of every week.



4.9. Running an Anatella graph from another language (i.e. from Python, Jenkins, Php, Perl, etc.)

When running an Anatella graph from another language or from Jenkins, you might want to monitor in “real-time” the correct execution of your graphs. To do so, you want to “have a look” at the content of the Anatella-Log-Window. To send a copy of the log window content back to the calling process (thourgh the “stderr” channel), the preferred solution is to run:

```
AnatellaConsole.exe mygraph.anatella
```

See the section 4.7. for the complete description of all the available command-line parameters.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.9.1. Deprecated Solution (you can disregard this whole section)

Instead of using "AnatellaConsole.exe", in some unnommon situation, you might want to use the "Anatella" executable with the "-te" option (this is the old, **deprecated** solution). The "-te" option outputs the content of the log window to the "Standard Error Output" (stderr). For example:

```
Anatella.exe -e mygraph.anatella  -te
```

Then you get a graphical window that allows you to monitor the messages that are passing on the "StdErr" to detect any errors during the execution of your Anatella graph.

Please note that, the calling process *must* read the characters arriving on the "StdErr". If the calling process does not read any characters arriving on the "StdErr", then Anatella stops running, waiting for these characters to be read.

If the calling process does not read "fast enough" the characters arriving on the "StdErr", Anatella will be "blocked" all the time (waiting for the character-buffer to be flushed) and Anatella will thus run at a very low speed.

This means that this option can severely impede the speed of the Anatella computations (especially when there are many messages produced inside the log window): Use this option very carefully: Before using this option, I strongly suggest you to do some testing (i.e. measure the computation time).

When there are some errors (or warnings) during the Batch Execution of an Anatella Graph, you might want to "have a look" at the description of these errors inside the Anatella-Log-Window (to be able to understand them and fix these errors). Thus, when there are some errors (or warnings), the Anatella-Log-Window can't suddenly disappear (to allow you to "have a look" at the errors) and thus the Anatella process can't close as soon as the computations are finished:

This behavior (i.e. the fact that, by default, "Anatella.exe" **does not always close** immediately after the end of the computations) is annoying when you call Anatella from Jenkins or from another language: i.e. Indeed, when running an Anatella graph from another language (or from Jenkins), you want to be sure that, as soon as the computations are finished, Anatella **always closes**, so that:

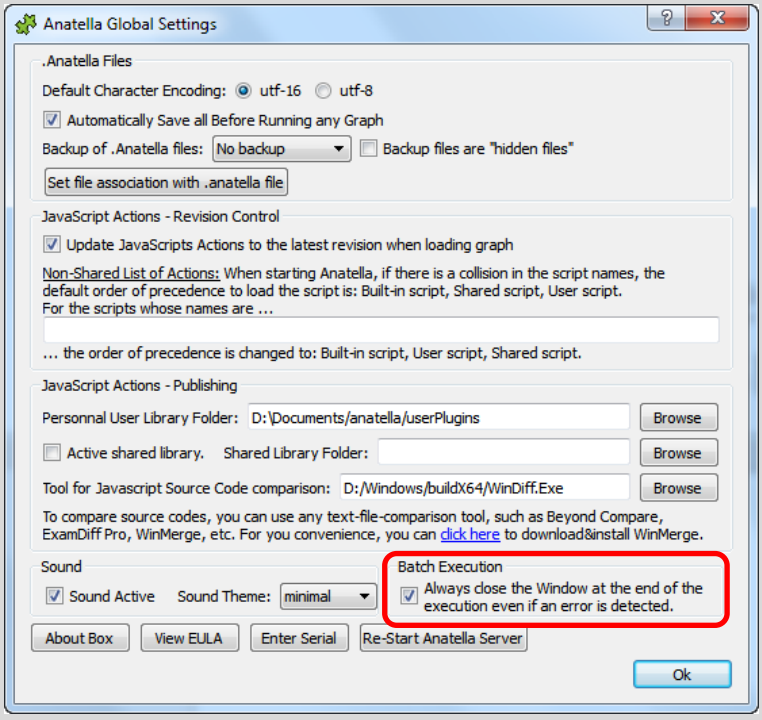
- …your Python, Php, etc. script can proceed further along.
- …Jenkins can report the error to you.

There are several ways to guarantee that the Anatella process always closes at the end of the computations, but the easiest & preferred solution is simply to run "AnatellaConsole.exe" (and never use "Anatella.exe").

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

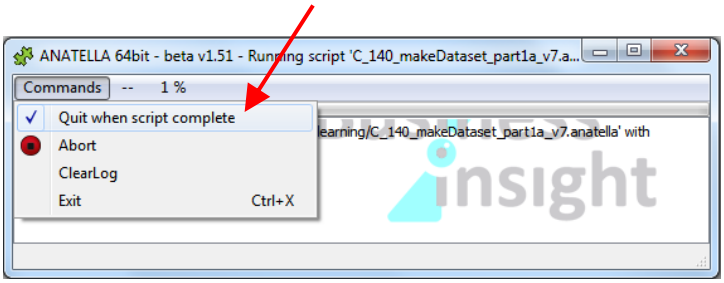
Alternatively, here are two other solutions that both guarantee that the "Anatella.exe" process always closes at the end of the computations (WARNING: THESE ARE "LEGACY" AND "DEPRECATED" SOLUTIONS: DO NO USE THEM!):

1. Use the "-s" option (i.e. the "silent" mode option) on the command-line. Since no Anatella-Log-Window is visible, it's impossible for you to to "have a look" at the errors, and thus "Anatella.exe" always closes immediately at the end of the computations.
2. Enable the "Always close" parameter inside the "Anatella Global Settings":



The "Always close" parameter is a computer-wide parameter: It's global for the whole machine. If different graphs require different behaviors, use the "-s" option.

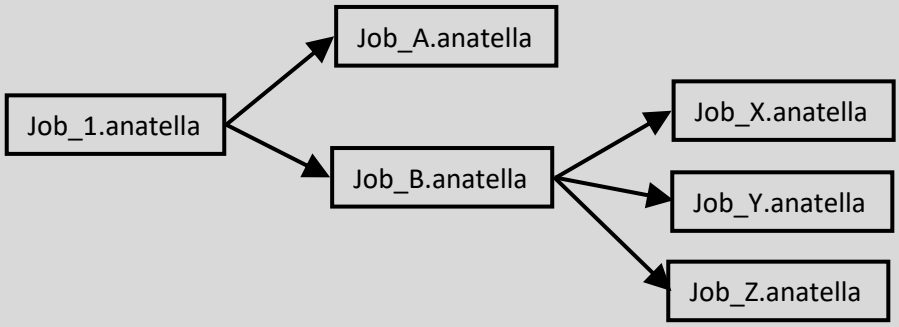
If you saw something interesting inside the Anatella-Log-Window, you can always manually uncheck the "*Quit when script complete*" option inside the "*Commands*" menu:



...this will always prevent the Anatella-Log-Window to close (even if the "Always close" parameter is enabled)

Please note that the (legacy&deprecated) "-te" and "-s" options of the "Anatella.exe" process "propagate down" to all the called subgraphs. To illustrate this propagation, let's look at the following example:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Let's assume that you executed the following command-line from Jenkins:

```
Anatella.exe -e Job_B.anatella -s -te
```

"-te" option: This ouputs inside the "Jenkins log" all the log messages emitted from the following data-transformation-graphs: Job_B.anatella, Job_X.anatella, Job_Y.anatella, Job_Z.anatella.

"-s" option: This executes the following data-transformation-graphs in "Silent" (i.e. non-blocking) mode: Job_B.anatella, Job_X.anatella, Job_Y.anatella, Job_Z.anatella.

A better alternative is to write:

```
AnatellaConsole.exe Job_B.anatella
```

This has the exact same effect.

4.10. Linux Support: Running an Anatella graph in Linux

Currently, under Linux, we only support batch execution of Anatella graphs through the "AnatellaConsole" executable. We do not support the edition of new graphs (using "Anatella"). Typically, you'll design an Anatella graph under windows and put it in production under linux.

To run Anatella under linux, you'll need to install the package named "Wine".
(i.e. Anatella runs inside Wine – Wine is not a Virtual Machine)

The procedure to install "Wine" under "Ubuntu" is the following (This is an extract from https://wiki.winehq.org/Ubuntu):

```
wget -nc https://dl.winehq.org/wine-builds/Release.key
sudo apt-key add Release.key
sudo apt-add-repository https://dl.winehq.org/wine-builds/ubuntu/
sudo apt-get update
sudo apt-get install --install-recommends winehq-stable
```

During the Wine installation, if you get some messages about downloading some additional optional components/packages (e.g. about downloading "Mono" or "Gecko"), you can safely skip these downloads: They are not required to make Anatella run properly (basically, you can safely skip any optional components).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The Anatella installation procedure under Linux is in 4 steps:

1. Download the portable version of Anatella (i.e. download the large ZIP file) and unzip the zip file in a suitable location: Typically, you'll unzip the zip file inside the directory "/opt/timi": i.e. You'll type:

```
cd /opt
sudo mkdir timi
cd timi
sudo unzip /home/<user>/TIMiPortableFull_x64.zip
```

2. Setup "Wine". "Wine" emits many warning messages inside the console. To get rid of these (annoying) warning messages, add the following line inside your "~/.bashrc" file:

```
export WINEDEBUG=-all
```

To be able to launch the "AnatellaConsole" executable in an easier way, you can add the following line inside the (same) "~/.bashrc" file:

```
alias anatellaconsole='WINEDEBUG=-all wine /opt/timi/bin/AnatellaConsole.exe'
```

To edit the "~/.bashrc" file, I like to use a very simple text editor named "Joe". To install "Joe":

```
sudo apt-get install joe
```

Then, to edit the "~/.bashrc" file:

```
cd
joe .bashrc
<Scroll to the end of the file and add the required lines>
<Press [CTRL]+[K]+[X] to save your modifications inside the
".bashrc" file and exit>
<Press [CTRL]+[C] to cancel, ignore your modifications and exit>
```

3. To enter your Anatella "serial number", run "TIMiEnterLicenceConsole.exe" inside a console: i.e. Typically, you'll type:

```
cd /opt/timi/bin/
wine TIMiEnterLicenceConsole.exe <registration_name> <serial_number>
```

(Type "`wine TIMiEnterLicenceConsole.exe`" without any parameters to get help).

It's possible to run both the "`AnatellaConsole`" executable (to run Anatella graphs in batch) and the "`Anatella`" executable (to edit or create new Anatella graphs) but we only support running Anatella in batch in Linux through the "`AnatellaConsole`" executable.

Once the "`anatellaconsole`" alias is created, it's very easy to run Anatella graphs: For example, to run the "`MyGraph.anatella`" graph, you just type:

```
anatellaconsole MyGraph
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

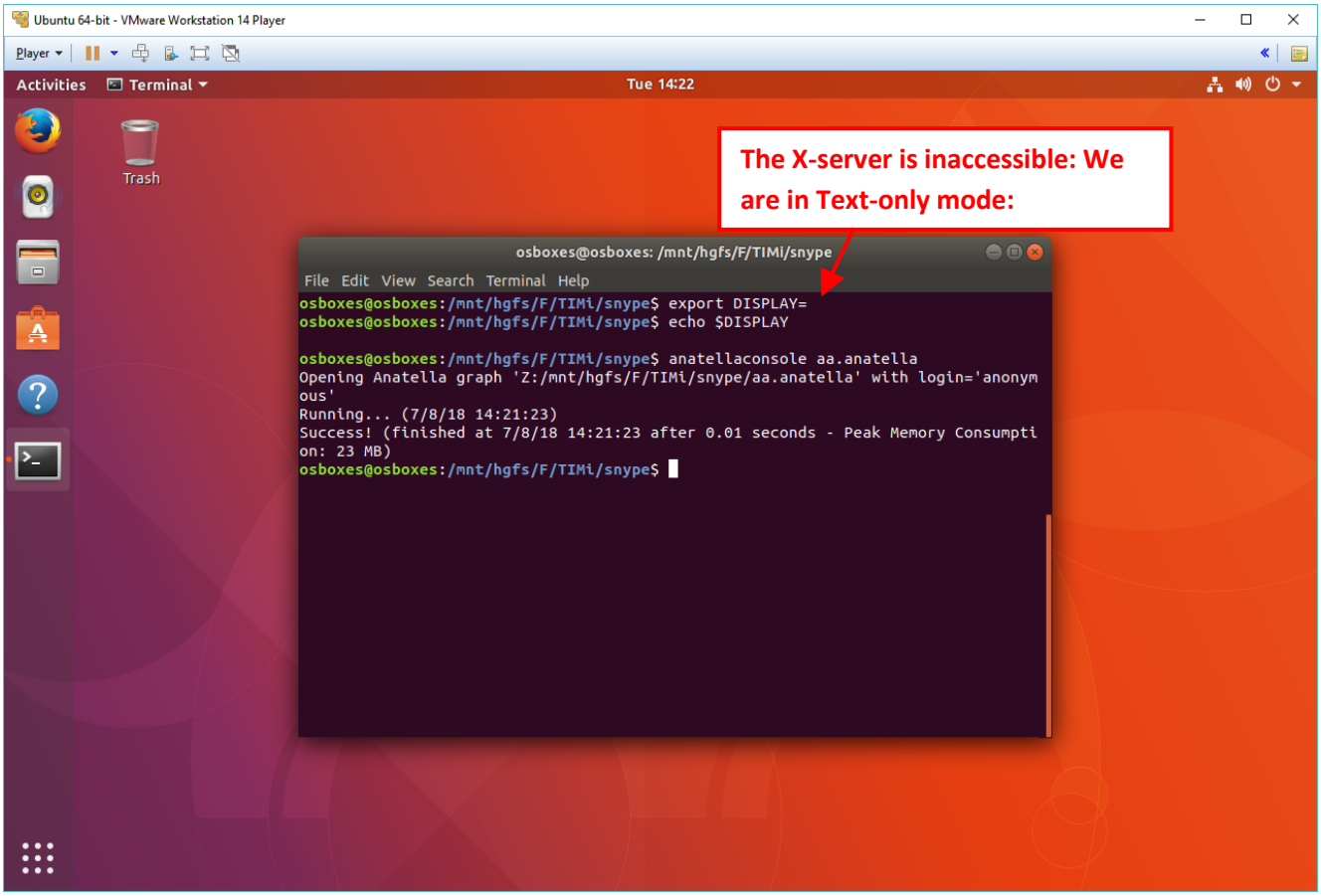
The command-line options are the same as usual: For example, here we re-define the global parameter named "`myGlobalParameter`" to the value "`xxxx`":

```
anatellaconsole MyGraph -DmyGlobalParameter=xxxx
```

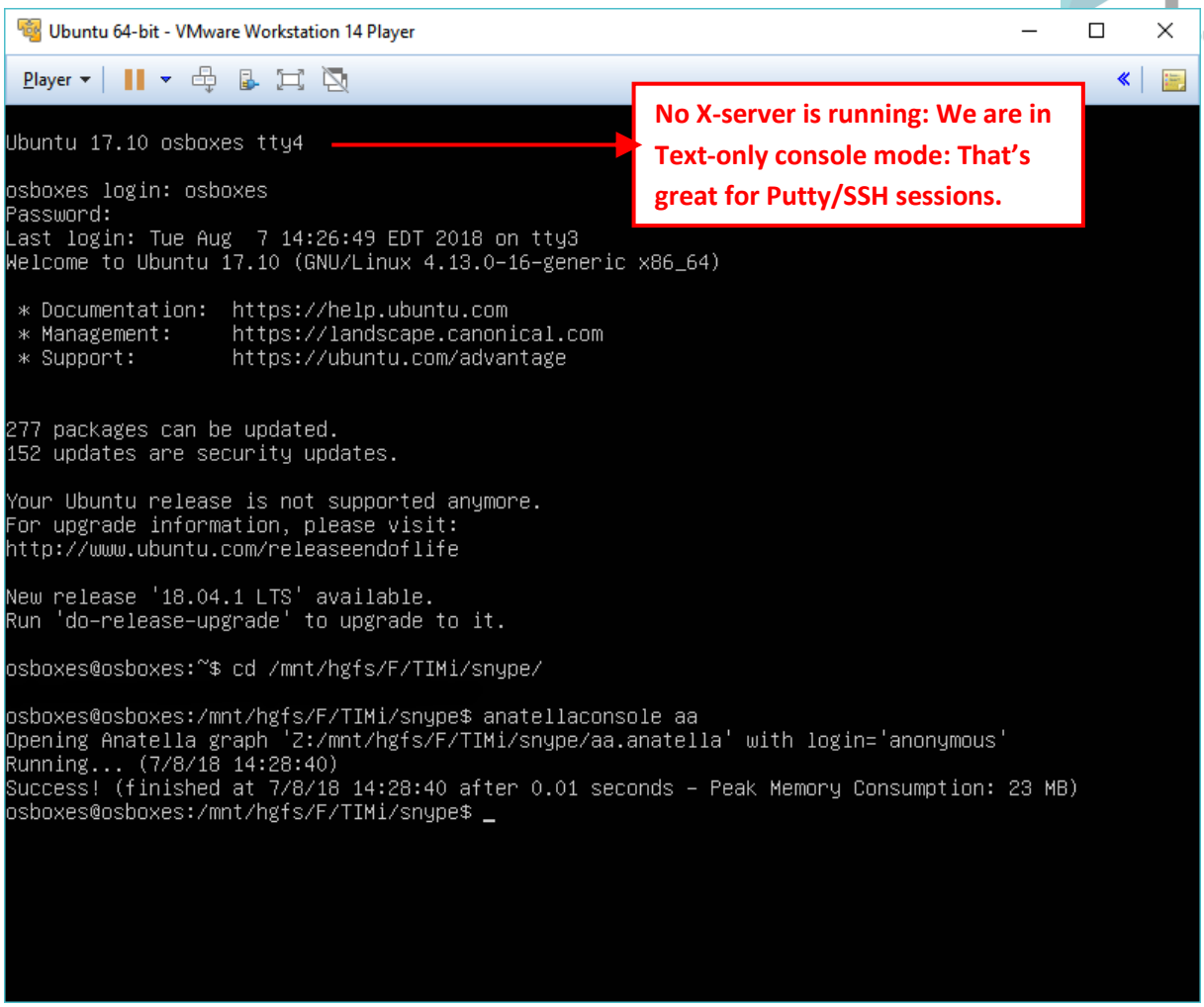
Since the "`AnatellaConsole`" executable is a pure "console-based" executable, you can run it directly inside a "Putty/SSH" console (No X-Server is required! No graphical GUI is required!). This is very handy when working on a far remote Linux server that has a very low connection bandwidth (so low actually, that it won't support any type of Graphical User Interface). Typically, in such situation, you'll edit your .anatella files locally (on your windows PC – and since all the edition process is done locally, everything is fast and responsive), then you push your .anatella files on the remote linux server (e.g. using "`WinSCP`" – since the .anatella files are very small files this is not time-consumming to transfer them), and finally run your .anatella files on the remote Linux server by typing inside a Putty/SSH console something like:

```
anatellaconsole MyGraph
```

Here are two example screenshots (Ubuntu 17.10 in 64 bit):



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Some Anatella Actions require, in order to run, a working Graphical environment (i.e. the text console is not enough to make them run). Such that, it won't be possible to execute these Actions inside a pure Putty/SSH text console.

The Anatella Actions that requires a Graphical environment are the Actions included inside the "R Visulation" category. Indeed, the objective of these "visualization" Actions is to display some bart chart, scatter plot, etc. inside a Graphical Window and they, of course, won't work properly without a working Graphical environment.

You can call any Linux tool from Wine/Anatella using the "`start /unix`" command. For example, the Javascript function "*runLinux()*" defined here below runs any linux command that is given as parameter:

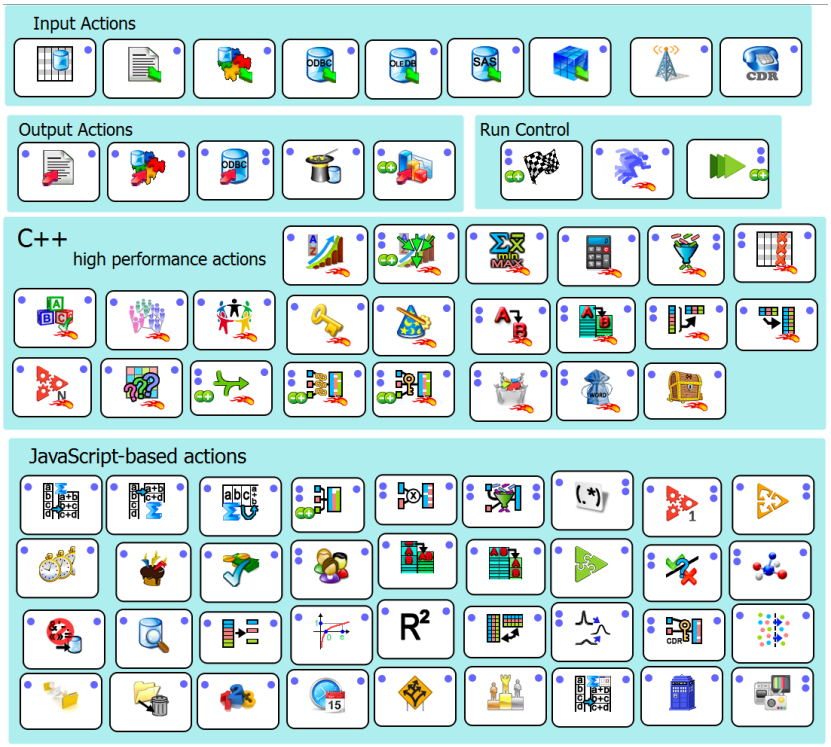
```
function runLinux(s)
{
  if (!isLinux()) throw "Error: You are not using Linux!";
  var p=ProcessRunner();
  p.setVerbose(0);
  return p.runImmediate("cmd",["/c","start","/unix","/bin/bash","-c",s]);
}
```

Traditionnaly, a return value of "zero" means no error.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5. Detailed description of the Actions

Some of the currently available Actions are:

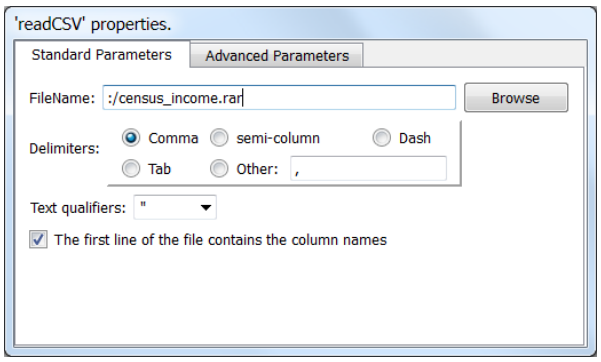


5.1. General Parameters used in many actions

5.1.1. Filenames

Many actions (e.g. the “CSV File reader/writer”, “Excel File reader”, “Anatella Gel File reader/writer”, etc.) require you to give a filename as parameter.

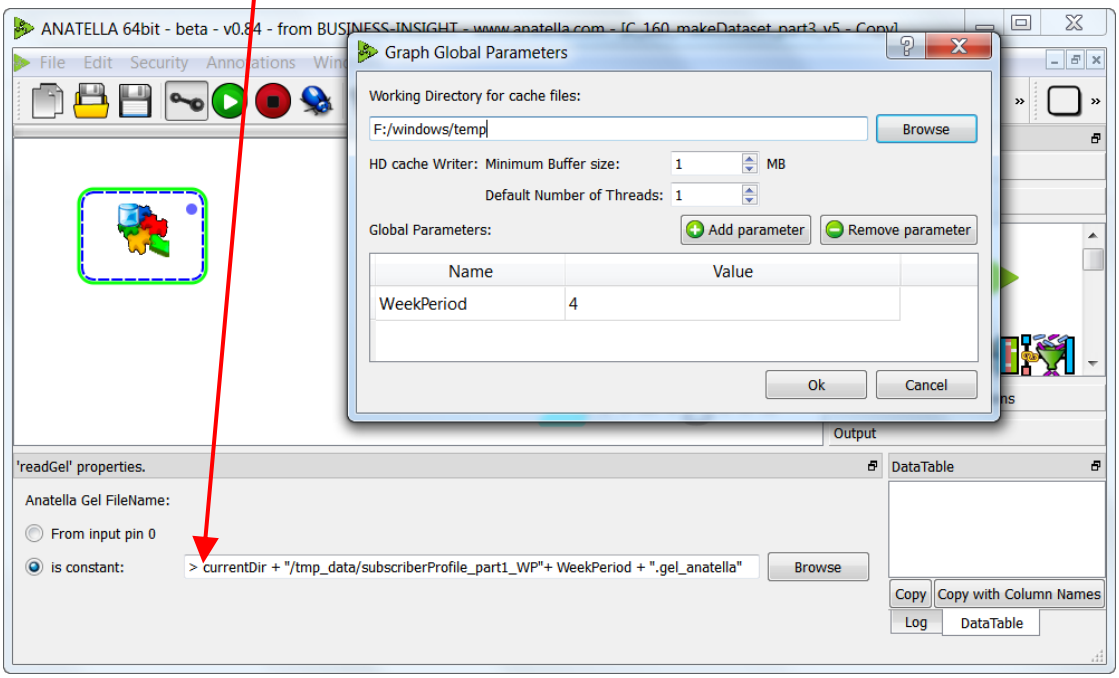
Inside Anatella, you can specify a path to a file in different ways. For example, if the Text/CSV file that you want to read is inside the same directory (or inside a subdirectory) as your currently edited Anatella-Graph file, then Anatella will automatically rewrites your path in a “relative way” (The re-written path is relative to the location of your current Anatella Graph File): You will see a ‘:’ character as the first character of your path: For example:



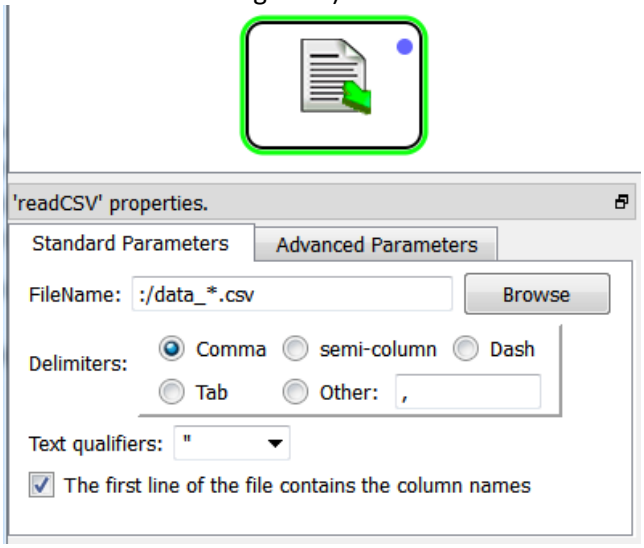
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In this example, the file “census-income.rar” is located inside the same directory as the Anatella-Graph file. Thanks to this functionality, the location of your Txt/CSV file is now relative to the location of your Anatella-Script. This allows you to easily copy/move your Script&Data into another directory. This functionality is interesting when you want to share with your colleagues your Anatella-Graphs because you never know in which directory your colleague will copy your Script&Data and it should still work properly.

Another way to specify the path to a file is to use a small bit of JavaScript code to compute the path. When you set the character “>” as the first character of the filename, the “filename” is computed using JavaScript code. For example: This will open the file “*subscriberProfile_part1_WP4.gel_anatella*” that is located inside the directory “tmp_data” (The directory “tmp_data” is, itself, a direct subdirectory of the directory containing the Anatella-Graph-File):



Another way to specify the path to a file is to use the wildcard character (i.e.: The “*”character)(Wildcard characters are only accepted in the filename, not in the directory-path). All the files matching the wildcard filter will be opened one after each other and their content will be concatenated together in one unique large output table. For example: This will open all the .csv files starting with “data_” inside the current directory (i.e. the directory from the Anatella-Graph-File). The content of all these files will be merged together (this is equivalent to using the “Append” action to concatenate all these .csv files together):



When using the wildcard character (i.e.: "*") inside a "readCSV" action, all the .csv files must contain exactly the same columns, in exactly the same order (not one more, not one less).

5.1.2. Data Types

There are 3 basic data-types available inside Anatella:

1. Unknown type (or String type)
2. Float Type (also named "double")
3. Key type

The dates (or the times) are a special case: They can be represented using either the Unknown/String type or the Key type.

Use the ChangeDataType Action to convert from one data-type to another.

Each different data-type has specific properties:

Data-Type	Unknow type (or String type)		Float Type (double)		Key type	
Memory consumption for the storage of one cell/value in RAM.	String Size = s	bytes	8 bytes		4 bytes	
	NULL value	1				
	s < 128 chars	1 + 2 s				
	s < 16384 chars	2 + 2 s				
	s < 2097152 chars	3 + 2 s				
	otherwise	4 + 2 s				
Space consumption for the storage on the hard drive.	File format:	Compression	File format:	Compression	File format:	Compression
	.gel_anatella	★★★	.gel_anatella	★★	.gel_anatella	½
	.cgel_anatella	★★★½	.cgel_anatella	★★½	.cgel_anatella	★★★★★★★
Admissible values.	NULL value, empty string, any string		NULL Value, NaN (Not A Number), +Inf, -Inf, any number		NULL Value, Any positive integer number less than 4,294,967,294	
Computation Speed (Efficiency in manipulating variable of this type).	Moderate		High		Very high	

In general, the most efficient data-type for storage is the "Key" Type because:

1. One "Key" value consumes only 4 bytes of RAM memory for storage.
2. One "Key" value uses very little space on the hard drive (especially when using the ".cgel_anatella" file format: The compression is very efficient in this case)

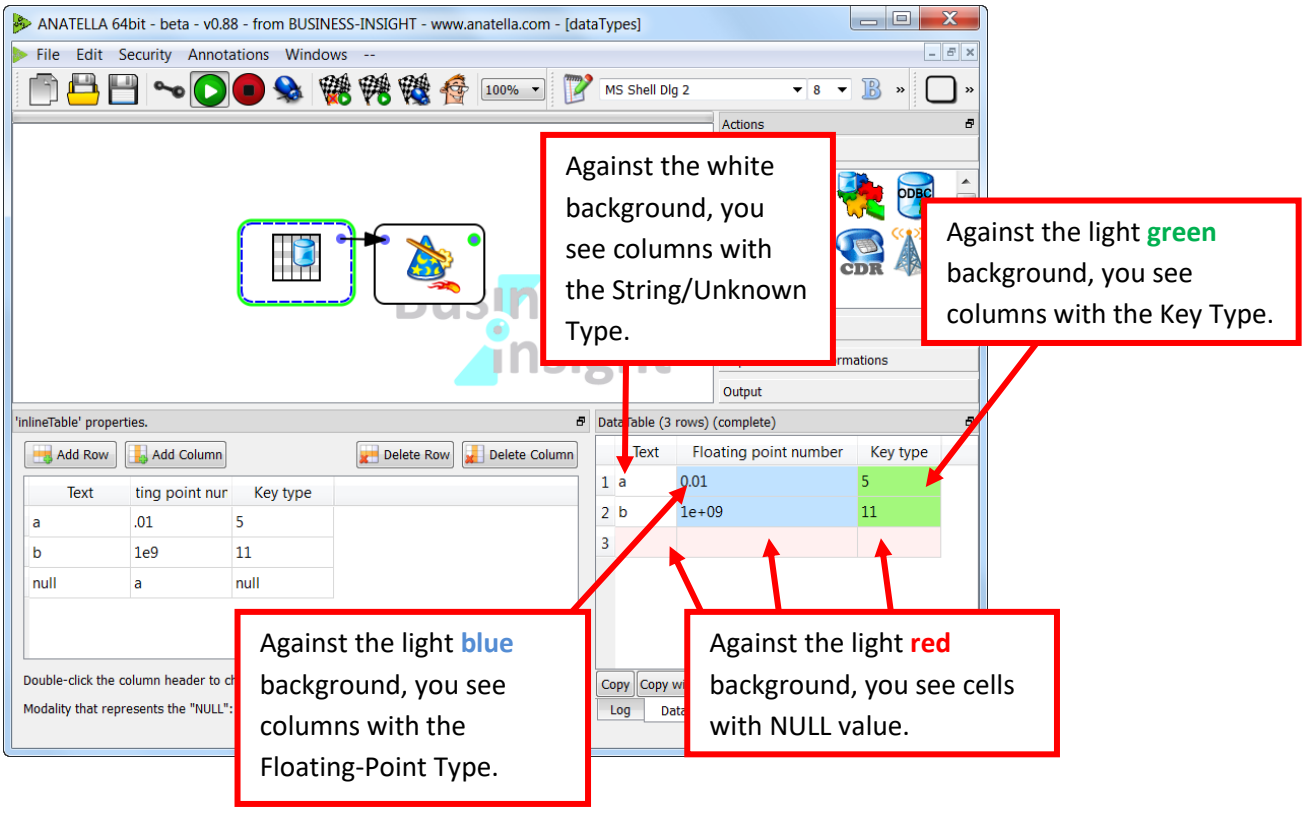
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

A one character-String consumes only 3 bytes and is thus a little bit more efficient, in terms of RAM memory consumption, than the "Key type" (that uses 4 bytes). Please don't forget that, if you intend to make many number computations based on a 1 character-String, the transformation from String to float can be CPU intensive (especially for large tables). Furthermore, the "Key" type compresses *a lot* better than the "Unknown/String" type, so the consumed hard-drive space is very much lower when using the "Key" type. To summarize: Even for numbers with only one digit, it's nearly always better to use the "Key" type.

Using a more efficient Data-type will allow you to:

1. Consume less disk-space for your ".gel_Anatella" or ".cgel_anatella" files.
2. Increase processing speed in many actions: e.g. join operations (with the "MultiJoin" Action), Variable creation (with the "Calculator" Action), Filtering (with the "FilterRows" Action).
3. Handle larger Slave Tables in the "MultiJoin" Action. Indeed the "MultiJoin" Action starts by loading into the central RAM memory all the slave tables. If these tables consume less memory space (because of a more efficient data-type), you'll be able to compute MultiJoins on slave tables containing a larger number of rows.

You can easily see the data-type of each column in the Data-preview window:



When a cell is empty, it can contain either an empty string or a NULL value. To know the exact content of the cell, look at the background's color: If the background's color is **red**, then the cell contains the NULL value (otherwise the cell contains the empty string "").

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.3. Dates

Dates fields are typically received in input of an Anatella Data Transformation graphs as simple Strings.

You can convert Dates-stored-as-String to Date-stored-as-Key using the ChangeDataType Action.

All computations on Dates-stored-as-Key are a lot faster than on Dates-stored-as-String. The memory consumption of Dates-stored-as-Key is also significatively smaller (and thus better). When working on large data volume, it's strongly suggested to rather use Date-stored-as-Key, to have the best speed and the lowest hard drive & RAM memory consumption.

There exist many Actions in Anatella that can directly use Dates-stored-as-Strings:

(sort) (dateTimeFormatter) (transformToAge)

(valueBackInTime) (customerAggregate) (ChangeDataType)

Inside these actions, we'll need to provide the Date Format, so that Anatella can interpret the date correctly.

A date format is composed of two parts:

1. The first part specifies the date of the day (for example: March 24th, 2010).
2. The second part specifies the time in the day (for example: noon: 12:00)

For the first part of the "date format", these expressions may be used:

Expression	Output
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. 'Mon' to 'Sun').
dddd	the long localized day name (e.g. 'Monday' to 'Sunday').
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. 'Jan' to 'Dec').
MMMM	the long localized month name (e.g. 'January' to 'December').
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

About Localized day name and Localized month name
Day and Month names must be given in the user's local language. It is only possible to use the English names if the user's language on the computer is English.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For the second part of the “date format”, these expressions may be used:

Expression	Output
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
H	the hour without a leading zero (0 to 23, even with AM/PM display)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without leading zeroes (0 to 999)
zzz	the milliseconds with leading zeroes (000 to 999)
AP or A	interpret as an AM/PM time. *AP* must be either "AM" or "PM".
ap or a	Interpret as an AM/PM time. *ap* must be either "am" or "pm".

For any field that is not represented in the format the following defaults are used:

Field	Default value
Year	1900
Month	1 (January)
Day	1
Hour	0
Minute	0
Second	0

The expressions that don't have leading zeroes (d, M, h, m, s, z) will be greedy. This means that they will use two digits even if this will put them outside the range and/or leave too few digits for other sections.

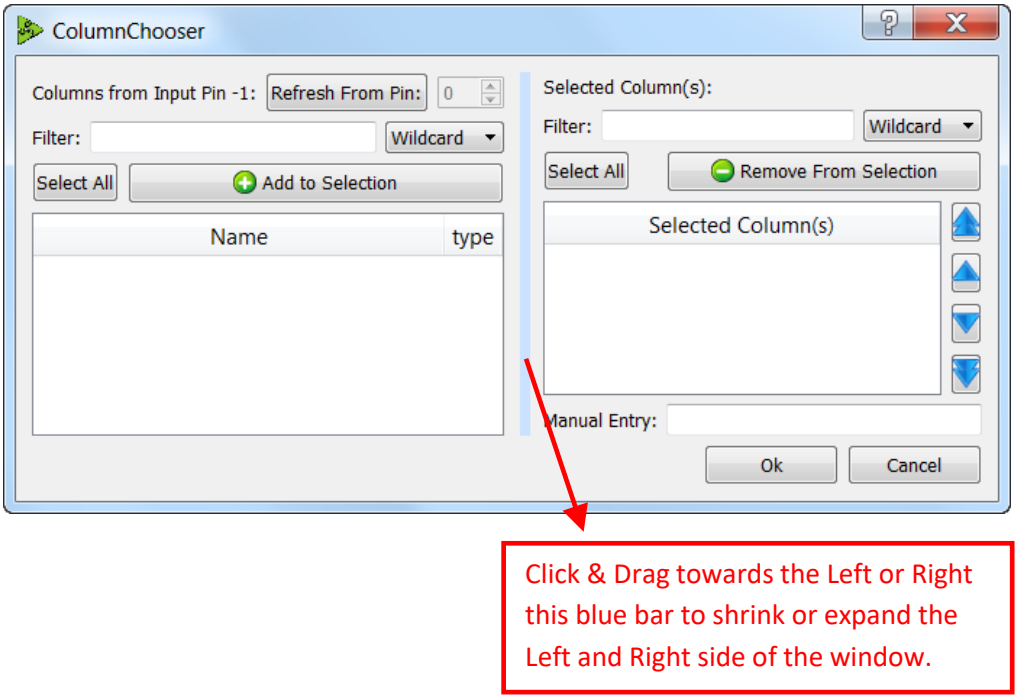
Some examples:

Content of data table	Date format	value
131	HHh	13:00:00
1apA	1amAM	01:00:00
M1d1y9800:01:02	'M'M'd'd'y'yyhh:mm:ss	1 January 1998 at 00:01:02
130	Mm	<invalid>
1.30.1	M.d.s.	30 January 1900 at 00:00:01

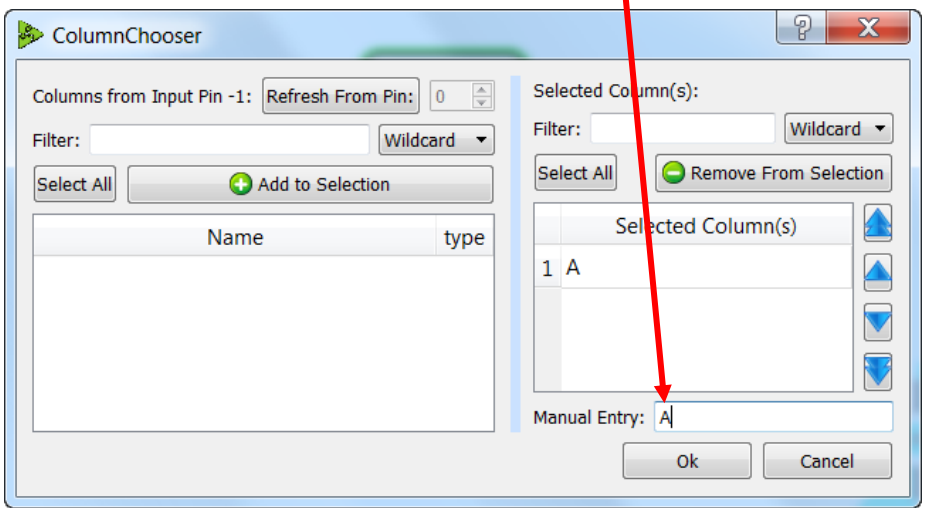
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.4. Column Chooser

Nearly all Anatella Action requires you to select, at some point, some column(s) of the input table(s). Column Selection is performed using a standard window named the "Column Chooser":

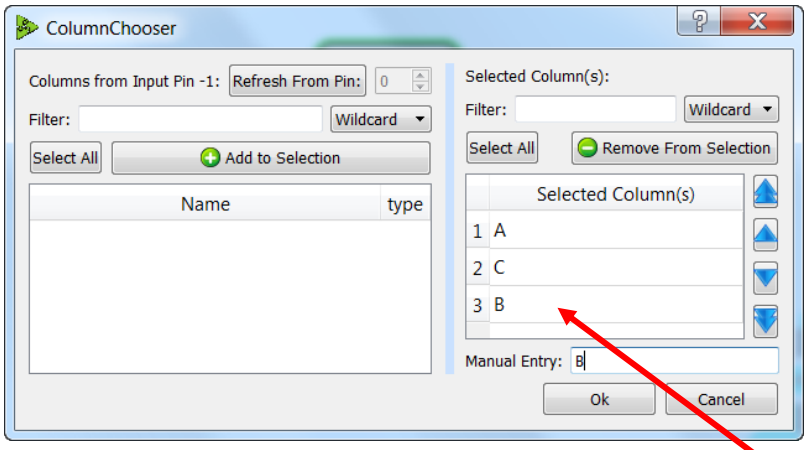


For example, the "Column Chooser" window appears when you want to Sort a table: To sort a table, use the "Sort" action and click on the "*Add Column on which to sort*" button inside the property window of this action: The "Column Chooser" window appears to help you specify on which column(s) to sort your table. Let's say that you want to sort your table on the "A" column: Simply write "A" inside the "Manual Entry" text box and press [return]:

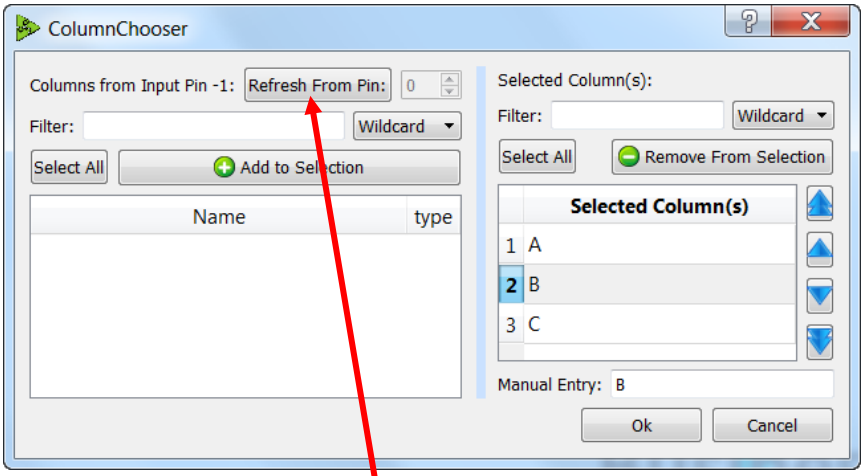


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can add other columns on which to sort: For example, let's say that you want to sort your table on the "A", "C" and "B" columns:



The order in which the columns appear inside the Selection is important. Indeed, sorting on columns "A", "C" and "B" is different from sorting on columns "A", "B" and "C". You can re-order the Selected columns using the buttons. For example, click on the "B" column and thereafter on the "up" button: You obtain:



Entering manually the column names is the fastest option when you have a limited number of easy names to specify. When you need to select a large quantity of columns with very complex names, you should use another, "assisted" technique: More precisely:

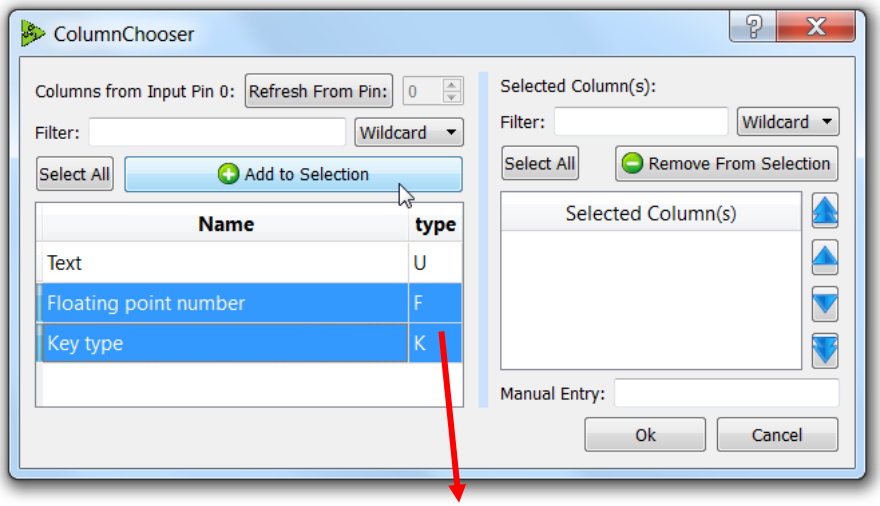
1. Click on the "*Refresh From Pin*" button. Anatella will run the data transformation graph to compute the column names of the input table and display these column names inside the List on the left side of the Column Chooser Window.

   Please note that obtaining this Column List implies running the data transformation graph. This is an operation that can potentially take a large amount of time (or it can also simply fail). To reduce computation time, you can:

   a. use the Hard Drive cache mechanism to pre-compute the input table.
   b. enter manually the column names.
   c. copy-paste a column-name-list using CTRL-C (to copy) and CTRL-V (to paste).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

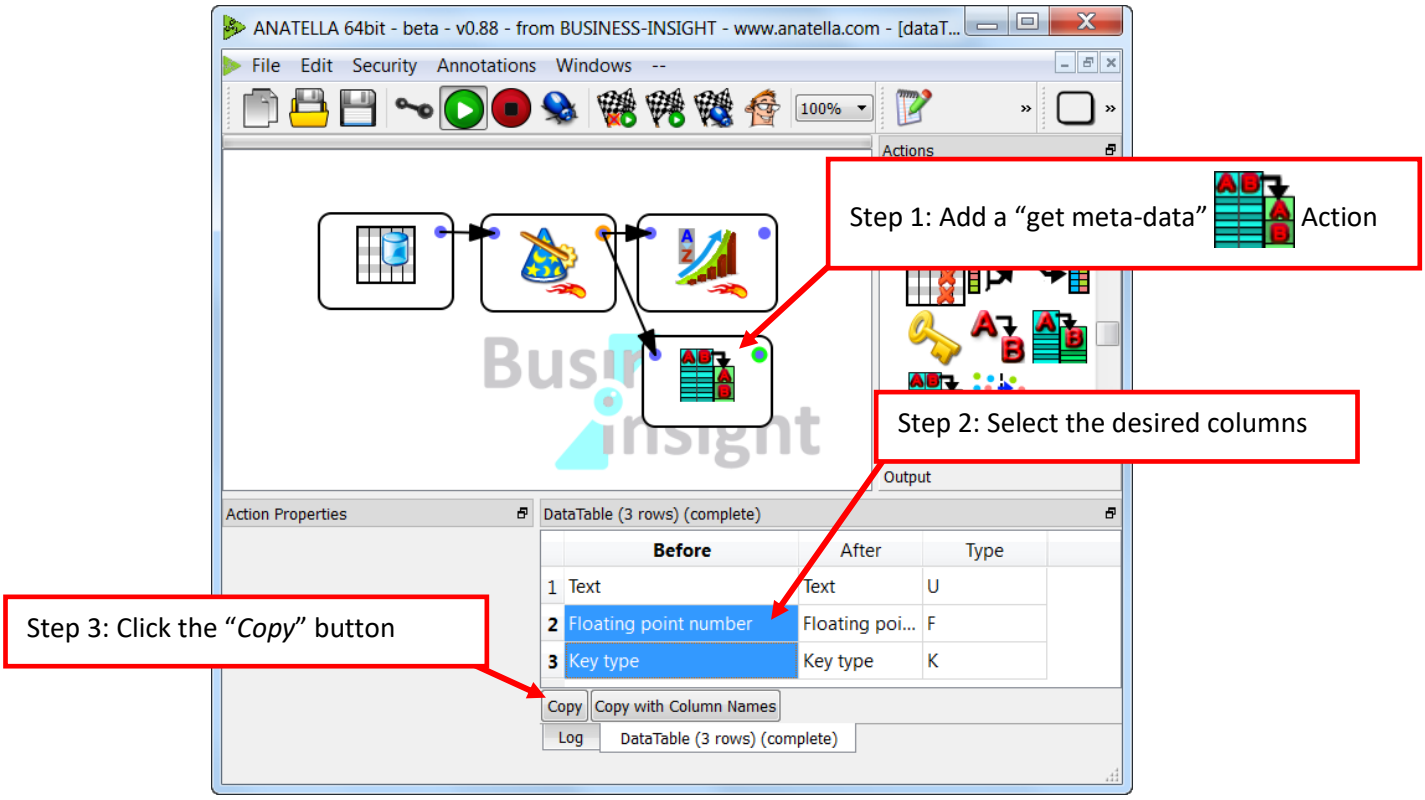
2. You now need to copy some column from the Left Side to the Right Side of the window. To do this:
    a. Double-click a column name: it will be instantaneously added to the list of Selected Column on the right side.
    b. Select some column in the right-side list and click the "*Add to Selection*" button. For example, this will add the columns named "Floating point number" and "Key type" to the list of Selected Columns:



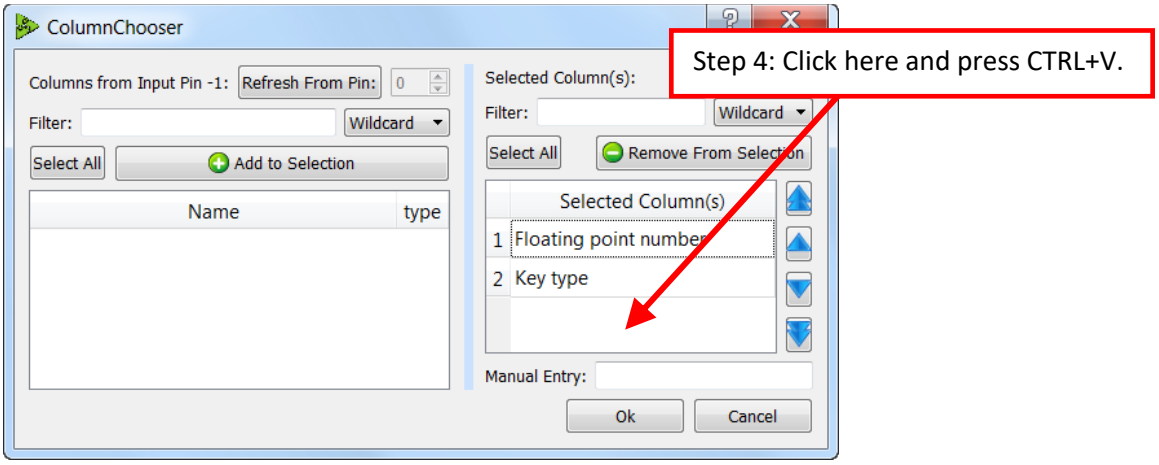
The Data type of the input columns is visible here:
U means Unknown (or String),
F means Floating-Point,
K means Key.
For more information about column data types, see section 5.1.2.

Another very fast way of creating a selection of Column is to use Copy-Paste. Inside Anatella, you can copy from nearly all visible tables by selecting some rows of the table and pressing CTRL+C. For example, you can use the "get meta-data" Action to get a list of column inside the Data Preview Window. Select the desired columns in the Data Preview Window and click the "copy" button (or press CTRL+C): Here is an illustration:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Thereafter, you can paste your column selection: Open the Column Chooser window, click (i.e. select) the right-side list, press CTRL+V:



Anatella is designed to work with extremely large (wide) tables. This means that, when you click the Refresh From Pin: button, the list of columns displayed on the left pane of the "Column-Chooser window" can contain 30.000 different column names without any problem. When you have such a large amount of columns, it's not always easy to find the right one(s). This is why Anatella offers you several tools to find the columns that you are searching for:

- You can filter the list of columns using:
  - A wild card filter notation (for example: "rev*" will give you all the columns that start by "rev").
  - A regular expression.
  - A fixed string

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- When you click on the table header, you sort the column names in alphabetical order (if you click a second time, the column names are sorted in inverse alphabetical order. A third click cancels the sort).

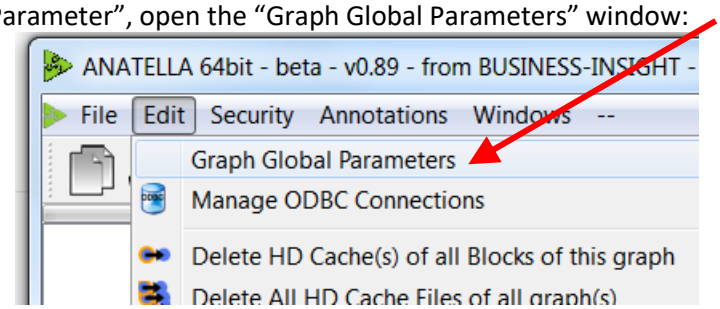
5.1.5. Graph Global Parameters

The "Graph Global Parameters" are variables that are accessible from any Action anywhere inside your transformation graph. You can use "Global Parameters" inside:

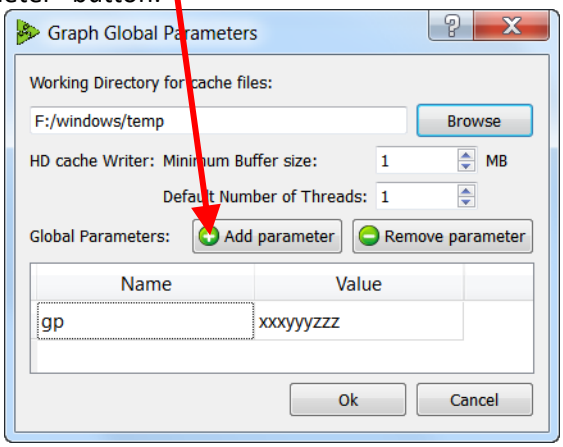
1. Any Javascript-based Action
2. Any filename defined using a JavaScript code (see section 5.1.1.)
3. Any SQL statement defined using a JavaScript code (see sections 5.2.2. and 5.2.3.)
4. Any "Calculator" or "FilterRow" Actions (even if these Actions are not Javascript-based Actions)

"Graph Global Parameter" are extensively used when creating and running sub-graphs (e.g. in a loop). Typically, the parameters required for the proper execution of the sub-graphs are passed from the "master graph" (i.e. the "calling" graph) to the "Sub-Graphs" (i.e. the "called" graphs) using "Graph Global Parameter": See the section 5.3.3. about calling sub-graphs.

To create a "Global Parameter", open the "Graph Global Parameters" window:



… and click the "Add Parameter" button:



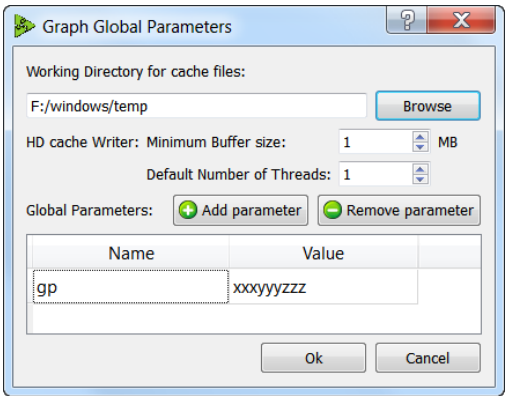
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can change the value of existing "Global Parameters" using:

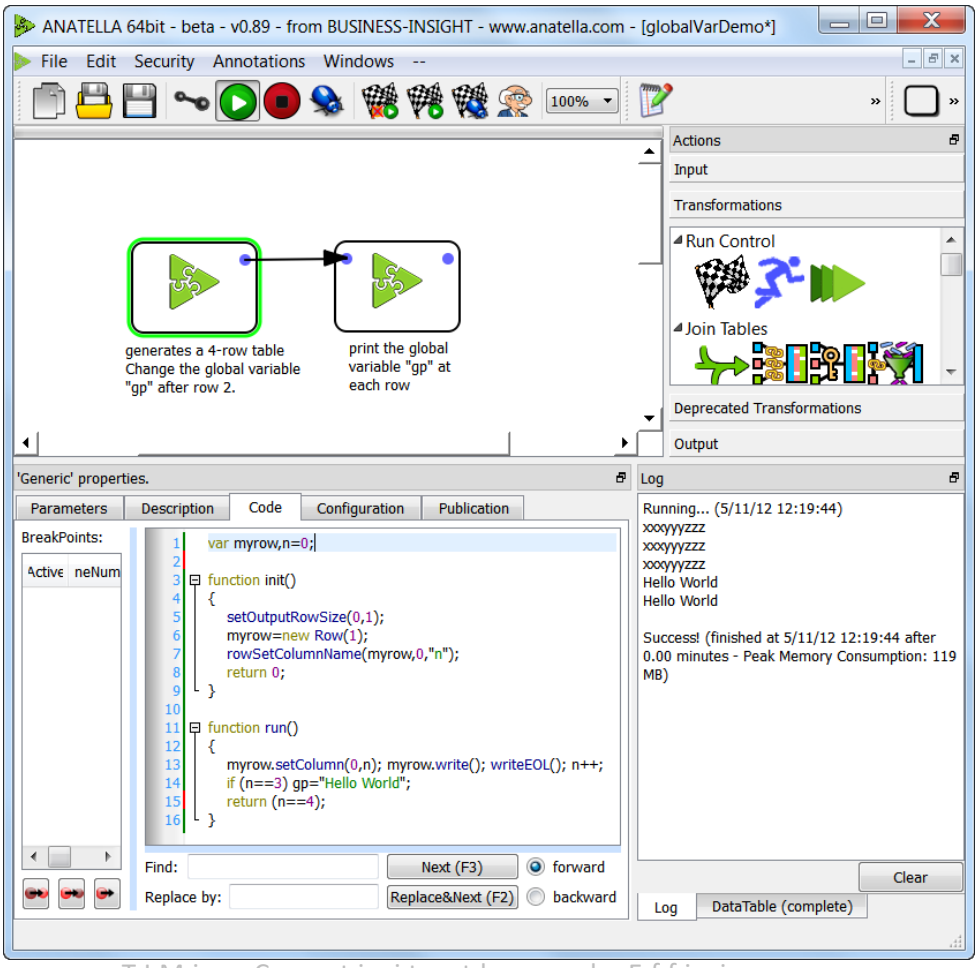
1. **Command-line options:** (see section 4.7.1). That's the most common way of defining the "Global Parameters" values. This allows you to easily "parameterize" your Anatella-Data-Transformation-Graph.
2. **Javascript code:** This allows you to "communicate" between different Actions. The Javascript codes executed in different Actions are isolated from each other (to prevent any interference between different Actions). One easy way to "send" some (small amount) of data between different Actions is to use Global Parameters.

   Here is a small example:

   - We define a new Global Variable named "GP":

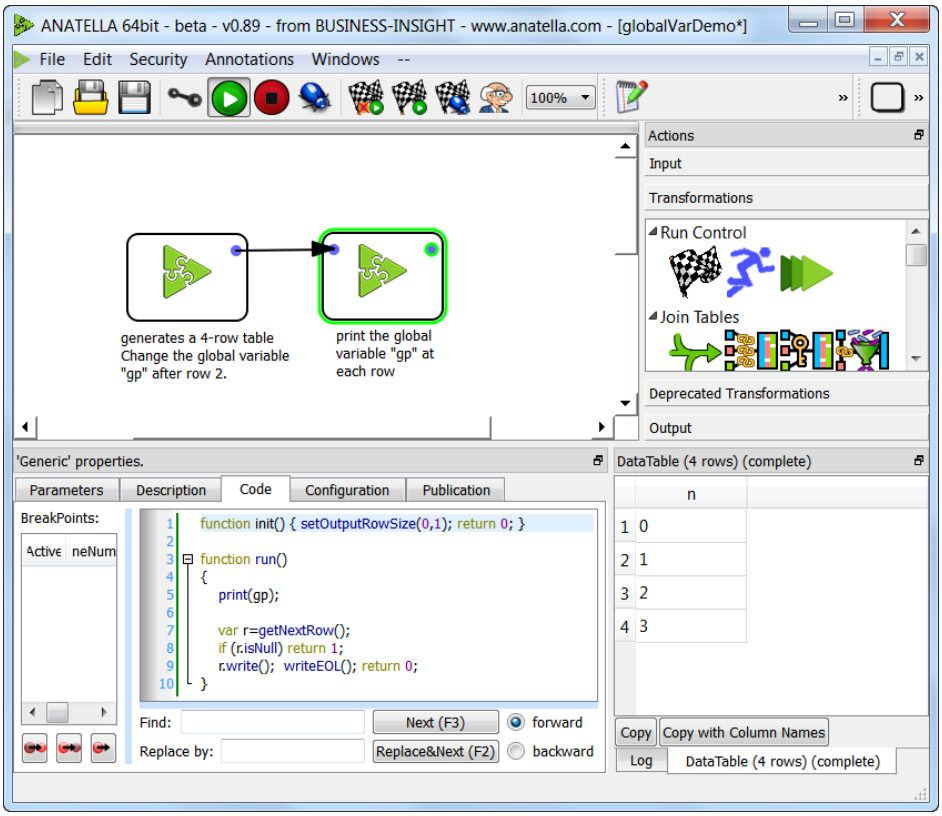


   - The transformation-graph is composed of 2 Actions. The first Action generates a table with 4 rows and changes the value of the "GP" variables after generating 3 rows:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- The second Action simply prints inside the log window the value of the "GP" variables as observed at each row. In the above screenshot, you can see that the value of the "GP" variable changed after three "prints". Below is the JavaScript code of the second Action:



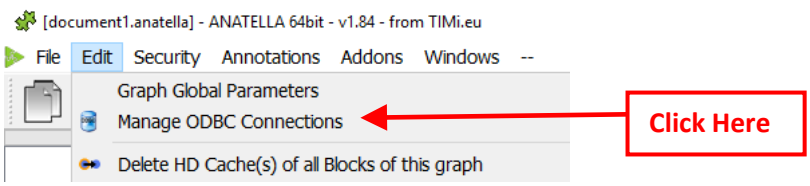
5.1.6. ODBC Connections

ODBC is a generic technique to access the content of almost any relational databases.
ODBC is the oldest technique available to access data contained into relational databases.
All the databases have an ODBC driver.

The readODBC action (see section 5.2.3), the upsertODBC action (see section 5.26.4), the CreateTable action (see section 5.26.5) and the TeradataWriter action (see section 5.26.19) all require an ODBC connection to work.

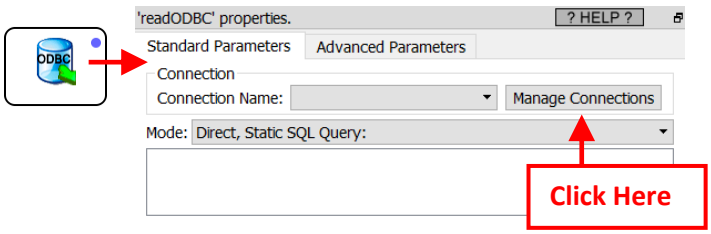
To setup a new ODBC connection for use inside Anatella, you go to the "*Manage OBC Connections*" window. You can access this window in two different ways:

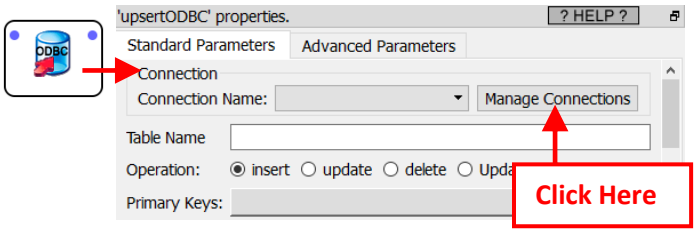
- Inside the drop-down menu "Edit", click the "Manage OBC Connections" option:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

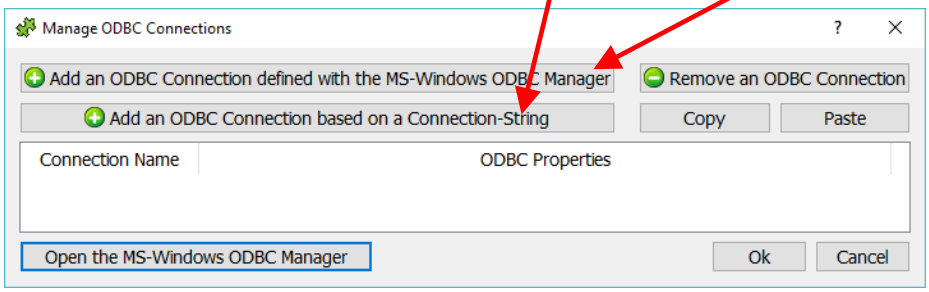
- Click the "*Manage Connections*" button inside the properties of the readODBC action or the upsertODBC action:





Inside the "*Manage OBC Connections*" window, you can add&define two different types of ODBC connections:

- **Type 1:** An ODBC connection that was defined with the MS-Windows ODBC Manager: Click here;
- **Type 2:** An ODBC connection based on a connection string: Click here;



Each ODBC connection type ("Type 1" or "Type 2") has its PRO's and CON's:

Topic	Type 1	Type 2	Comments
Compatibility	**Always supported**	**Only supported by a limited number of ODBC drivers**	"Type 2" is supported by Oracle, MS-SQL Server, Teradata, DB2. "Type 2" is **un**supported by PostgreSQL, MySQL.
Administrative rights	**Required**	**Not required**	Administrative rights are required to create&edit the "ODBC DSN" inside the MS-Windows ODBC manager (not to use it).
Documentation	**Extensive**	**Partial**	Very often, the parameters of the "Type2" ODBC connections are not well documented by the database vendors. In opposition, an easy "wizard" based interface is usually provided to create "Type1" ODBC connections (inside the MS-ODBC Manager).
Deployment	**Complex**	**Easy**	The "Type2" ODBC connections are easier to deploy because once the connection-string to your database is working, it's very easy to copy-paste it everywhere you need connectivity. In opposition, creating a "Type1" ODBC connection requires many clicks and editing operations inside the MS-ODBC Manager. Furthermore, only the people with administrative rights can do these editing operations inside the MS-ODBC Manager.
Robustness	**Low**	**High**	Once a "Type2" ODBC connections is setup, it will continue to work, almost indefinitively. A "Type1" ODBC connection requires the presence of a matching "ODBC DSN" inside the MS-Windows ODBC manager. If this "ODBC DSN" disappear from the MS-Windows ODBC manager (e.g. your administrator removed it), your database connection is broken.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

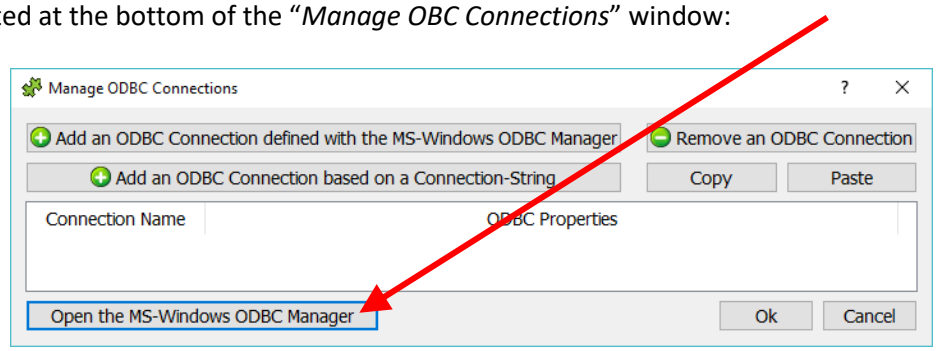
In the next 2 sections (5.1.6.1 and 5.1.6.2.), we'll review the procedure step-by-step to create each ODBC connection type ("Type 1" or "Type 2"). The sections from 5.1.6.3 to 5.1.6.6. give more explanations on the detailed setup of the ODBC drivers of each database.

5.1.6.1. Type 1 ODBC connection: Create&Use an ODBC connection that was defined using the MS-Windows ODBC Manager

Let's give a small example of usage. The first operation that you need to do when working with a "Type 1" ODBC datasource is to use the the "*Microsoft ODBC Administrator*" to create a "link" (also named an "ODBC DSN") toward the database that you want to use.

In the below example, we'll create a "link" to access the content of a "MySQL database". The technique is the same for all the databases (Oracle, Teradata, SQL Server, My SQL, DB2, etc.). You'll find additional information for the proper setup of an ODBC connection to an Oracle, Teradata, MySQL& Sybase database in the sections 5.1.6.3. to 5.1.6.6.

The easiest way to run the "*Microsoft ODBC Administrator*" is to click the "*Open the MSWindow ODBC Manager*" button located at the bottom of the "*Manage OBC Connections*" window:



When using a 64-bit Windows OS, there is a little "twist" related to the "Microsoft ODBC Administrator": There are actually 2 completely different "Microsoft ODBC Administrator" software: one for 32 bit applications and one for 64 bit applications. To add more confusion, these 2 different "Microsoft ODBC Administrator" software looks (nearly) exactly the same.

If you are using Anatella 64-bit, you must use the 64-bit "Microsoft ODBC Administrator" software to define your ODBC DSN.

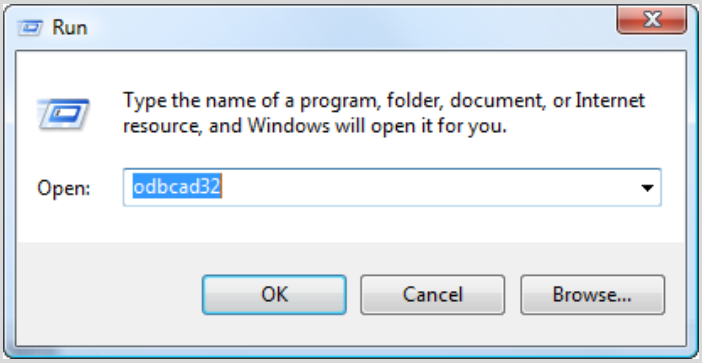
If you are using Anatella 32-bit, you must use the 32-bit "Microsoft ODBC Administrator" software to define your ODBC DSN.

Don't worry: When you click the "*Open the MSWindow ODBC Manager*" button inside Anatella, the correct version of the "Microsoft ODBC Administrator" software appears automatically and there are no risks of errors.

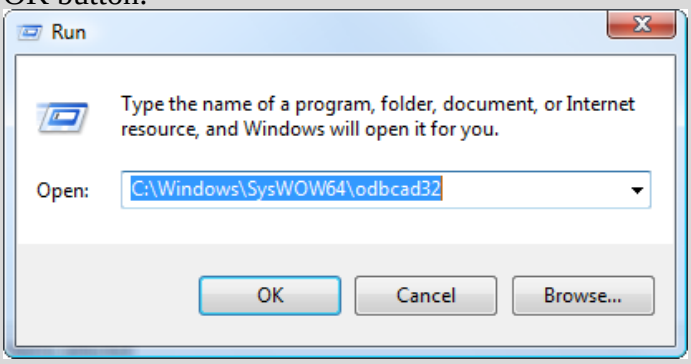
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is another procedure to run the "*Microsoft ODBC Administrator*": Press the [Windows]+[R] keys simultaneously (the "Run" windows pops up) and then:

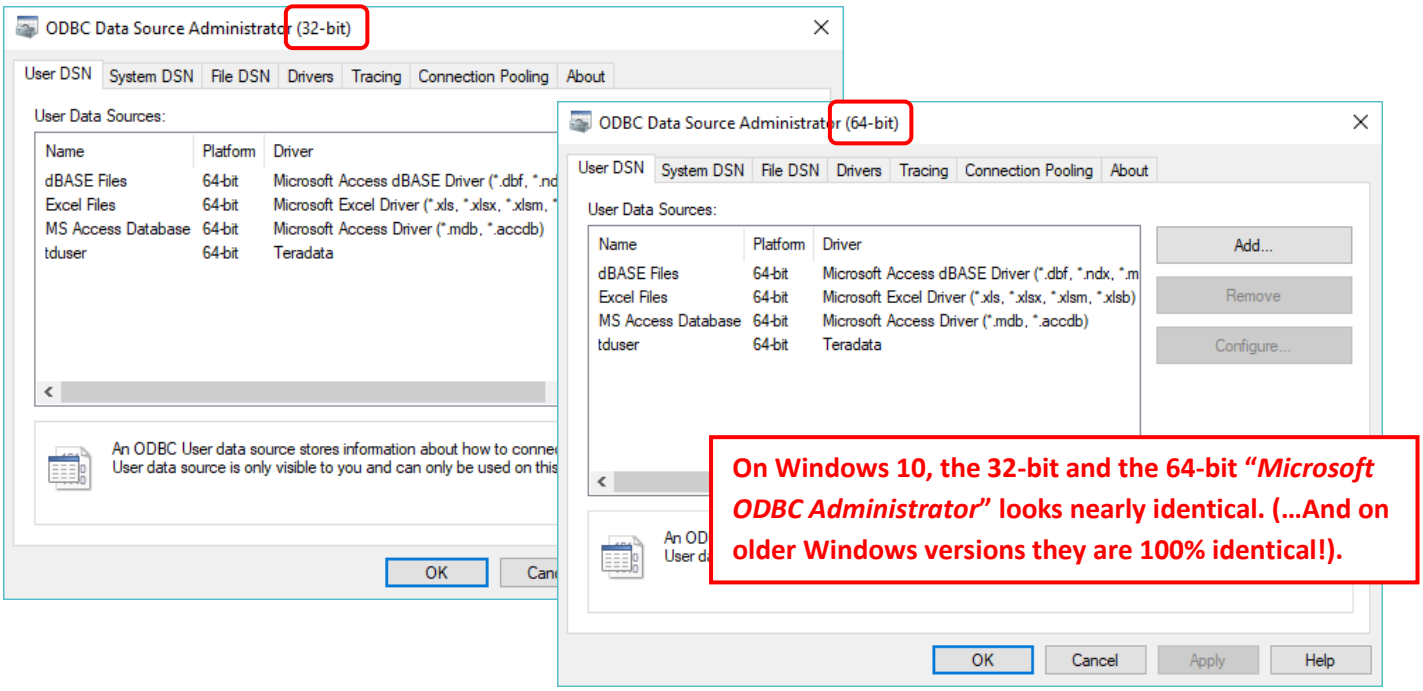
* To run the 64-bit "Microsoft ODBC Administrator" software, write "odbcad32" inside the textbox and click the OK button:



* To run the 32-bit "Microsoft ODBC Administrator" software (on a 64-bit Windows System), write "C:\Windows\SysWOW64\odbcad32" into the textbox and click the OK button:



We should now see the standard "*Microsoft ODBC Administrator*" window (This is nearly the same window for the 32-bit version and the 64-bit version of the "*Microsoft ODBC Administrator*"):



On Windows 10, the 32-bit and the 64-bit "*Microsoft ODBC Administrator*" looks nearly identical. (…And on older Windows versions they are 100% identical!).

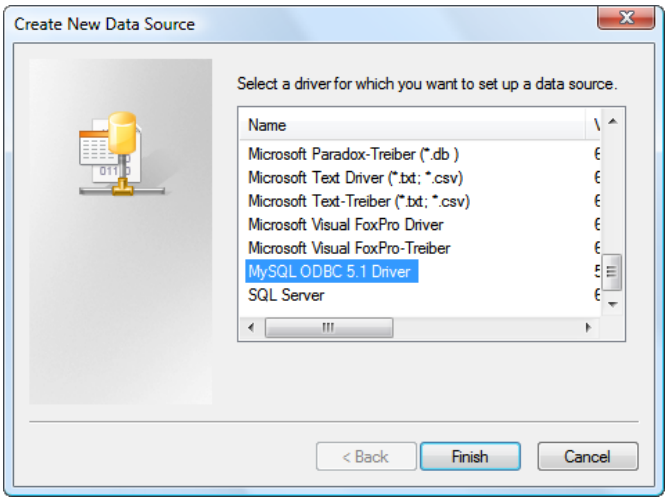
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We can now create a new "ODBC DSN" (i.e. a new "Link" to your database).
You can create 2 types of "ODBC DSN":

1. User DSN
   This is the type of DSN that you will create for most of your applications. Only the current user of this machine can see these DSN's. The Add... button is used to create a new DSN, Remove button is used to delete an existing DSN, and the Configure button is to reset the configuration of a DSN.

2. System DSN
   This type of DSN's are available to all users on this machine (including users running NT Services). If you run some Anatella scripts as part of an automated Windows Service, you should rather create "System DSN's". If you use "simple User DSN" inside a Windows Service, the ODBC connection will be refused inside Anatella because the user executing the Windows Service is not allowed to use "User DSN".

Select the tab ("UserDSN" or "SystemDSN" depending on which type of links you want) and then click on the Add... button. A new window appears. This new window contains all the ODBC drivers installed on your computer. In this example, we will select the "*MySQL ODBC 5.1 driver*" because we want to access some table from a MySQL database:



If you can't see an ODBC driver that you just installed, it means that you are using the 64-bit "Microsoft ODBC Administrator" software (for Anatella 64-bit) and you installed some 32-bit ODBC drivers (or vice versa).

Be sure to install 64-bit ODBC drivers to be able to use Anatella 64-bit

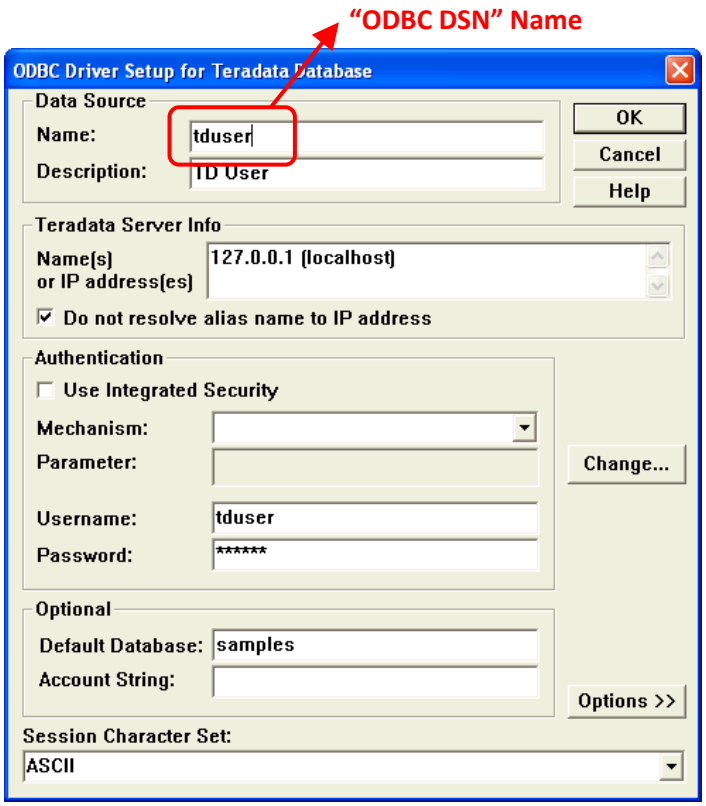
(or install 32-bit ODBC drivers, if you want to use Anatella 32-bit).

For your convenience, you can go here to directly download the most common ODBC drivers (for MS-SQLServer, Oracle, Teradata, MySQL, SQLite, Access, Sybase, DB2):
http://www.anatella.com/html/anatella-download.html
http://download.timi.eu/ODBC/
The above drivers have been thoroughly tested and are known to work well with Anatella.

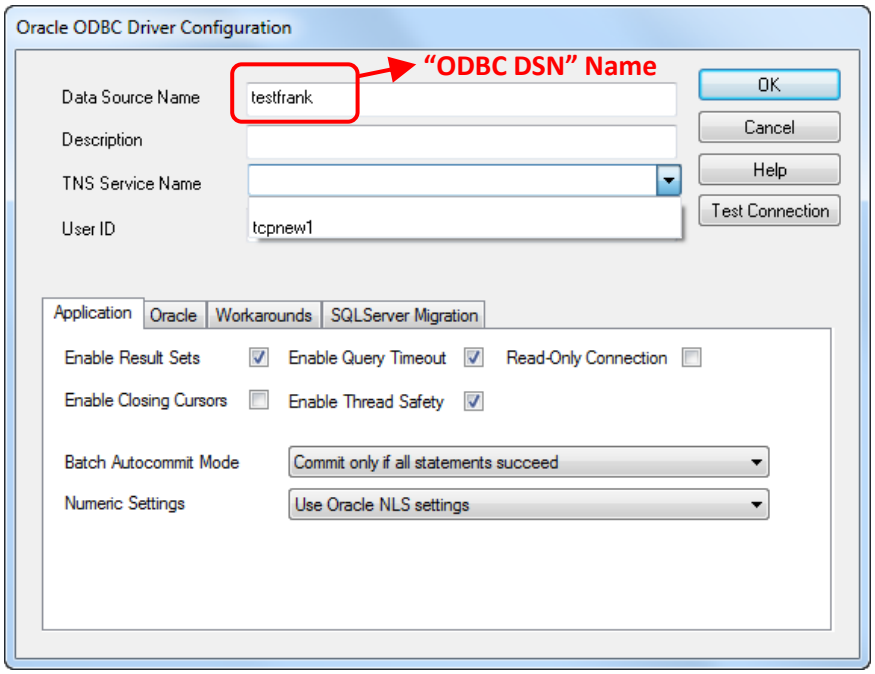
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The next screens are different for each ODBC driver. Here are some examples:

- For a Teradata database, we will have the following configuration window:

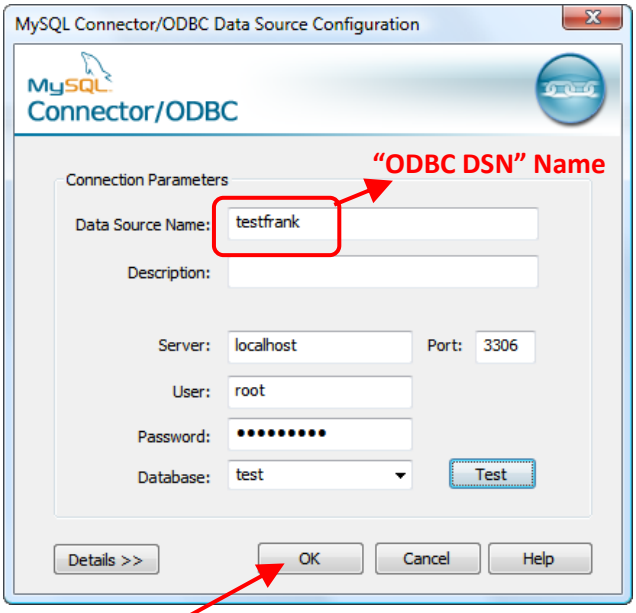


- For an Oracle database, we will have the following configuration window:

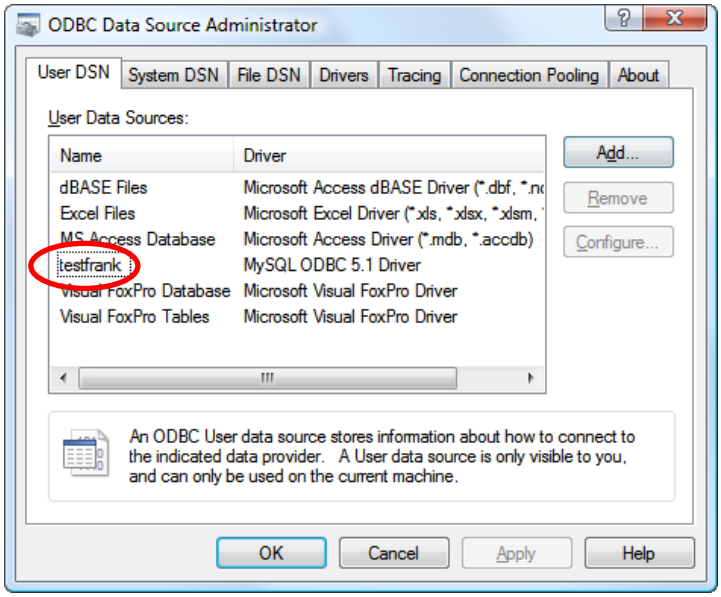


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

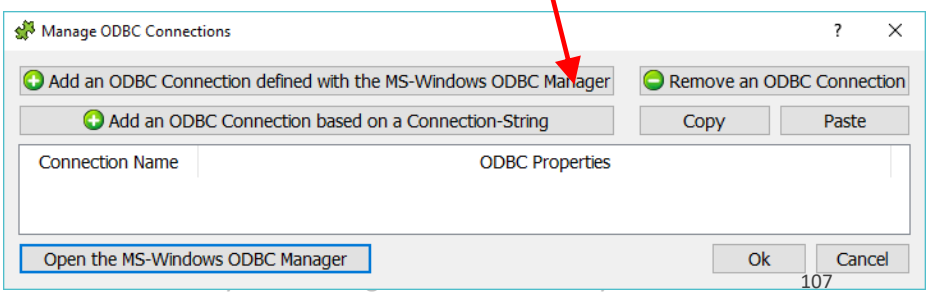
- For a MySQL database, we will have the following configuration window:



Please note that, inside the above example, we used the string "*testfrank*" as "ODBC DSN" name. Click the OK button here. We are now back on the first screen of the "*Microsoft ODBC Administrator*". Please note that a new "ODBC link" (named "testfrank") has been created:

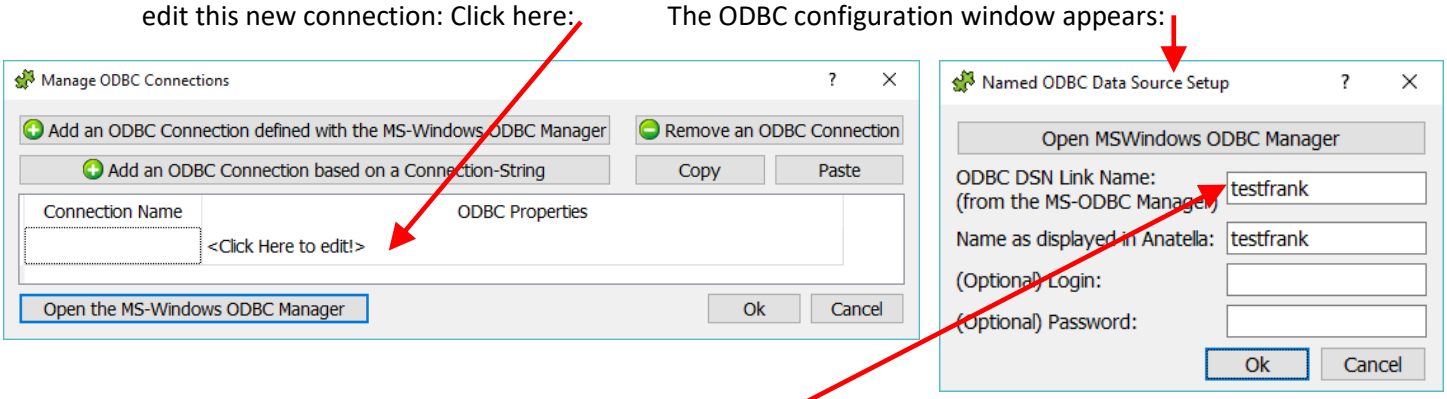


Let's now "go back" to Anatella (i.e. We close the "*Microsoft ODBC Administrator*" windows and re-open the Anatella window). Inside the "*Manage OBC Connections*" window, click the button named "*Add an ODBC connection defined with the MS-Windows ODBC Manager*" here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

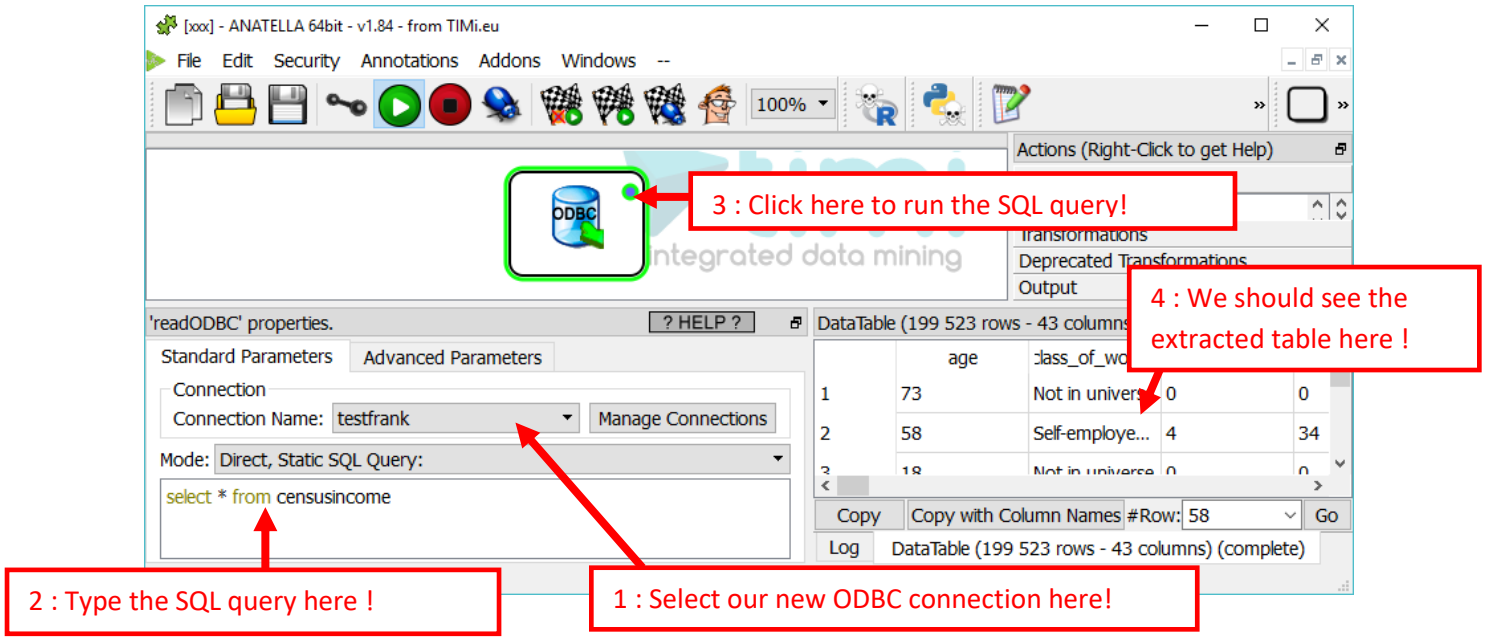
This adds a new connection inside the table listing all ODBC connection available from Anatella. Let's edit this new connection: Click here: The ODBC configuration window appears:



At this point, you must pay attention to write here: the exact same "ODBC DSN" name that you just created inside the "*Microsoft ODBC Administrator*". Anatella uses this "ODBC DSN" name to find, amongst all the already available ODBC connections, the one that you just defined.

The setup of our new ODBC connection is now complete. This means that we can now use our new ODBC connection inside a readODBC action (see section 5.2.3) or inside a upsertODBC action (see section 5.26.4).

For example: Let's use a readODBC action to extract the table "censusincome" from your database:

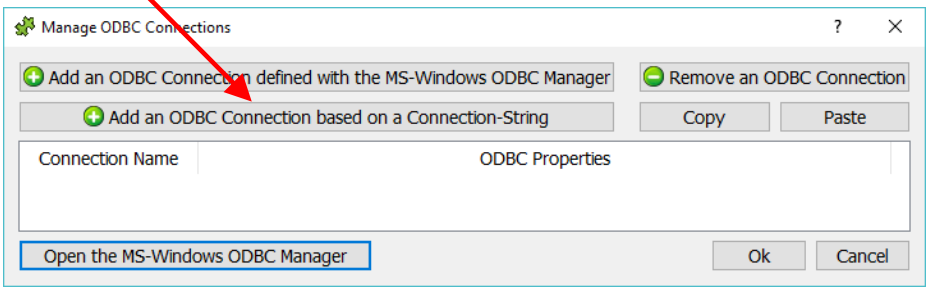


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

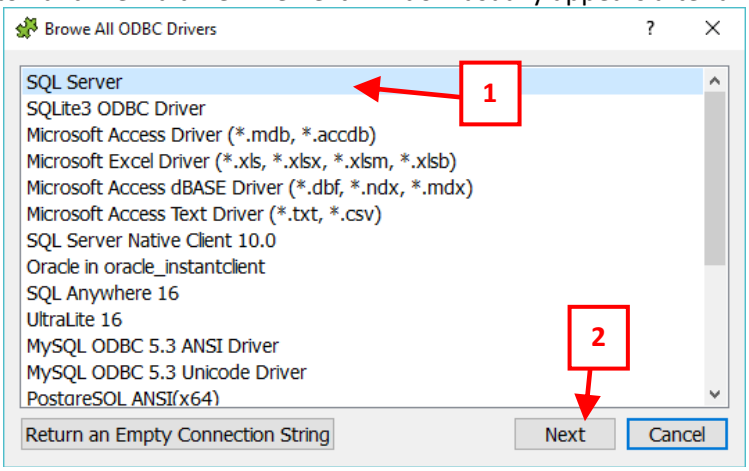
5.1.6.2. Type 2 ODBC connection: Create&Use an ODBC connection based on a Connection-String

Let's give a small example of usage. In the below example, we'll create a new "Type 2" ODBC connection to access the content of a "Microsoft SQLServer database". The technique is similar for all the databases (Oracle, Teradata, SQL Server, DB2, etc.). You'll find additional information for the proper setup of an ODBC connection to an Oracle, Teradata, MySQL & Sybase database in the sections 5.1.6.3. to 5.1.6.7.

Inside Anatella, inside the "*Manage OBC Connections*" window, click the button named "*Add an ODBC connection based on a Connection-String*":



A window that lists all the installed ODBC drivers opens. Since we want to access the content of a "*Microsoft SQLServer database*", we'll select the "*SQL Server*" ODBC driver on the first line (Thereafter, we click the "Next" button and we wait: i.e. The next Window usually appears after a long time):



For your convenience, you can go here to directly download the most common ODBC drivers (for MS-SQLServer, Oracle, Teradata, MySQL, SQLite, Access, Sybase, DB2):
http://www.anatella.com/html/anatella-download.html
http://download.timi.eu/ODBC/
The above drivers have been thoroughly tested and are known to work well with Anatella.

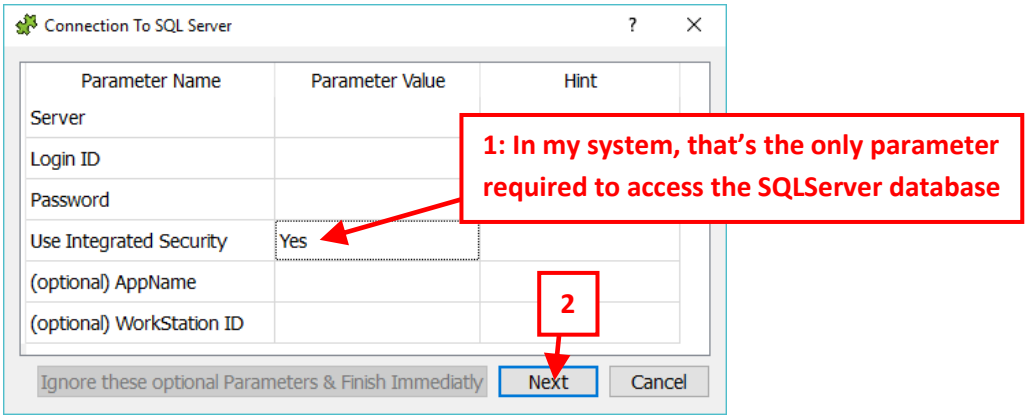
If you cannot see an ODBC driver that you just installed, it means that you are using the 64-bit "Microsoft ODBC Administrator" software (for Anatella 64-bit) and you installed some 32-bit ODBC drivers (or vice versa).

Be sure to install 64-bit ODBC drivers to be able to use Anatella 64-bit

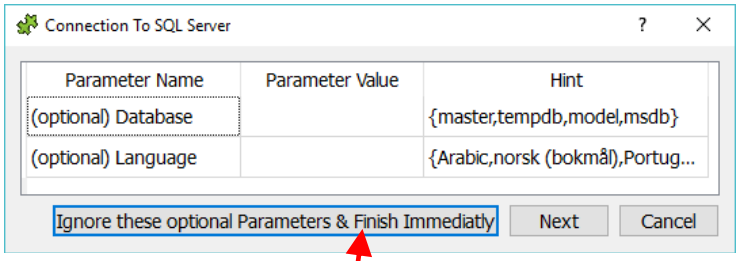
(or install 32-bit ODBC drivers, if you want to use Anatella 32-bit).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

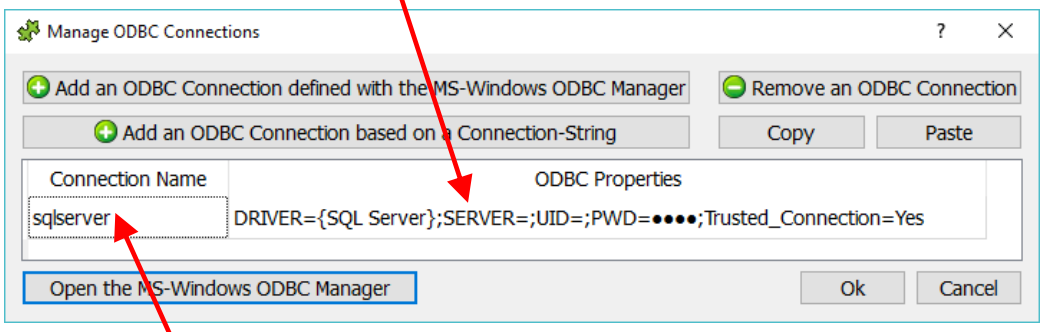
All the next Windows are all based on the same canvas: They are progressively collecting all the required informations to properly contruct the connection string. For example, the first requested parameters are:



The second set of requested parameters are:



Since, inside this second set, all the parameters are optional, you can skip all later steps and directly click the "*Finish Immediately*" button here: We should now have one additional row/connection inside the "*Manage OBC Connections*" window:

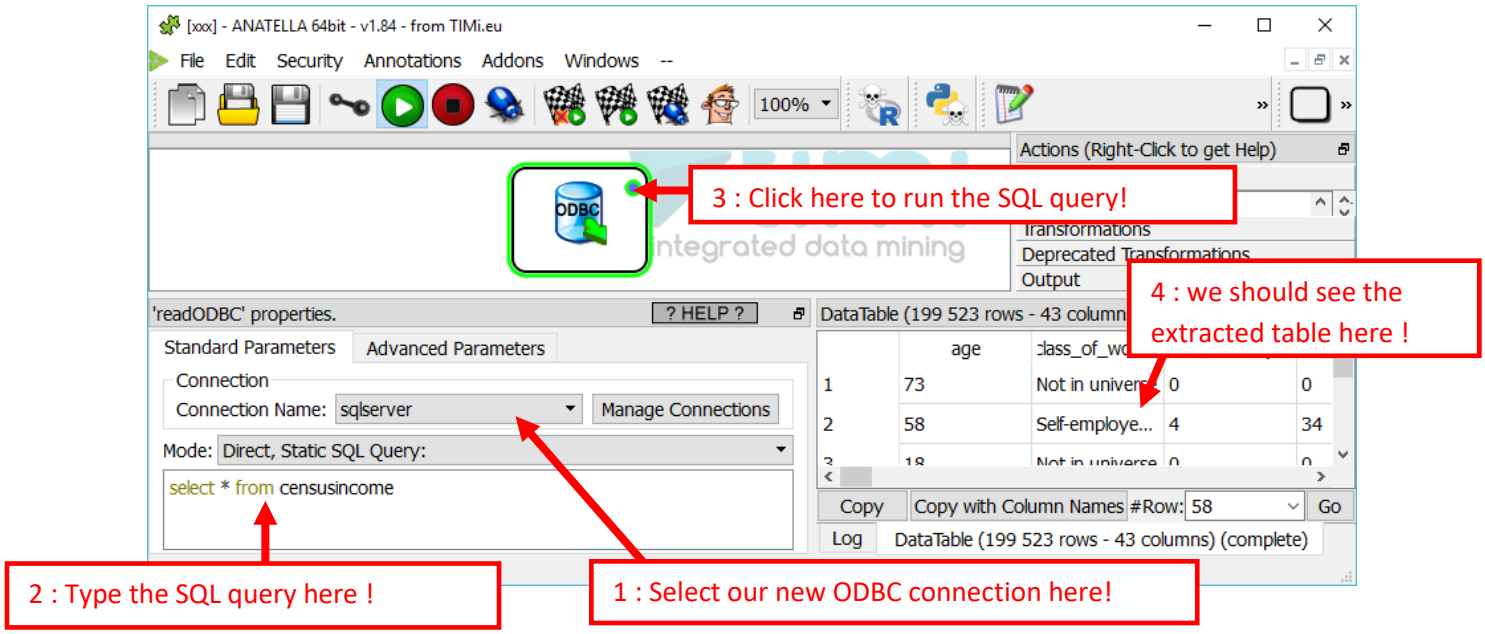


Let's give a proper name to this new connection: In the example above, the name is "sqlserver":

The setup of our new ODBC connection is now complete. This means that we can now use our new ODBC connection inside a readODBC action (see section 5.2.3) or inside a upsertODBC action (see section 5.26.4).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example: Let's use a readODBC action to extract the table "censusincome" from our SQLServer database:

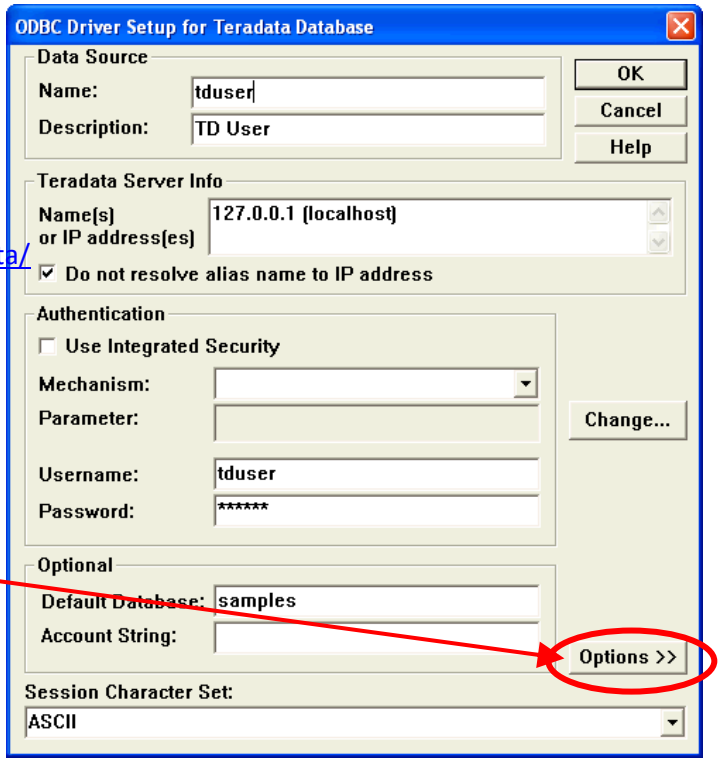


5.1.6.3. Teradata ODBC setup

For your convenience, you can go here to download the latest Teradata ODBC drivers: http://download.timi.eu/ODBC/ODBC_drivers_Teradata/

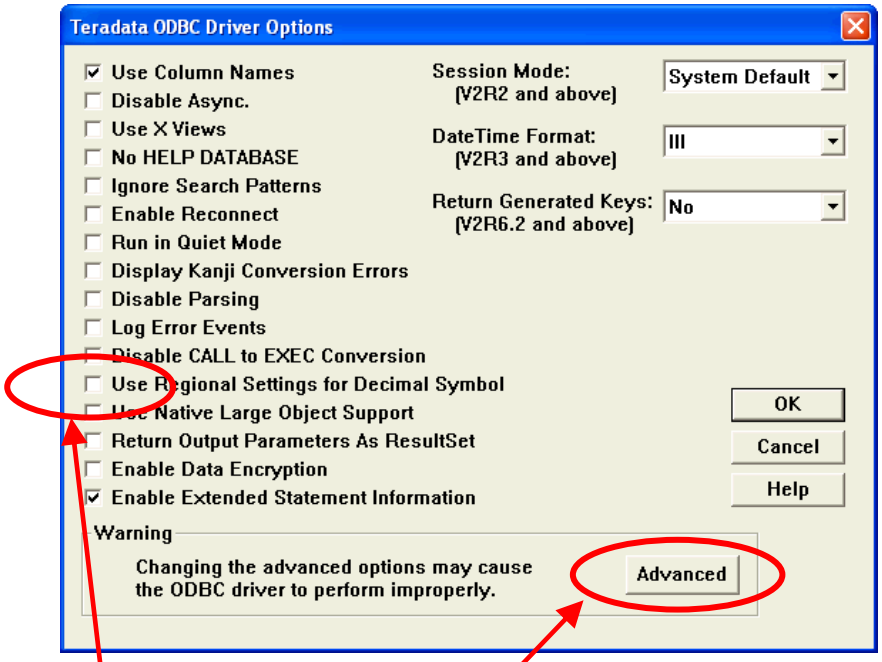
At the same URL, you'll also find the "FastLoad" Teradata utility: See the section 5.26.19. to know more about this subject.

Inside "*ODBC Driver Setup for Teradata Database*", click on the "Options>>" button:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

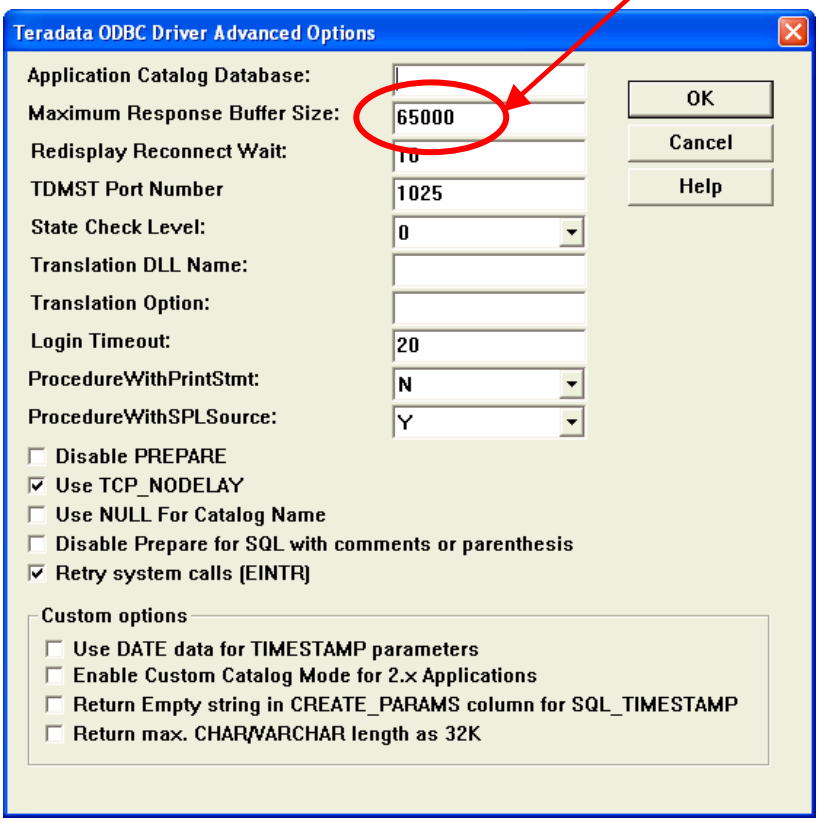
After clicking the "*Options >>*" button, the "*Teradata ODBC Driver Options*" window opens:



Anatella expects the decimal separator for numbers to be the dot (".") whatever the Regional Settings inside the server are. Thus, you need to uncheck the option named "*Use Regional Settings for Decimal Symbol*".

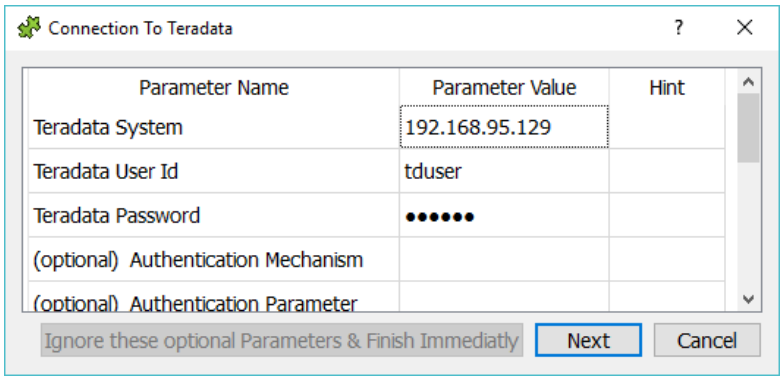
When you click the "*Advanced*" button in the above window, the "*Teradata ODBC Driver Advanced Options*" window opens.

Change the parameter named "Maximum Response Buffer Size" to 65000 (instead of the default value: 8192). This will allow better network performances during communication exchanges between TIMi/Anatella/Stardust and the Teradata database.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is a typical set of parameters to setup an ODBC Connection String (for a "Type 2 ODBC Connection") for accessing the content of a Teradata database:



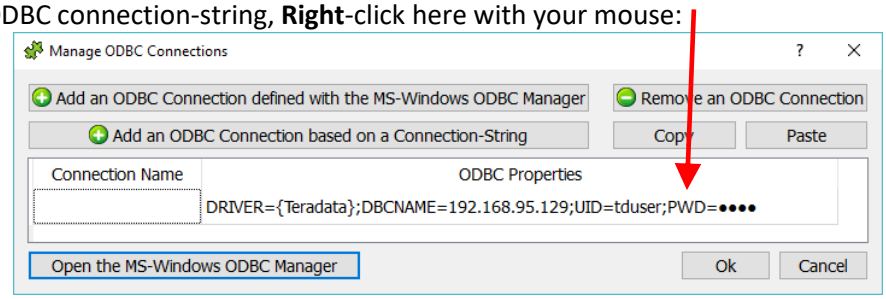
At the end, you should get a connection string that is similar to:

```
DRIVER={Teradata};DBCNAME=192.168.95.129;UID=tduser;PWD=xxx;MaxRespSize=65000;DsnOpt
ions=1000101000000100100000111
```

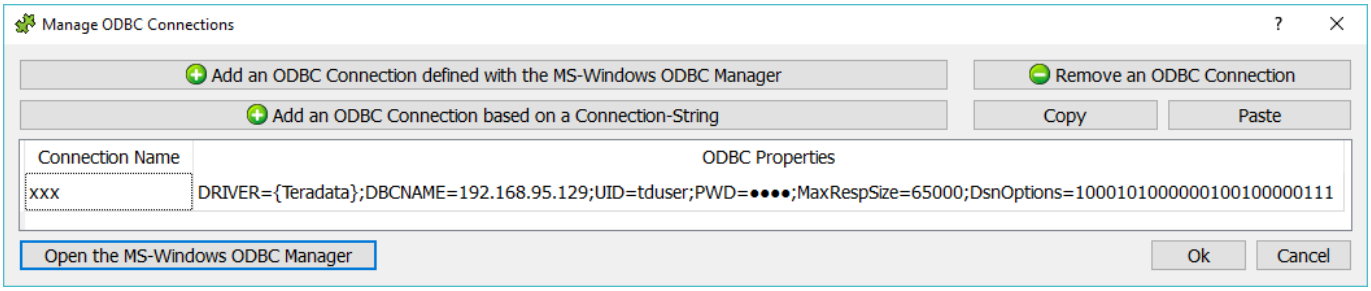
Unfortunately, the Wizard running inside Anatella (to contruct the Connection-String) does not set the last 2 parameters (named "MaxRespSize" and "DsnOptions") to the proper value: i.e. You should manually edit the Connection-String and append at the end:

```
;MaxRespSize=65000;DsnOptions=1000101000000100100000111
```

To manually edit the ODBC connection-string, **Right**-click here with your mouse:



After the edition of the Connection-String, you should have something that looks like this:



5.1.6.4. Oracle ODBC Setup

The "*Oracle Instant Client ODBC*" driver is very quick to install. The connection from Anatella to Oracle is usually setup in a few minutes.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

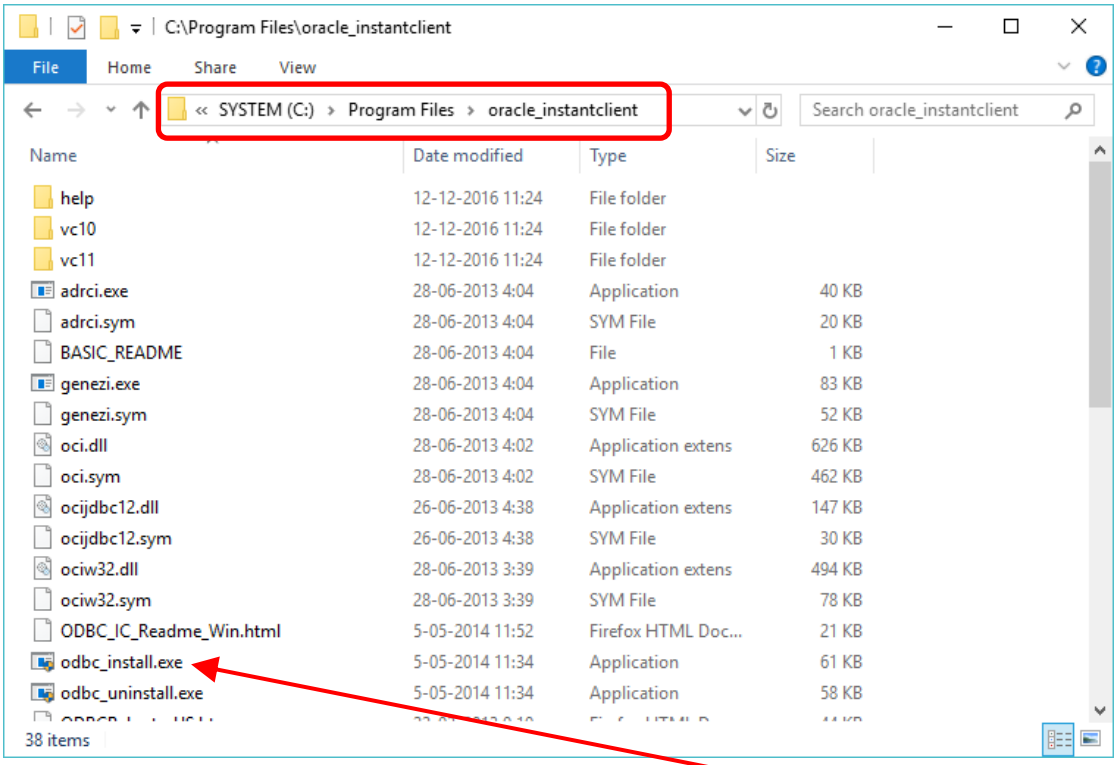
Follow these steps:

1. Unzip the zip file containing the driver:
   - Select the 64-bit ODBC driver if you plan to use the 64-bit Anatella.
   - Select the 32-bit ODBC driver if you plan to use the 32-bit Anatella.

   For your convenience, you can go here to download the latest Oracle ODBC drivers:
   http://download.timi.eu/ODBC/ODBC_drivers_Oracle/

2. **Important**: Don't click *now* on the "odbc_install.exe" executable that's inside the ZIP file (otherwise the installation of the "instant client" will fail): First and before, you have to move the extracted content of the zip file to the final destination folder: Typically, you'll have:
   On a 64-bit system for a 64-bit ODBC driver: Copy the files into "C:\Program Files"
   On a 64-bit system for a 32-bit ODBC driver: Copy the files into "C:\Program Files (x86)"
   On a 32-bit system for a 32-bit ODBC driver: Copy the files into "C:\Program Files"

   For example, on my 64-bit system (and 64-bit ODBC driver), I will have the following files:



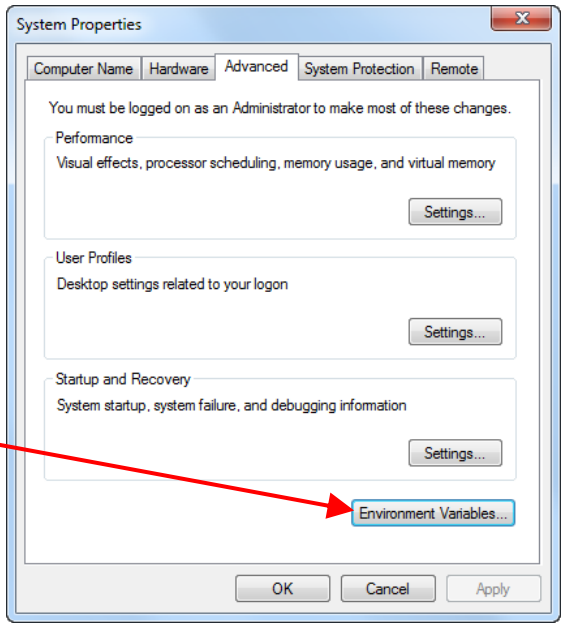
   Now, you can double-click on the "odbc_install.exe" executable here:
   If you didn't move the files in the final location before running the "odbc_install.exe" executable, you'll have to uninstall everything and restart from step 1.

3. For the ODBC driver to work properly, the easiest way is to ask you DBA for the "TNSNAMES.ORA" file that contains all the connections parameters to oracle. Once you have this file, you must copy it inside the directory stored inside the DOS environment variable "TNS_ADMIN". If the environment variable variable "TNS_ADMIN" is not set (and this is the default for the "Instant Client"), you'll have to set it up yourself.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

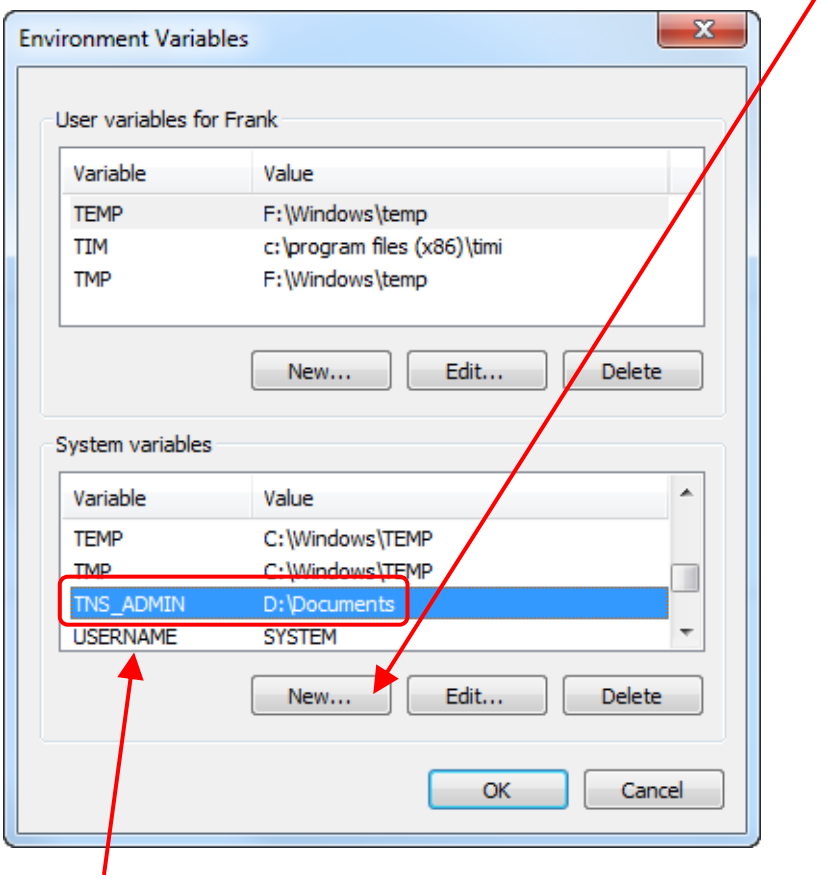
To setup the environment variable variable "TNS_ADMIN":

- Open the control panel, select system, select Advanced system settings: the "system properties" window opens:



Go to the "Advanced" tab and click the "Environment Variables" button:

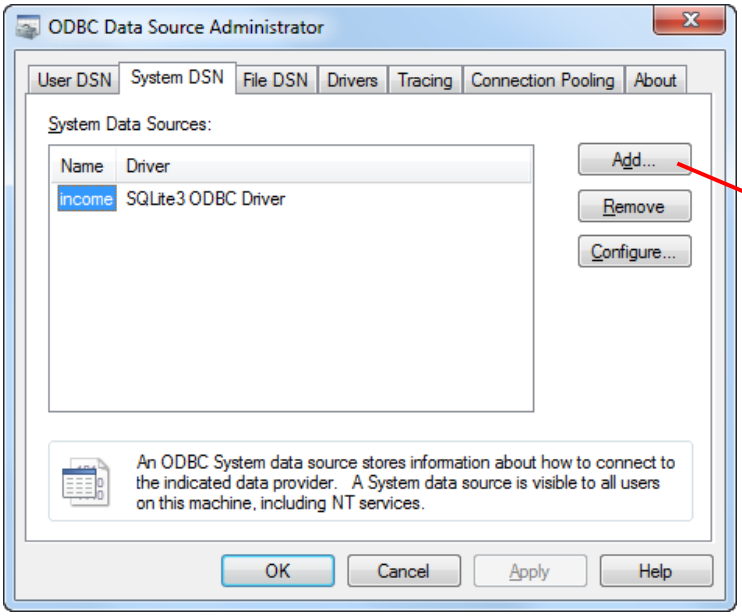
- The "Environment Variables" Windows open: click on the "New" button:



- Define the value of the "TNS_ADMIN" environmental variable (in the example above, it's "D:\Documents"). **At this point, you need to log-off/log-in so that MS-Windows takes into account the new value of the "TNS_ADMIN" environmental variable. If you don't log-off/log-in (or even better: reboot the computer), the ODBC drivers won't work. This is the most common cause of failure when using the Oracle ODBC drivers.**
- Copy the "TNSNAMES.ORA" file inside the proper directory (in the example above, it's "D:\Documents")

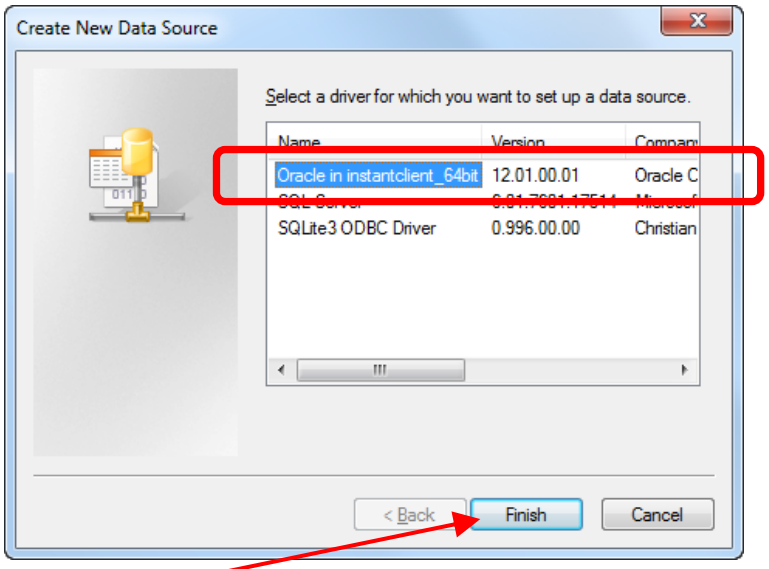
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. Test the ODBC driver: Run the "*Microsoft ODBC Data Source Administrator*": To open this Window: See section 5.1.6.1. ("Type 1 ODBC Connection"). You should now see:



Click on the "Add…" button

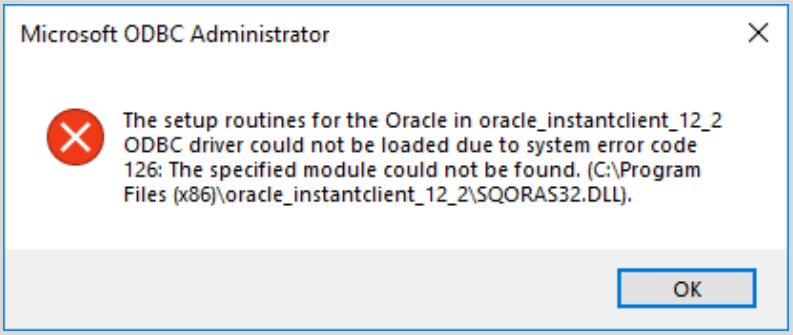
… select "Oracle in instantclient_64bit":



… and click the "Finish" button.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

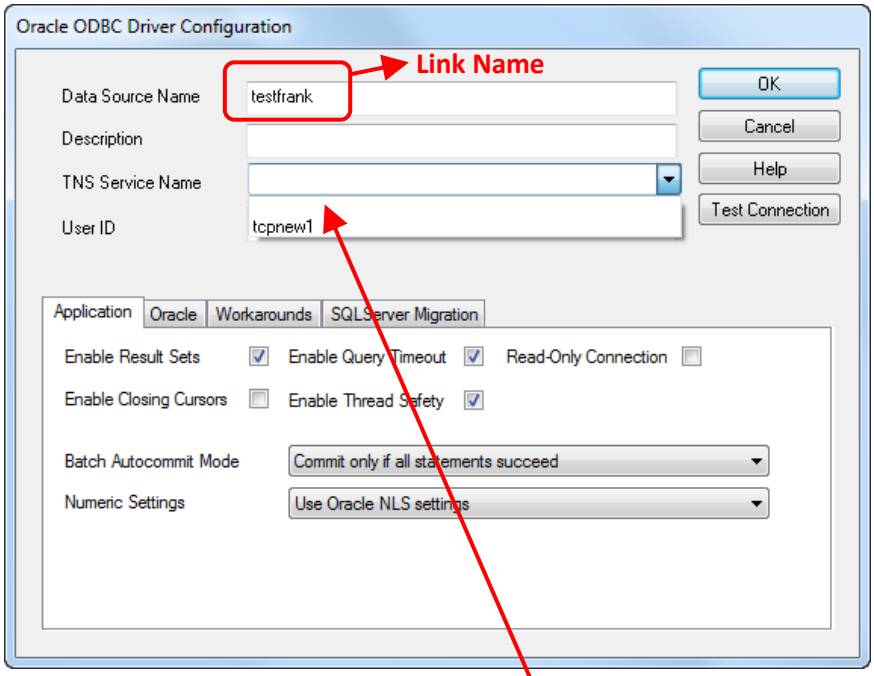
At this point, if you see this window:



…this means that you need to install the "Visual Studio 2013 Redistributables". The installation "Setup" files for the "Visual Studio 2013 Redistributables" are in the same location as the Oracle drivers:

http://download.timi.eu/ODBC/ODBC_drivers_Oracle/

You should now see:



In particular, if the "TNSNAMES.ORA" file has been properly installed, you'll see the list of available connections to Oracle on the drop-down combo menu here: . If you don't see anything, you might need to log-off/log-in (so that the MS-Windows takes into account the new value of the "TNS_ADMIN" environmental variable).

You can setup the connection and click the "Test connection" to see if everything works properly.

The "test connection" button returns the "connection successful" message ***even if the "TNS Service Name" parameter is improperly set*** (This is a really disturbing behvavior). If you get the following message inside Anatella:

SQL_Error::SQL Error State:HY000, Native Error Code: 3110, ODBC Error: [Oracle][ODBC][Ora]ORA-12560: TNS:protocol adapter error

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

…then you need to go back to the "Oracle ODBC Driver Configuration" window and change the "TNS Service Name" to a valid value.

Sometime you get the following error when you use the ODBC connection:

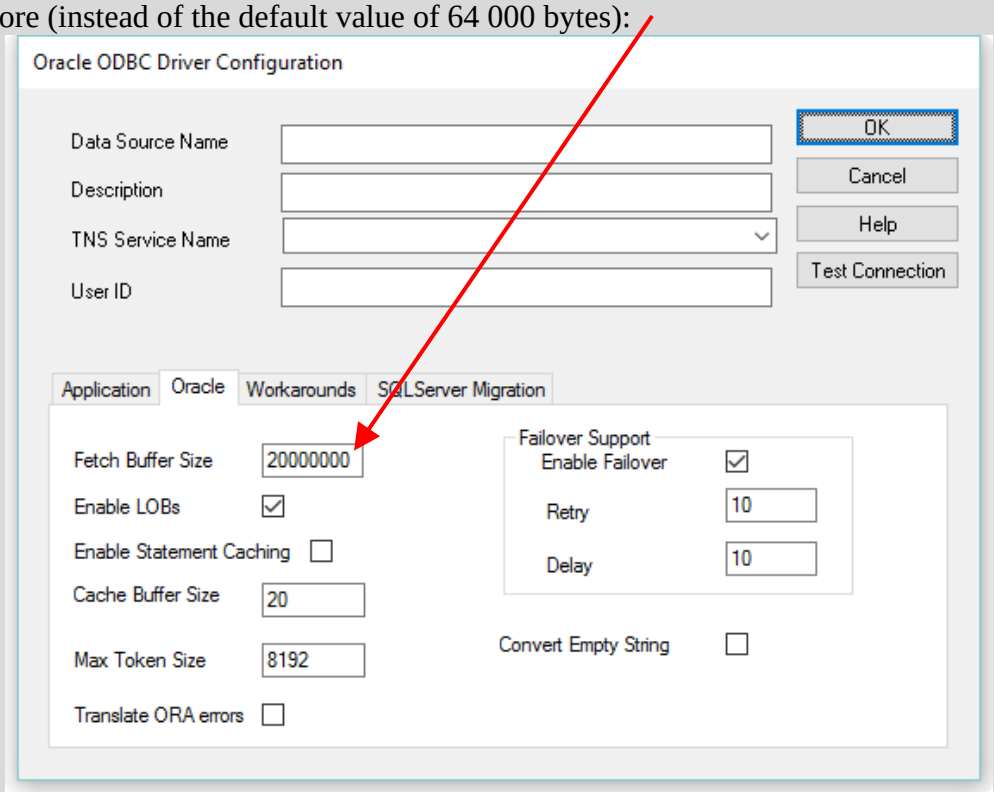
```
[Oracle][ODBC][Ora]ORA-12170: TNS:Connect timeout occurred
```

This error is disturbing because it happens even when the "*test connection*" button (inside the configuration panel of the ODBC driver) returns "connection successful".

This error means many different things. Most of the time, to solve the issue, you simply need to open a few firewalls. Sometime it means that the Oracle ODBC drive couldn't parse properly the "TNSNAMES.ORA" file. The solution is to use a Type-2 ODBC connection with the Oracle-Server-IP (or URL Name) and the Oracle-Service-Name directly embedded inside the ODBC connection-string. More precisely, you'll have an ODBC connection-string that looks like this:

DRIVER={Oracle in oracle_instantclient};UID=frank;PWD=xxx;DBQ={*IP of the server*}/{*Service Name*}

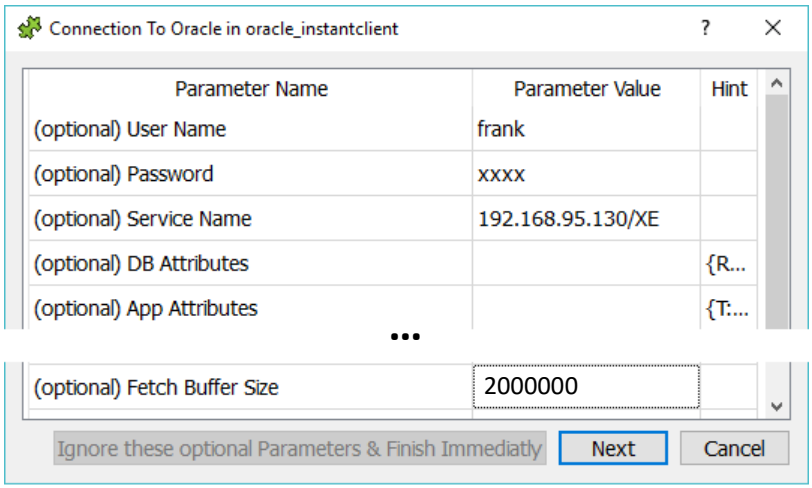
To improve the performance of the Oracle ODBC driver, you should increase the "Fetch Buffer Size" parameter of the ODBC Driver to 2 000 000 bytes or more (instead of the default value of 64 000 bytes):



The above parameter is particularly important when the Oracle Database is located on a very distant server compared to the machine running Anatella.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is a typical set of parameters used to setup an ODBC Connection String (for a "Type 2 ODBC Connection") for accessing the content of an Oracle database:

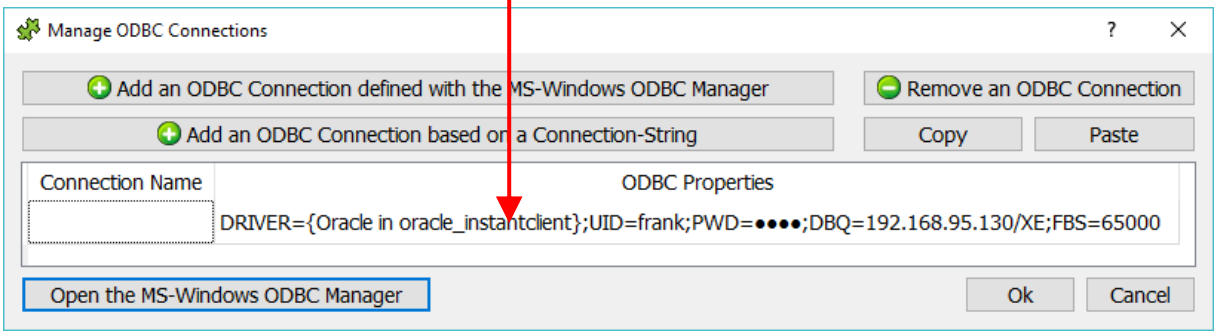


At the end, you should get a connection string that is similar to:

DRIVER={Oracle in oracle_instantclient};UID=frank;PWD=xxx;DBQ=192.168.95.130/XE;FBS=2000000

IP of Oracle Server **Oracle Service Name**

You can right-click with your mouse here: to manually edit the Connection-String (e.g. to add some missing part):



Let's assume that you are now using the Oracle ODBC connection inside Anatella (e.g. you are running a readODBC Action): If you receive an error message about an "*inconsistent architecture*" when running the Action, it means that you are trying to access a 32-bit Oracle Server using a 64-bit ODBC driver (or vice-versa). This is a well-known limitation of the "basic" drivers distributed by Oracle: The "basic" 32-bit ODBC drivers from Oracle can only connect to a 32-bit Oracle Database Server (and the 64-bit ODBC drivers can only connect to a 64-bit Oracle Database Server). There are several solutions to bypass this limitation that originates from Oracle (i.e. other databases do not have this limitation).

Let's now assume that you have a 32-bit Oracle Database Server (thus only the 32-bit ODBC driver will work) and you want to use Anatella 64-bit for most of your processing.

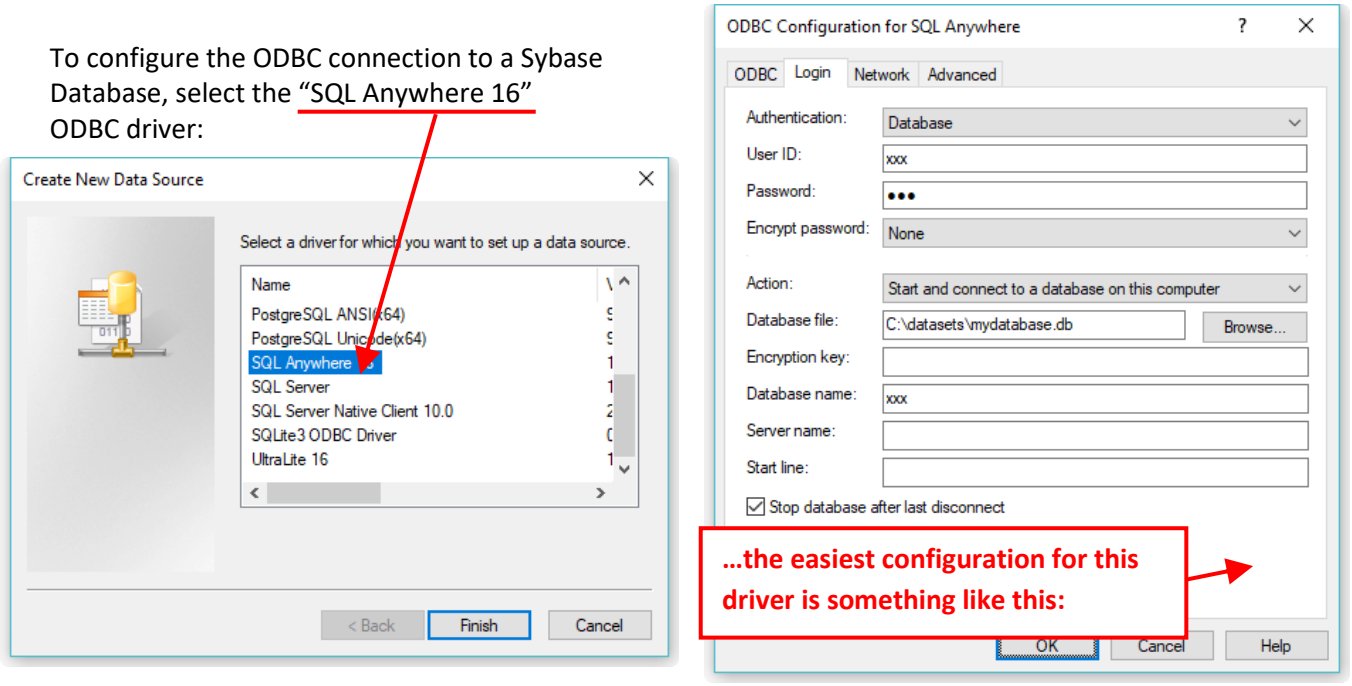
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To bypass the "*inconsistent architecture*" limitation of the default Oracle ODBC drivers:

- You can install on the same PC, at the same time, both the "Anatella 32-bit" and the "Anatella 64-bit". This means that you can still use the fastest "Anatella 64-bit" to do all your data transformations and only use the Anatella 32-bit to extract the required data from Oracle (i.e. you save every extracted tables inside some .gel_anatella files that you can use later with the normal Anatella 64-bit). You'll find mode informations on how to install simultaneously the "Anatella 32-bit" and the "Anatella 64-bit" on the same PC inside the section 10.11.
- Use the Oracle OCI driver (see section 5.2.26.) or the Oracle OleDB driver (see section 5.2.3.): These 2 drivers do not have the limitation of the ODBC driver.
- Use an ODBC-ODBC bridge. Here is a classical scenario: You can use an ODBC-ODBC bridge to run the 32-bit ODBC driver inside the "Anatella 64-bit".
  Several vendors are selling ODBC-ODBC bridges: For example:
  http://www.easysoft.com/products/data_access/odbc_odbc_bridge
- Use another ODBC driver (not the default one provided by Oracle): For example:
  https://www.progress.com/products/data-sources/oracle-odbc-jdbc-drivers

5.1.6.5. Sybase ODBC Setup

To configure the ODBC connection to a Sybase Database, select the "SQL Anywhere 16" ODBC driver:



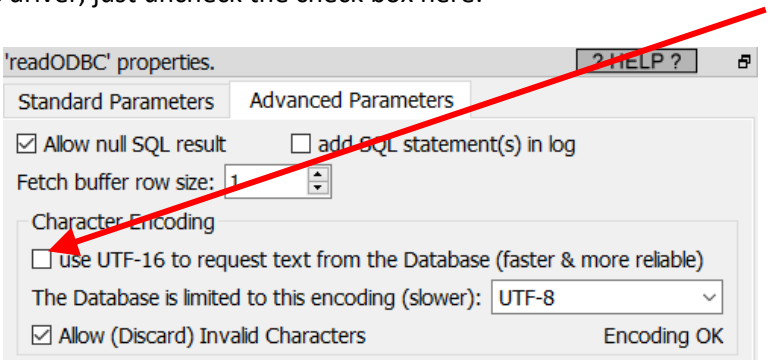
You can obtain a free version of the the "SQL Anywhere 16" ODBC driver (server+client) here:
http://sqlanywhere.de/en

For your convenience, you can go here to download the ODBC drivers for Sybase:
http://download.timi.eu/ODBC/ODBC_drivers_Sybase/

This driver is great to easily access a Sybase database because you don't need to install any Sybase database on your server to access the content of your .db file: i.e. The "SQL Anywhere 16" ODBC driver is the only thing that you need to install to get access to your data. However, the "SQL Anywhere 16" ODBC driver has an annoying limitation: it does not support the standard UTF-16 (unicode) character

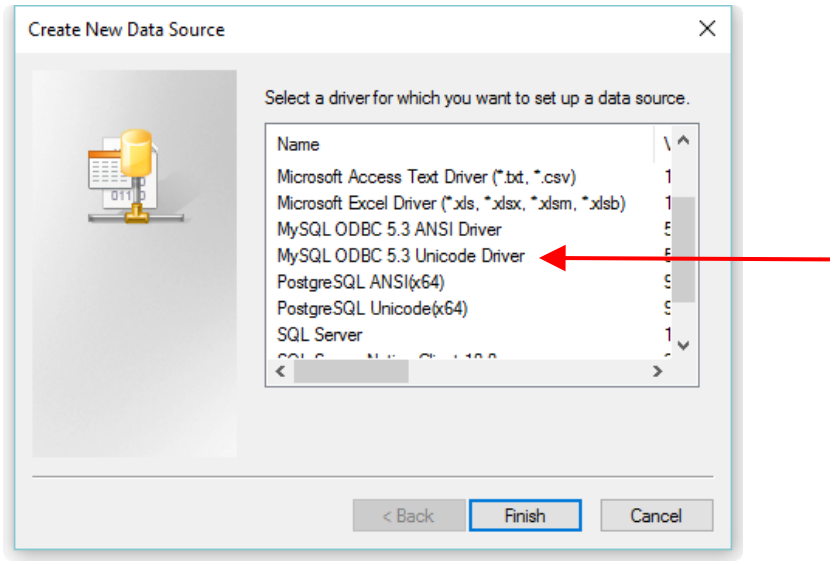
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

set (i.e. it fails silently returning only empty cells instead of text). To bypass this limitation of the "SQL Anywhere 16" ODBC driver, just uncheck the check box here:



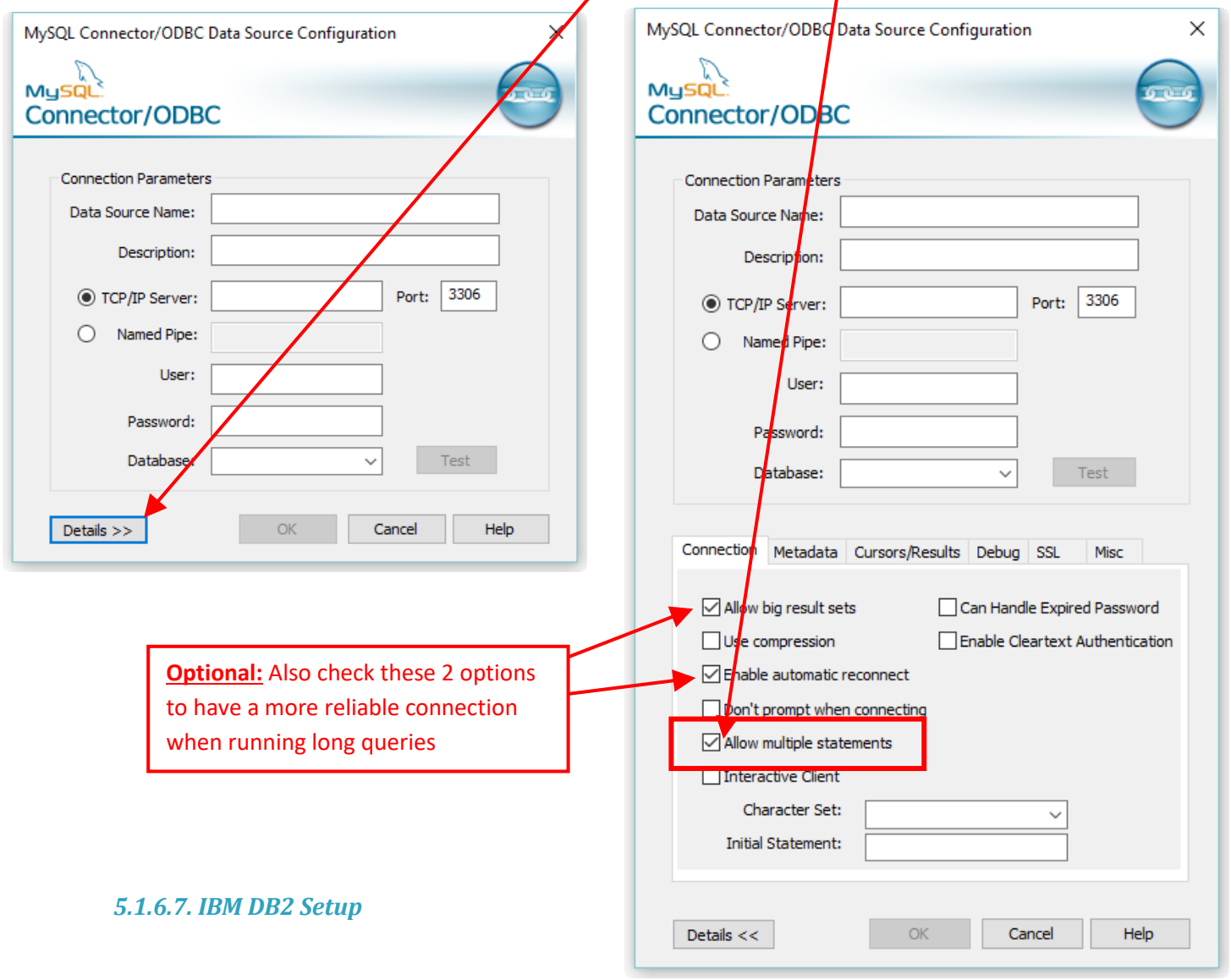
5.1.6.6. My SQL ODBC Setup

To configure the ODBC connection to MySQL, select the Unicode driver (and not the ANSI driver):



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Inside the ODBC configuration window, you must enable the "Allow Multiple Statements" option. To do so, first click on the "Details >>" button here and then click here:



5.1.6.7. IBM DB2 Setup

5.1.6.7.1. ODBC Driver installation

The "*IBM DB2 ODBC*" driver is very quick to install. The connection from Anatella to DB2 is usually setup in a few minutes.

Follow these steps:

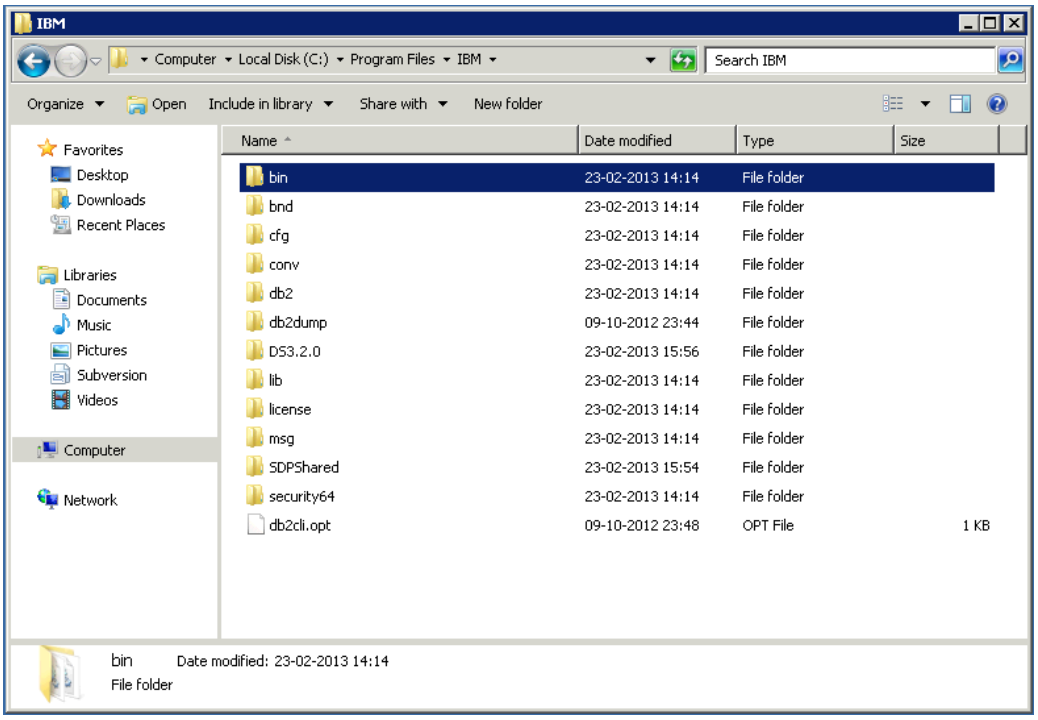
1. Unzip the zip file containing the IBM DB2 ODBC driver.

   For your convenience, you can go here to download the latest DB2 ODBC drivers: http://download.timi.eu/ODBC/ODBC_drivers_IBM_DB2/

2. **Important**: Don't run *now* any executable that originates from the ZIP file: First and before, you have to move the extracted content of the zip file to the final destination folder: Typically, you'll have:
   On a 64-bit system for a 64-bit ODBC driver: Copy the files into "C:\Program Files\IBM"
   On a 64-bit system for a 32-bit ODBC driver: Copy the files into "C:\Program Files (x86)\IBM"
   On a 32-bit system for a 32-bit ODBC driver: Copy the files into "C:\Program Files\IBM"

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example, on my 64-bit system (and 64-bit ODBC driver), I will have the following files:



3. Open an “Administrator Command Prompt” (right-click on the command prompt and choose "Run as administrator"), go to the install folder (e.g. “c:\program files\IBM\bin”) and type:

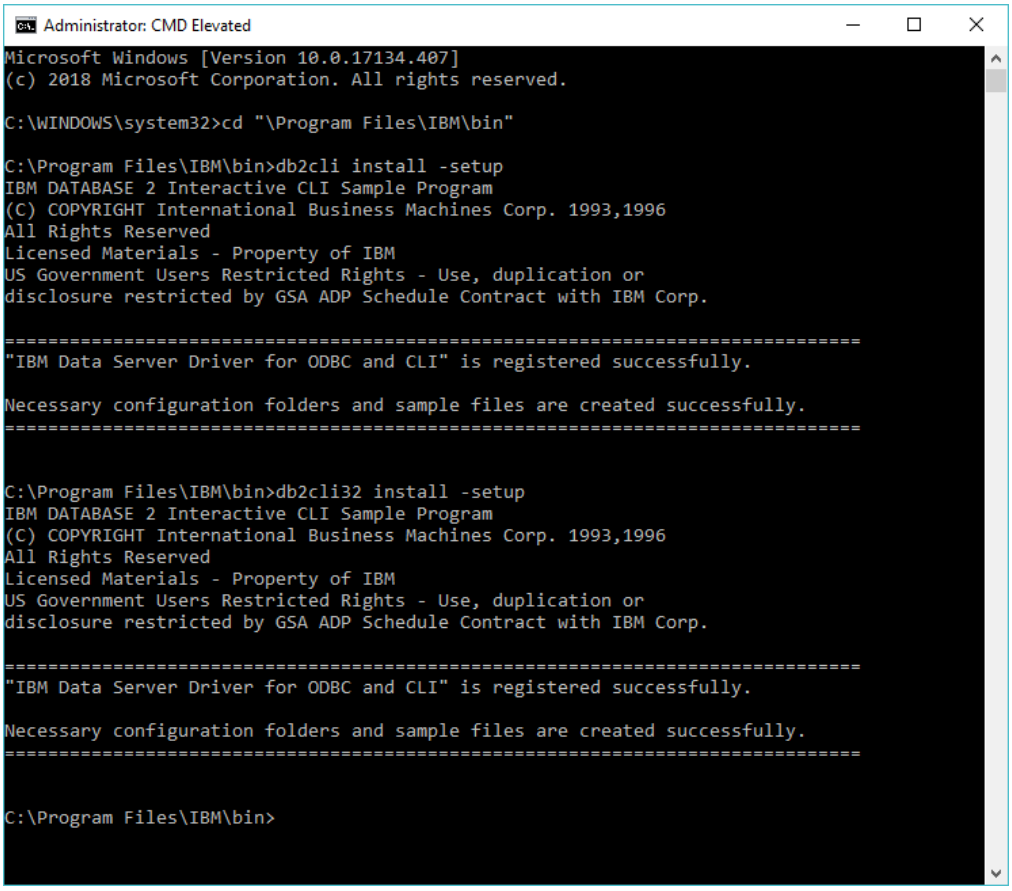
```
cd <uncompressed driver folder>/bin
db2cli install -setup
db2cli32 install -setup
```

This installs the 64-bit ODBC driver

This installs the 32-bit ODBC driver

Here is an illustration of the procedure:

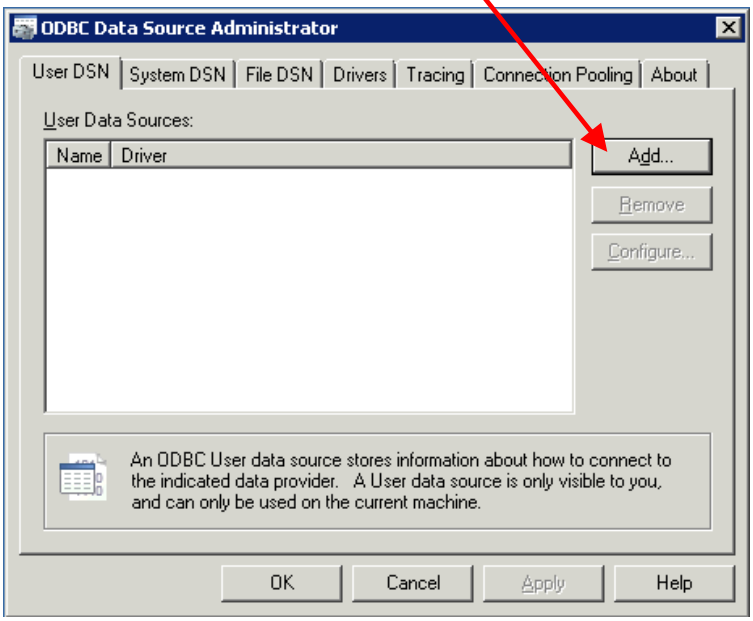
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



5.1.6.7.2. ODBC Driver Configuration to create a Type 1 ODBC connection to DB2

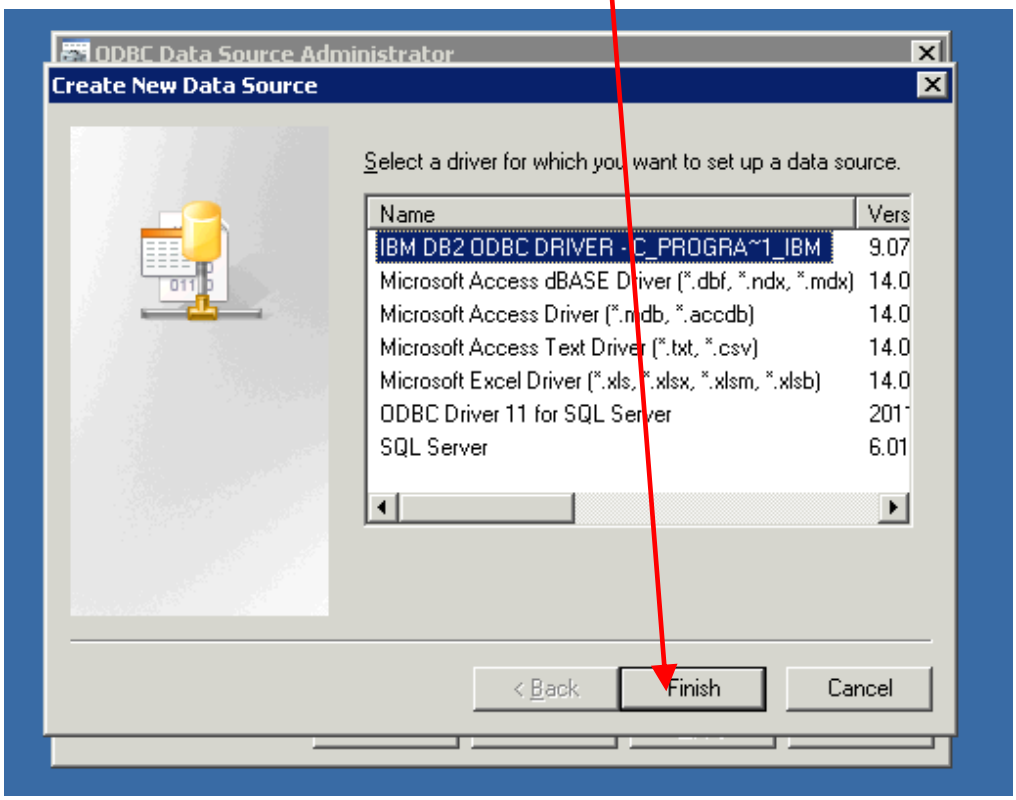
The procedure to create a Type 1 ODBC connection to IBM DB2 is the following:

1. Open the "*Microsoft ODBC Administrator*" (see the section 5.1.6.1. to have more details on how to open the "*Microsoft ODBC Administrator*")
2. Click the "Add…" Button to create a new ODBC DSN:



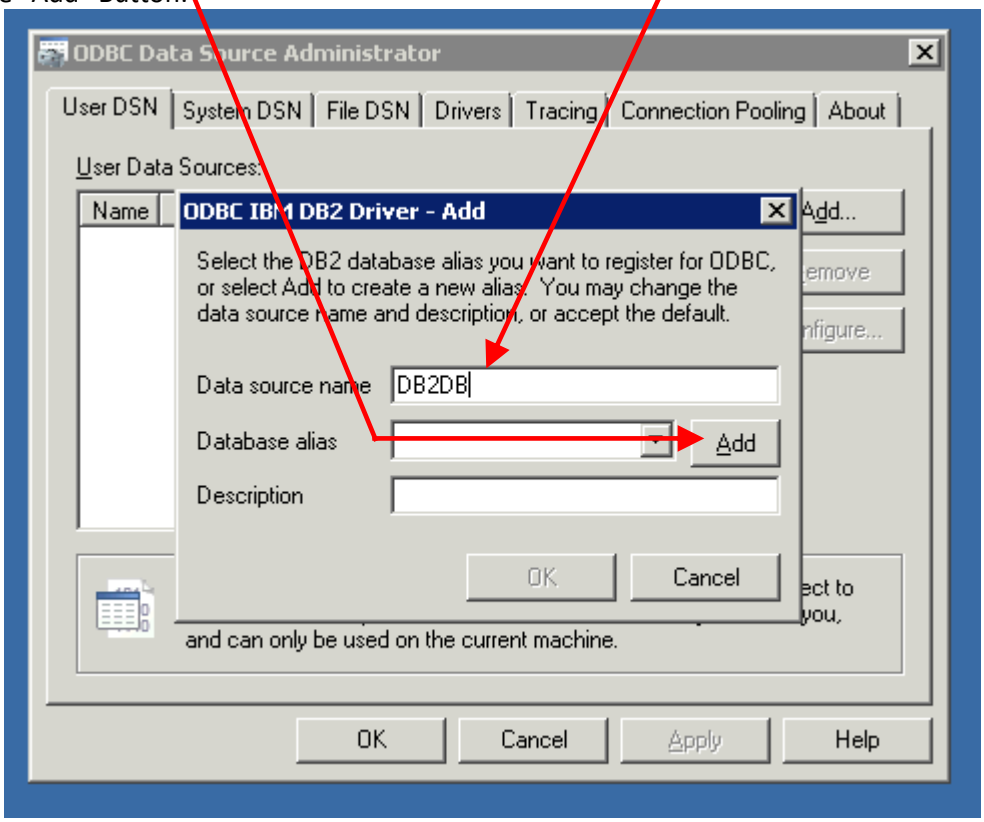
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Choose the DB2 ODBC-driver and press the Finish-button:



4. Give now the "*ODBC DSN Link name*" that you'll use inside Anatella to select the connection to your DB2 database. Here I have just given it the name DB2DB.

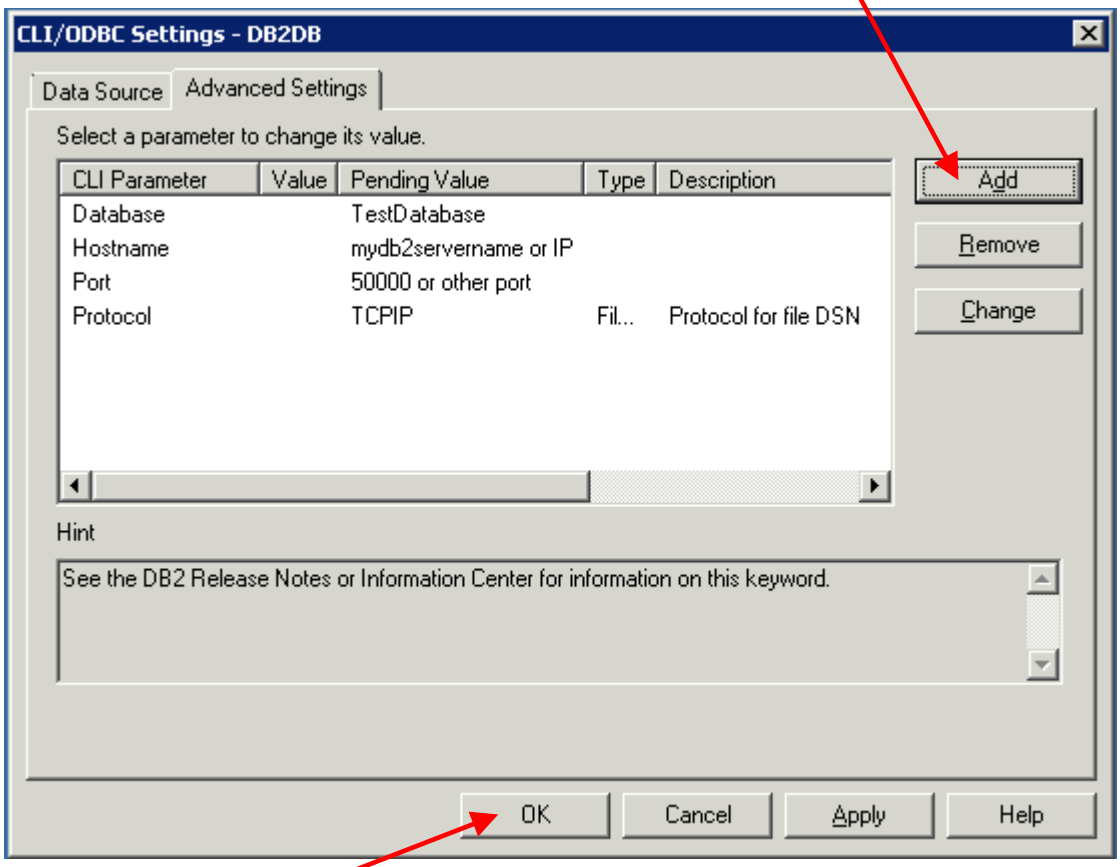
5. Click the "Add" Button:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. Go to the Advanced Settings-tab and press the "Add" button here several times: i.e. you need to "add" the following settings:
    - Database: Should point to the database that you want to connect to.
    - Hostname: Should point to the server where your DB2-server is installed.
    - Port: The portnumber where the DB2-server on the Hostname is responding.
    - Protocol: The protocol used to connect to the DB2-server (most of the time, select TCP/IP) .

    The final settings should be something like this:



7. Click the "OK" button here and you are done!

5.1.6.7.3. DB2 Unicode Character Support

Inside Anatella, all the characters are Unicode. This means that Anatella can handle any characters (e.g. any Chinese chars, any Cyrilic chars, any accentuated chars).

By default, inside DB2, the fields that are declared as "VARCHAR()" are not Unicode. For example, this means that DB2 will report an error if you use Anatella to insert any accentuated character (such as é,è,à,…) inside a "VARCHAR()" field. The solution is to never use any "VARCHAR()" fields and always use "VARGRAPHIC()" fields instead (when you declare a new field containing some characters inside a "CREATE TABLE" statement). More details: The "VARGRAPHIC()" fields are Unicode and can accommodate any character that Anatella might want to INSERT.

To remind you: The CreateTable automatically generates a "CREATE TABLE" statement based on the data contained inside the table to INSERT inside the database. Unfortunatelly, by default, inside the auto-generated "CREATE TABLE" statement, all the columns that contain some characters are declared as "VARCHAR()" and not "VARGRAPHIC()": i.e. This won't work properly inside DB2. This

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

means that you must manually edit the auto-generated "CREATE TABLE" statement to replace all "VARCHAR" strings with the "VARGRAPHIC" string. Alternatively, instead of manually editing the "CREATE TABLE" statement, you can use the following JS code inside the "*For Expert Users*" panel of the CreateTable Action:

```
function run()
{
    var s="",i,n=columns.length,c;
    if (dropTable) s+="DROP TABLE IF EXISTS "+tableName+";\n";
    s+="CREATE TABLE "+tableName+" (\n";
    for(i=0;i<n;i++)
    {
        c=columns[i];
        s+="  "+c.name+" ";
        if (c.declaration!=null)  s+=c.declaration;
        else
        {
             switch (c.type)
             {
             case 'F':
                    if (useFloat53) s+="float(53)";
                    else                  s+="float(24)";
                    break;
             case '0': case 'K':
                   s+="dec("+Math.ceil(mmax(c.len,1)*integerF1+integerF2)+")";
                   break;
             case 'U':
                   s+="vargraphic(50)";
                   break;
             default :
                   s+="vargraphic("+Math.ceil(mmax(c.len,1)*varCharF1+varCharF2)+")";
                   break;
             }
             if (pkIsNotNull&&c.isPK) s+=" NOT NULL"
        }
        if (i+1!=n) s+=",\n";
    }

    if (autoPK.length>0) s+=",\n  "+autoPK+" NOT NULL AUTO_INCREMENT";
    else if (listPK.length>0)
    {
        n=listPK.length;
        s+=",\n  PRIMARY KEY ("+listPK[0];
        for(i=1;i<n;i++) s+=","+listPK[i];
        s+=")";
   }
   s+="\n);\n";
   return s;
}
```

5.1.6.8. Microsoft SQL Server

The official link to download the MS-SQL-Server ODBC drivers is here:
https://docs.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server?view=sql-server-ver15

For your convenience, a copy of the ODBC drivers is here:
http://download.timi.eu/ODBC/ODBC_drivers_SQLServer/

There is nothing special to say about the MS SQLServer ODBC drivers: Everything works smoothly and directly.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.6.9. PostGre SQL Setup

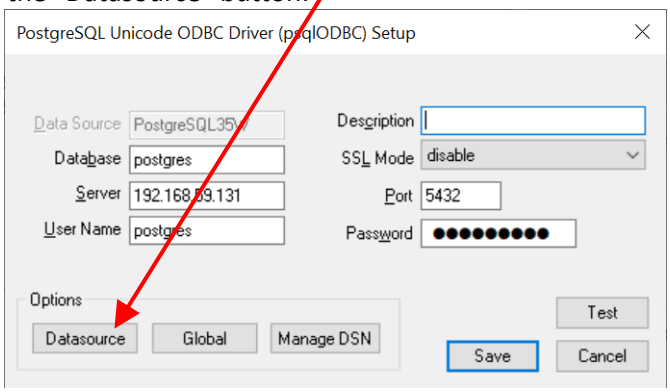
The official link to download the PostgreSQL ODBC drivers is here:
https://www.postgresql.org/ftp/odbc/versions/msi/

For your convenience, a copy of the ODBC drivers is here:
http://download.timi.eu/ODBC/ODBC_drivers_PostgreSQL/

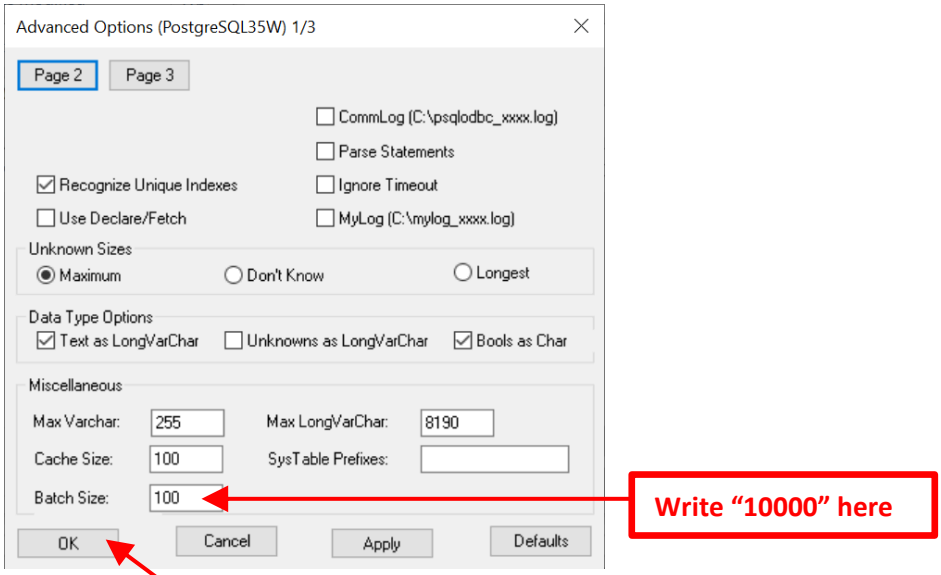
We won't give any details about the installation of the ODBC driver because it's straightforward. When using PostgreSQL, you have the choice between using an "ANSI" or using an "Unicode" ODBC driver: you should use the "Unicode" ODBC driver.

When using PostgreSQL, the "insert" speed is particularly slow compared to other databases (for example, despite using the "trick" given here below, PostgreSQL "insert's" are still around 20 times slower than MS-SQL-Server "insert's"). To get a better "insert" speed:

1. Open the configuration panel of the PostgreSQL ODBC driver inside the ODBC Manager and click on the "Datasource" button:



2. Increase the "batch size" parameter to "10000" (or higher):



3. Click the "OK" button

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.6.10. MS-Access Setup

The official link to download the Access ODBC+OleDB drivers is here:
https://www.microsoft.com/en-us/download/details.aspx?id=54920

For your convenience, a copy of the ODBC+OleDB drivers is here:
http://download.timi.eu/ODBC/ODBC_drivers_Access/

If you use Anatella 64bit, you need to install the 64bit ODBC/OleDB MS-Access drivers to connect to your Access database.
If you use Anatella 32bit, you need to install the 32bit ODBC/OleDB MS-Access drivers to connect to your Access database.

If you don't care about using the *Anatella SQL-Wizard* (to be able to create your SQL commands with the mouse), you can setup very quikly and easily a new "Type 2" ODBC connection to any MS-Access database: Just use the following ODBC connection string:

```
Driver={Microsoft Access Driver (*.mdb, *.accdb)};Dbq=.\mydatabase.accdb;Uid=Admin;Pwd=;
```

You can (and should) use *relative* filepath towards your .accdb file here: (instead of *absolute* filepath). *Relative* filepaths must start with these 2 characters: ".\" or "./".

If you want to use the Anatella SQL-Wizard to create your SQL commands with the mouse, then you need to authorize Anatella to read the table 'MSysObjects' inside MS-Access. To do so, follow this procedure (repeat this procedure for each MS-Access database that you manage with the SQL wizard):

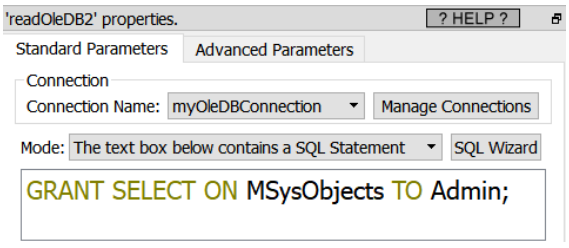
1. Find the location of your "System.mdw" file.
   In my computer, it was inside this directory:
   C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Access
2. Inside Anatella, setup an OleDB connection to you MS-Access database (it must be an OleDB connection: an ODBC connection won't work). Test it.
3. Inside Anatella, edit your OleDB connection string to add at the end:
   ;Jet OLEDB:System database=<the filepath to your "System.mdw" file>
   This means that your OleDB connection string should now looks something like this:

```
Provider=Microsoft.ACE.OLEDB.12.0;User ID=Admin;Data Source=F:\TIMi\TestDatabase.accdb;Jet OLEDB:System database=C:\Users\frank\AppData\Roaming\Microsoft\Access\System.mdw
```

4. Run inside the Action "readOleDB2" inside Anatella the following SQL command:

```
GRANT SELECT ON MSysObjects TO Admin;
```

   Inside Anatella, this looks like this:



5. **Optional:** Test if the whole procedure is ok: Run the following SQL command:
   SELECT * FROM MSysObjects
   You should see a table inside the Anatella Data preview. The procedure failed if you see:

```
ODBC Error 'Exec':State:42000, Native Error Code:FFFFF88D, Error Text: [Microsoft][ODBC Microsoft Access Driver] Record(s) cannot be read; no read permission on 'MSysObjects'.
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.6.11. RedShift Setup

The official link to download the Redshift ODBC drivers is here:
https://docs.aws.amazon.com/redshift/latest/mgmt/configure-odbc-connection.html#install-odbc-driver-windows

For your convenience, a copy of the ODBC drivers is here:
http://download.timi.eu/ODBC/ODBC_drivers_Redshift/

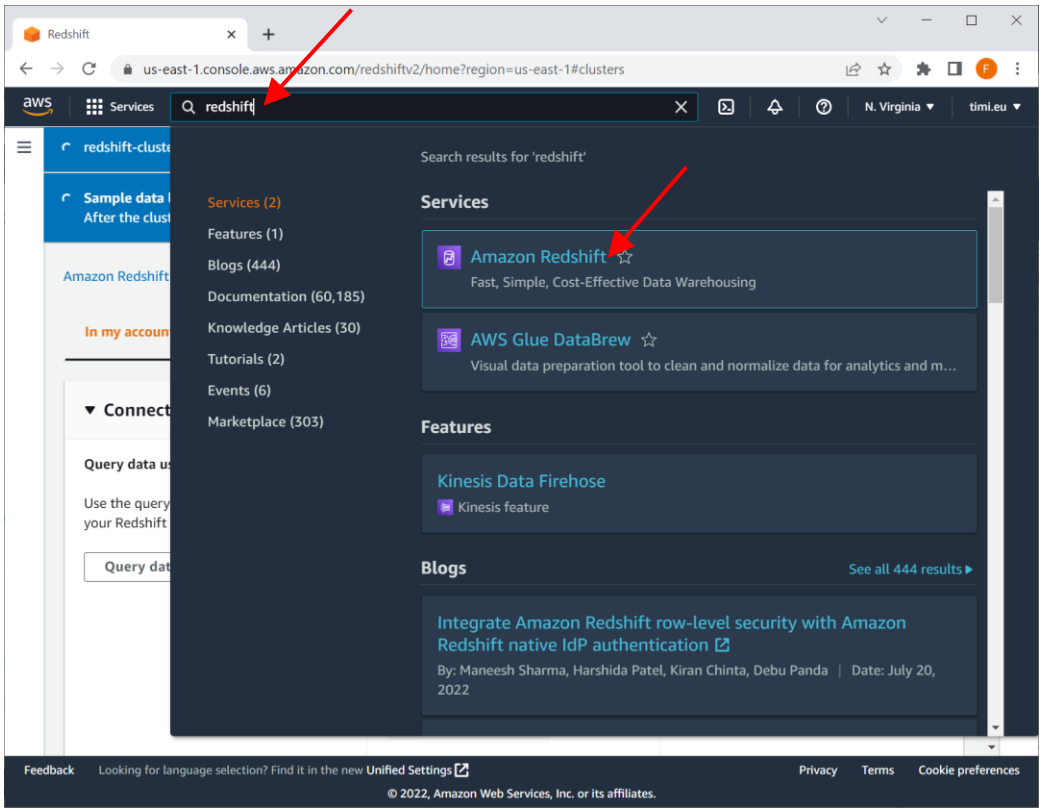
The Redshift ODBC driver can execute "Select" SQL statement but not "Insert" or "update" statements. If you need to "insert" rows inside a RedShift database, you should use the special Action named "RedShift Bulk Upload" described in section 5.24.2.

By default, you can only access a Redshift database from a limited set of machines hosted inside the AWS network. You will need to manually configure Redshift to be able to access it from a machine located outside AWS. This manual configuration is in two steps:

1. Make Redshift Publicly accessible:

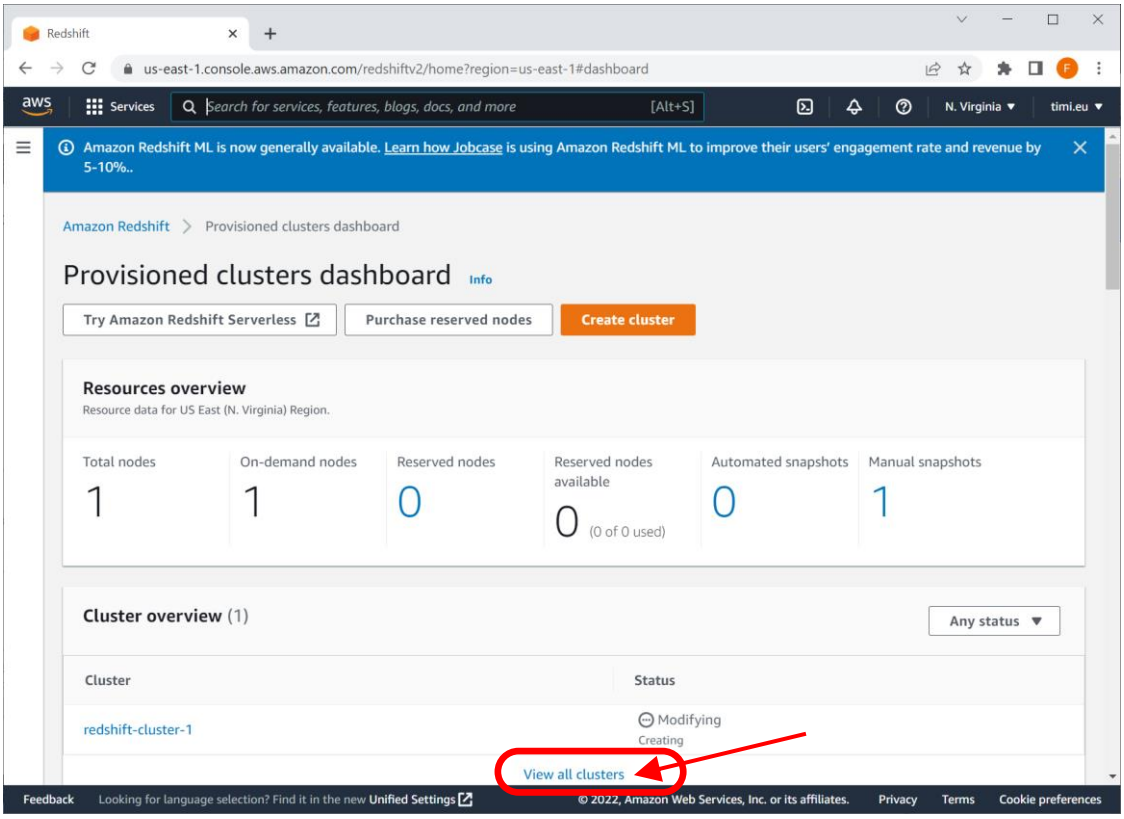
   1.1. Open the Amazon console: https://console.aws.amazon.com and login into AWS.

   1.2. Type "redshift" inside the search bar and click on "Redshift" in the results:

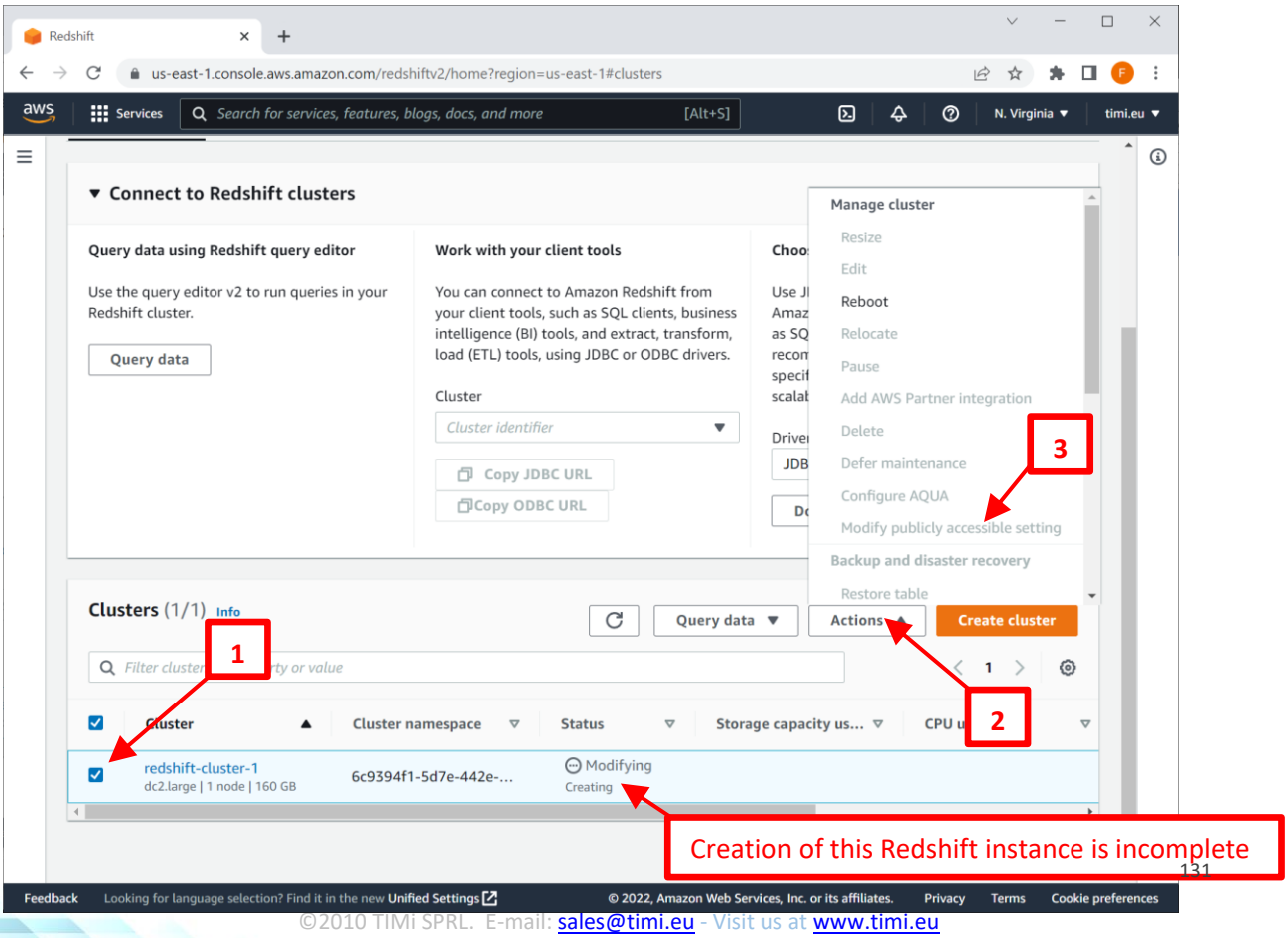


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1.3. Click on “View all Clusters”:

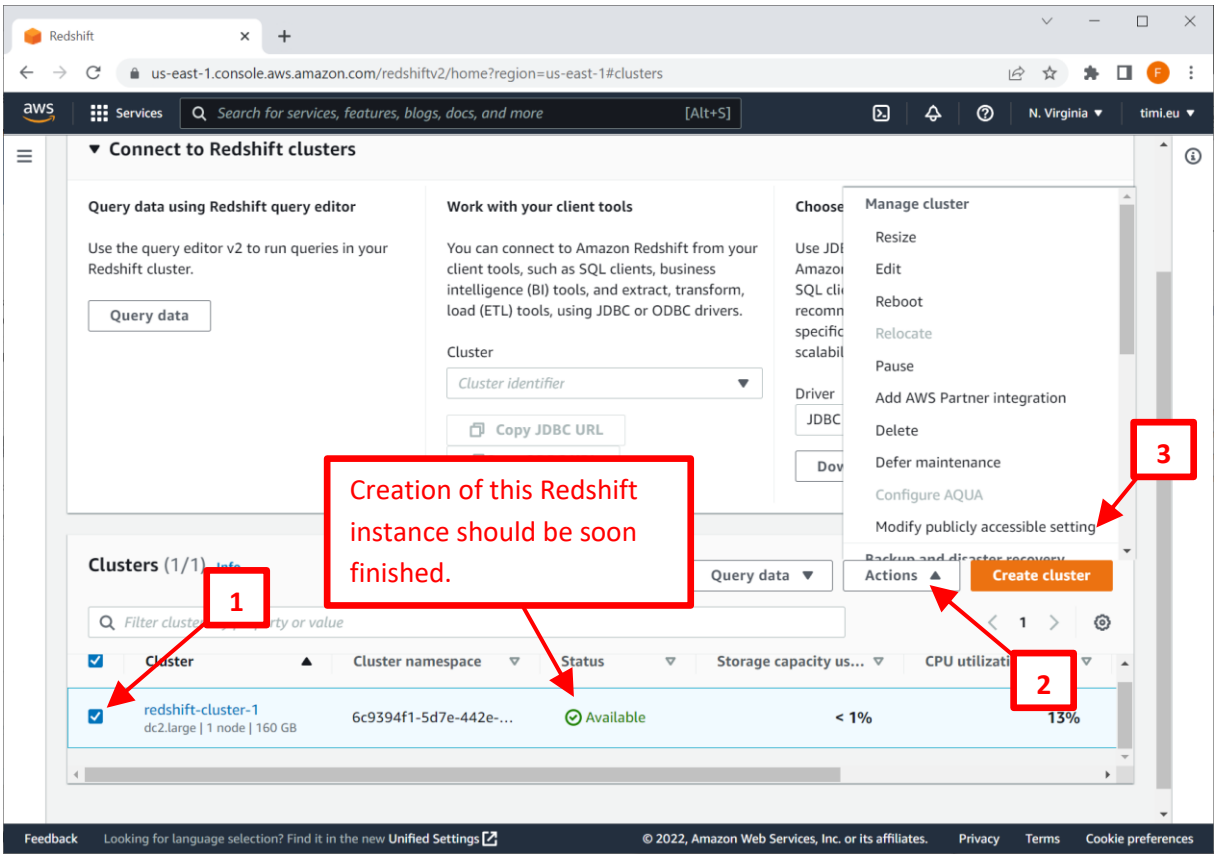


1.4. Check the “checkbox” in front of the redshift instance that you want to access with Anatella, then click on the “Actions” button, then click on the “Modify publicly accessible setting” option. In the screenshot below this option is grayed-out. This is because the creation of the redshift-cluster is not complete.

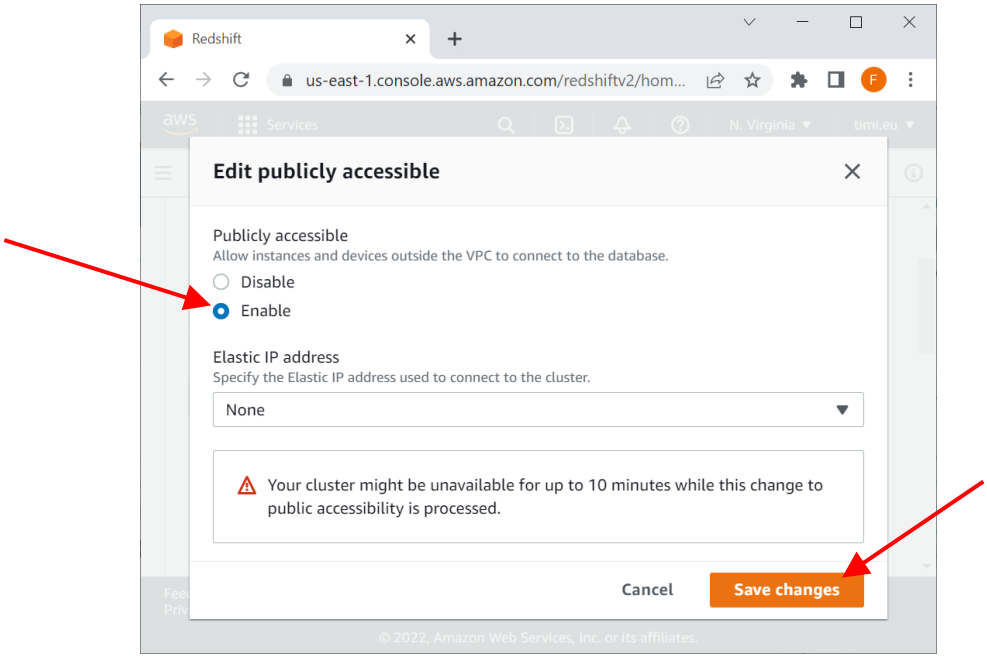


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1.5. Wait until your redshift-cluster is with the “available” status. This can take up to half an hour. When the status of your redshift-cluster is marked as “available”, you should be able to see the “Actions” menu that is not grayed-out. If that’s not the case (i.e. the “Actions” menu is still grayed out), you just need to wait a little bit more (up to half an hour! Regularly press F5 to refresh the page). Inside the “Actions” menu, select the “Modify publicly accessible setting” option.



1.6. Select “Enable” and click on the “Save Changes” button at the bottom:

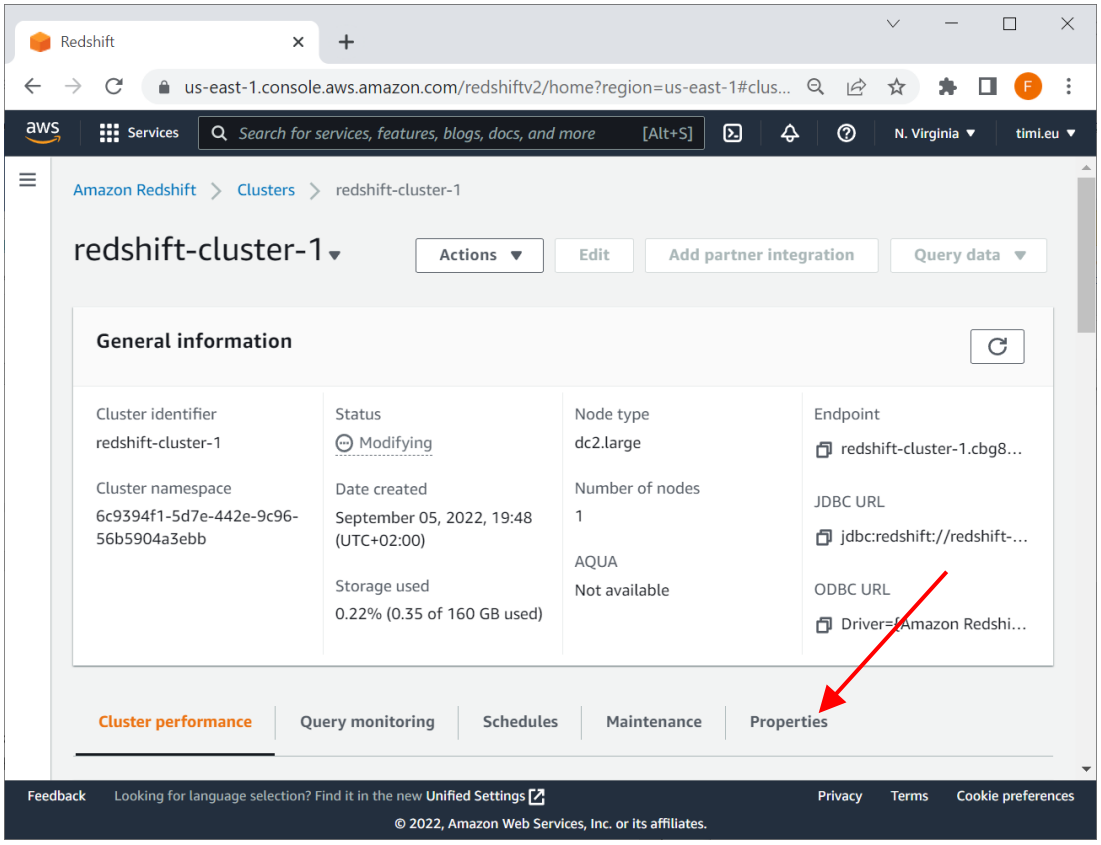


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

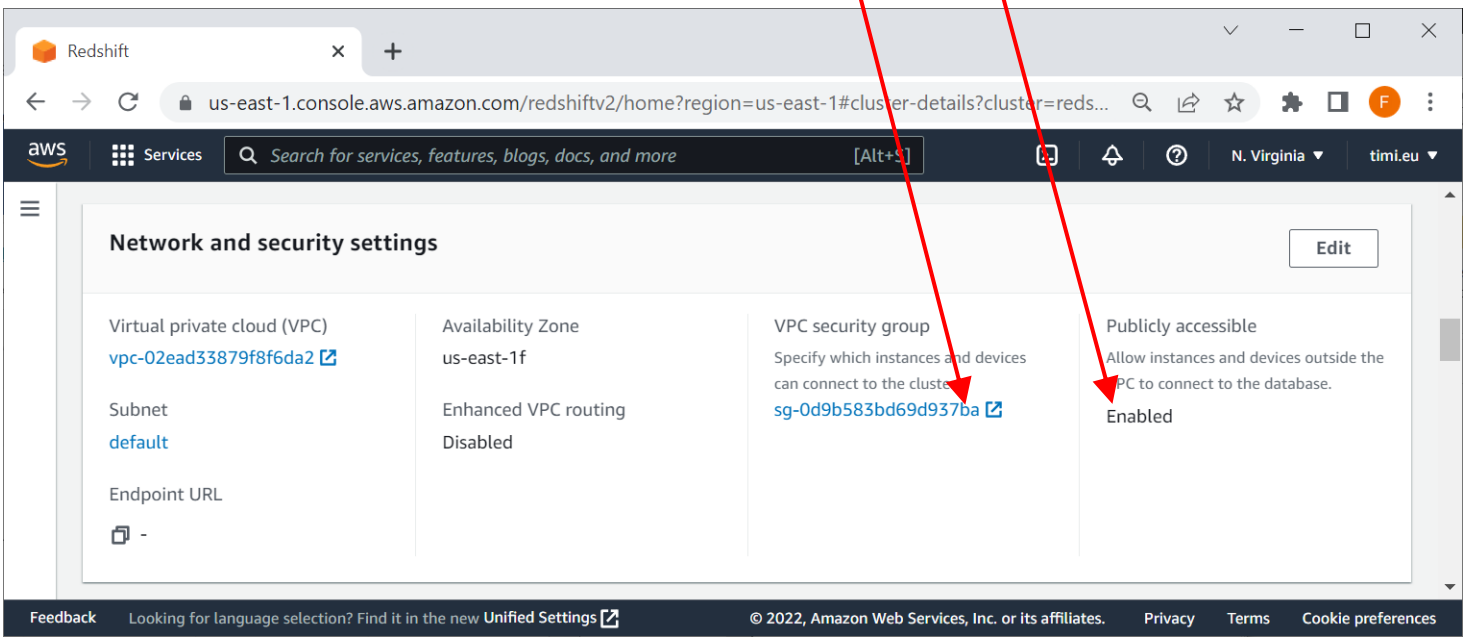
1.7. You should now see the “Global dashboard” related to your Redshift instance.

2. Open the firewall to access Redshift

2.1. Open the “Global dashboard” related to your Redshift instance (this should already be visible if you followed the previous steps) and click on the “Properties” tab:

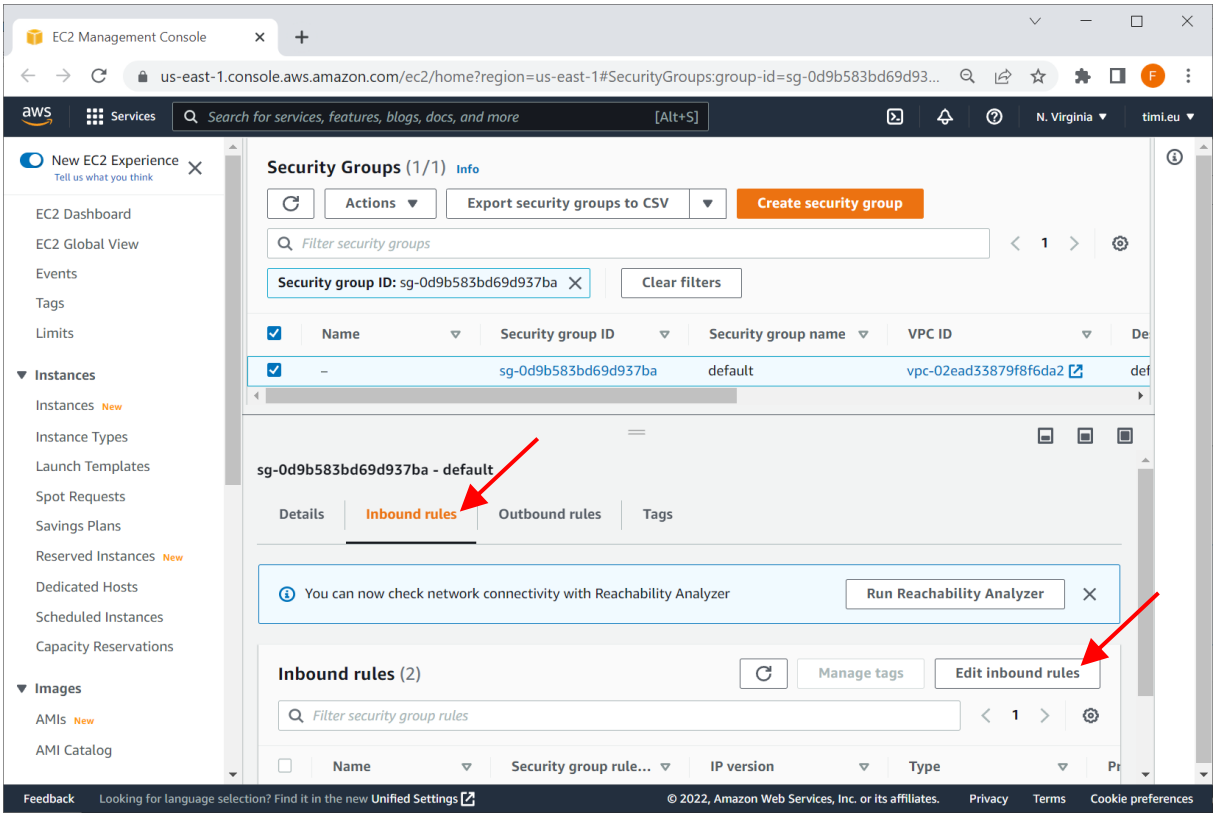


2.2. Scroll down to the “Network and security settings” section. Here, you should see that your Redshift is “Publicly accessible” (if that’s not the case, wait a little and refresh the webpage – press F5) : Check here:
Click on the link inside the “VPC security group”:

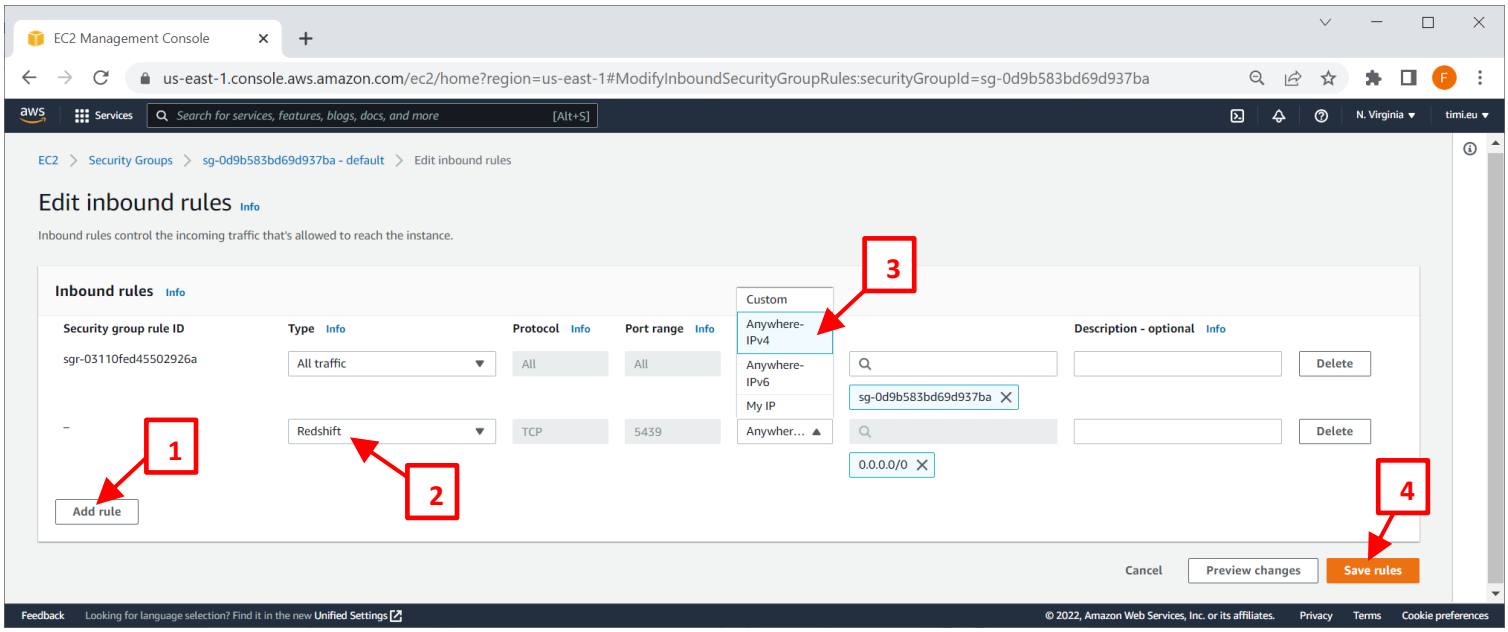


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2.3. Click on the “Inbound Rules” panel and click on the “Edit Inbound Rules” button:



2.4. Click on the “Add Rule” button: For the “Type” field, select “Redshift” and for the “Source” Field, select “Anywhere-IPv4”. Click the “Save Rules” button at the bottom.



2.5. You can now connect to Redshift wih Anatella from any machine.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.7. OleDB connections

Typically, you won't install simultaneously an ODBC driver and an OleDB drivers to connect to your database: it's either one or the other. If you have the choice, you should favor the installation and the usage of the ODBC drivers (rather than the OleDB drivers).

You should favor ODBC over OleDB because:

1. For almost all databases, ODBC connections are much faster (around 3 times to 20 times faster) than the equivalent OleDB connection.
2. OleDB drivers are not as common as ODBC drivers. This means that you'll have less choice when choosing a database backend.
3. OleDB drivers tend to be slower and with less options (For example, on February 2020, we know no OleDriver that supports the very important "DBPROP_IMultipleResults" option that allows to track the error status of "batch" operations).
4. The future of the whole OleDB technology is unclear: it may disappear soon.
   More details:

   Initially, the objective of OleDB was to supersede and to replace the older ODBC technique that was invented in 1991. When Microsoft introduced OleDB, Microsoft claimed that OleDB will run 100 times faster than ODBC in many common situations. Unfortunately, OleDB was a flop: Most of the database vendors ignored it and never produced any OleDB drivers. A few years later, Microsoft introduced ODBC v3 that is providing the same benefit as OleDB (i.e. fast INSERT speed and a database connection based on a simple Connection-String). ODBC v3 was (and still is) a great success. This success can be explained by the fact that ODBC v3 is retro-compatible with the old ODBC v1: i.e. If you have an application (such as Anatella) that is using ODBC v3, it can still access ODBC v1 drivers. This means that, with ODBC v3, you can access any tables stored in any database (whatever the ODBC version: v1,v2,v3).

   The final position of Microsoft with regard to OleDB is ambiguous: At one point (in 2011), Microsoft declared that OleDB is obsolete and deprecated, but a few years later, in 2018, Microsoft declared OleDB un-deprecated? Anyway, because of the lack of support from most database vendors and the unclear status about the future of the technology, OleDB is mostly a dead technology right now.

One advantage of OleDB over ODBC is that OleDB is not optimized to communicate with a database: it can connect to many data "provider" that are not necesseraly databases. In theory, this means that you could imagine to use an OleDB driver to connect to some very exotic "OleDB data source/provider" (Anatella supports such a use case by providing a large set of OleDB connection options). In practice, I never saw an OleDB drivers used reliably for something else than a database connection.

5.1.7.1. Microsoft SQL Server OleDB Setup

The OleDB driver for MS-SQLServer is not as optimized as the ODBC driver: it's around 3 times to 20 times slower than the ODBC driver.

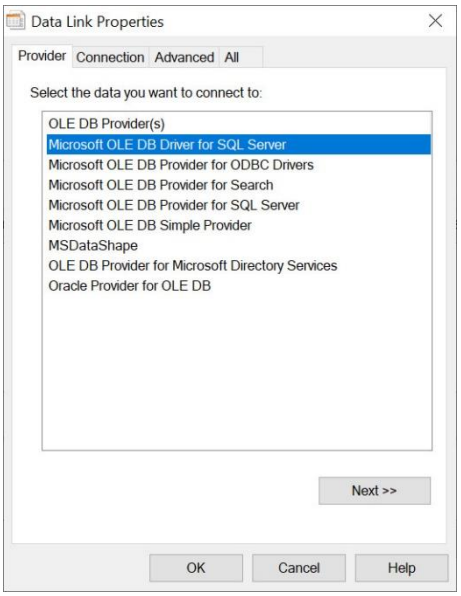
The OleDB drivers for MS-SQLServer are available for download on the Microsoft website here:
https://docs.microsoft.com/en-us/sql/connect/oledb/oledb-driver-for-sql-server
or, alternatively, here:
http://download.timi.eu/OleDB/OleDB_drivers_SQLServer/

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

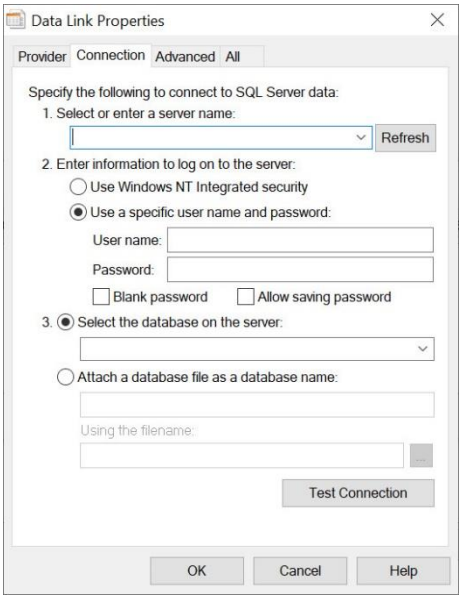
Win10 machines usually contain an old version of this OleDB driver which is now "deprecated" by Microsoft and replaced by a new version. You should use the new version. The differences between the new and the old version are:

OLD OleDB Driver (deprecated)

Named “MS OleDB *Driver* for SQLServer”:



Configuration panel:

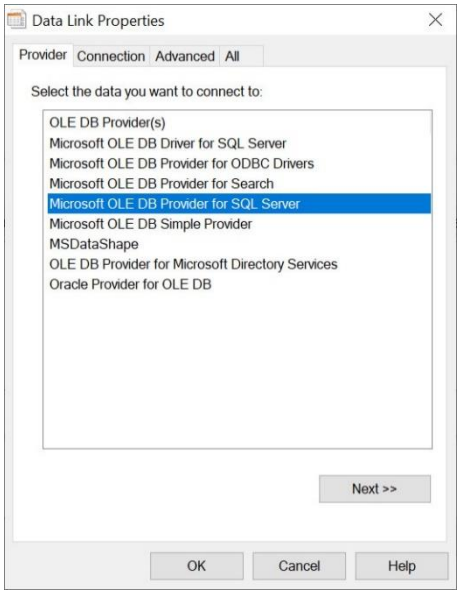


OleDB connection string starts with:

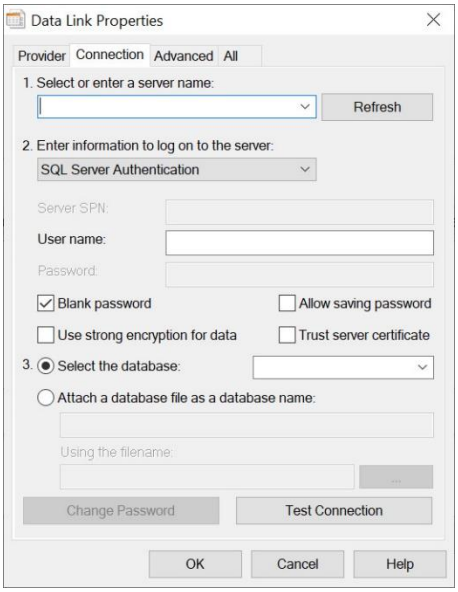
Provider=SQLOLEDB.1;

NEW OleDB Driver

Named “MS OleDB *Provider* for SQLServer”:



Configuration panel:

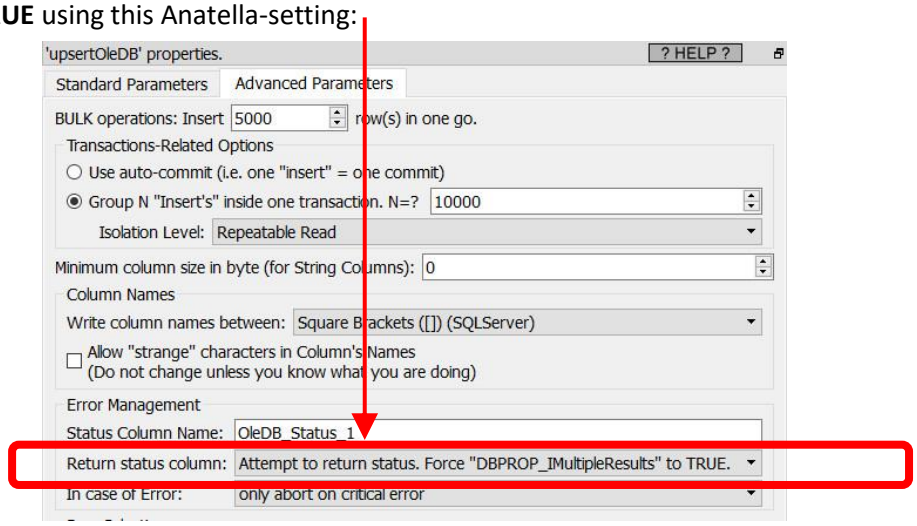


OleDB connection string starts with:

Provider=MSSQLOLEDB.1;

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

As of February 2021, the documentation of the MS-SQLServer OleDB driver claims that it supports the option "DBPROP_IMultipleResults=**TRUE**". By default, the value of this connection-option is **FALSE**. Unfortunately, the connection to the database silently fails when you attempt to set this connection-option to **TRUE** using this Anatella-setting:

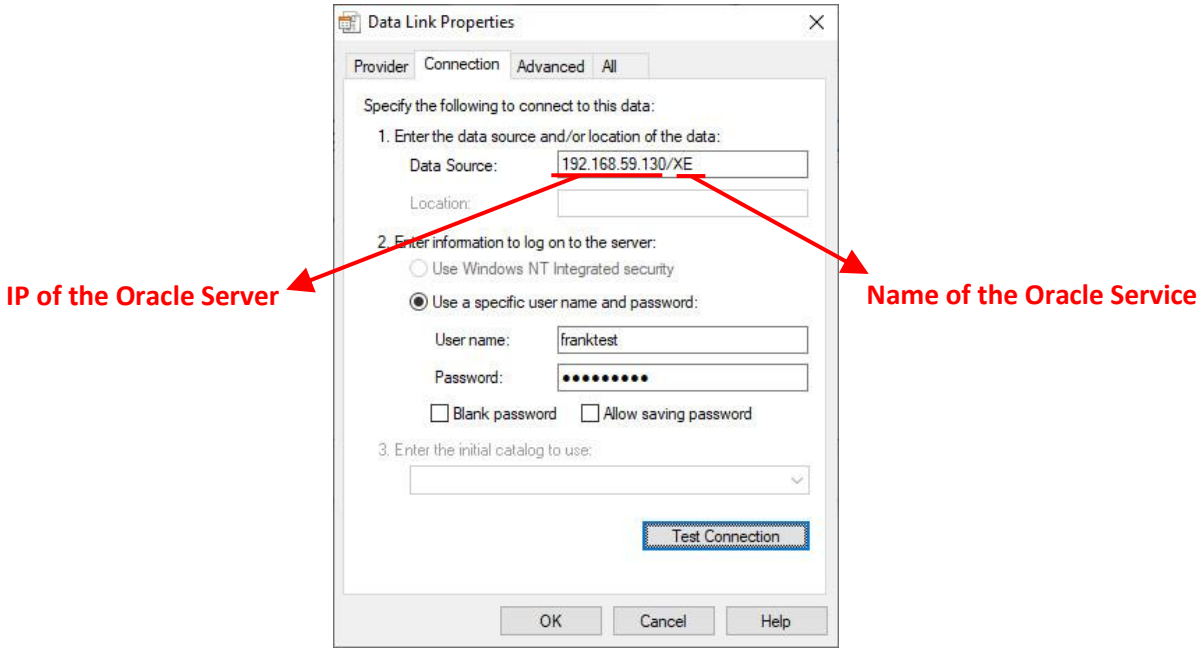


5.1.7.2. Oracle OleDB Setup

The OleDB driver for Oracle is not as optimized as the ODBC driver: it's around 3 times to 20 times slower than the ODBC driver.

The OleDB drivers for Oracle are available for download on the Oracle website here:
https://www.oracle.com/database/technologies/odac-downloads.html
or, alternatively, here:
http://download.timi.eu/OleDB/OleDB_drivers_Oracle/

Here is an example of OleDB-Oracle-setup-screen that allows you to get your connection-string:

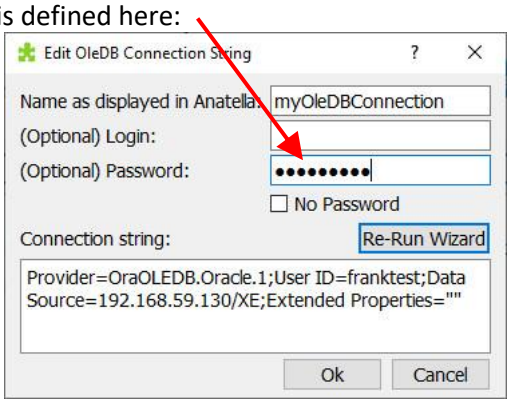


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The default Oracle OleDB connection string looks like this:

Provider=OraOLEDB.Oracle.1;Persist Security Info=False;User ID=<*your oracle login*>;Data Source=<*your oracle server ip*>/<*your oracle service*>;Extended Properties=""

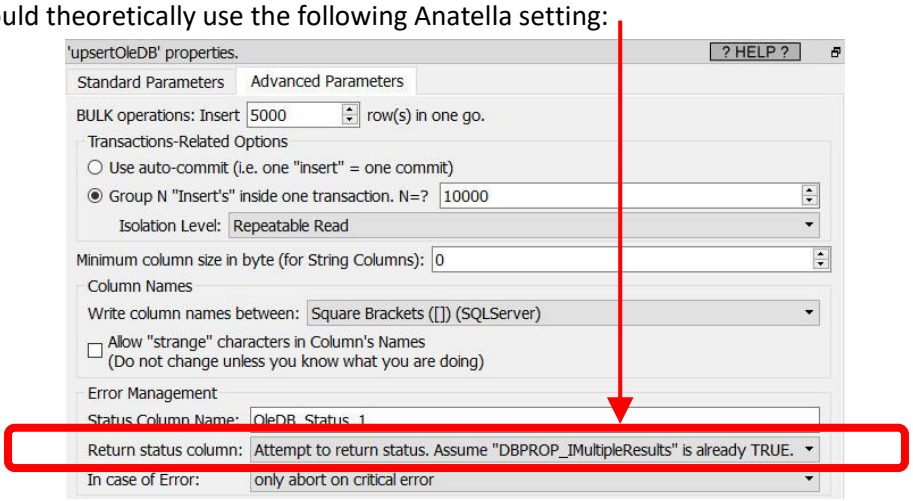
If you connect to Oracle using a password that is defined here:



…then you must manually remove from the OleDB-connection-string the parameter "*Persist Security Info=False;*" so that your OleDB-connection-string becomes:

Provider=OraOLEDB.Oracle.1;User ID=<*your oracle login*>;Data Source=<*your oracle server ip address*>/<*your oracle service*>

As of February 2021, the documentation of the Oracle OleDB driver claims that it supports the option "DBPROP_IMultipleResults=**TRUE**". By default, the value of this connection-option is already **TRUE**, so that we could theoretically use the following Anatella setting:



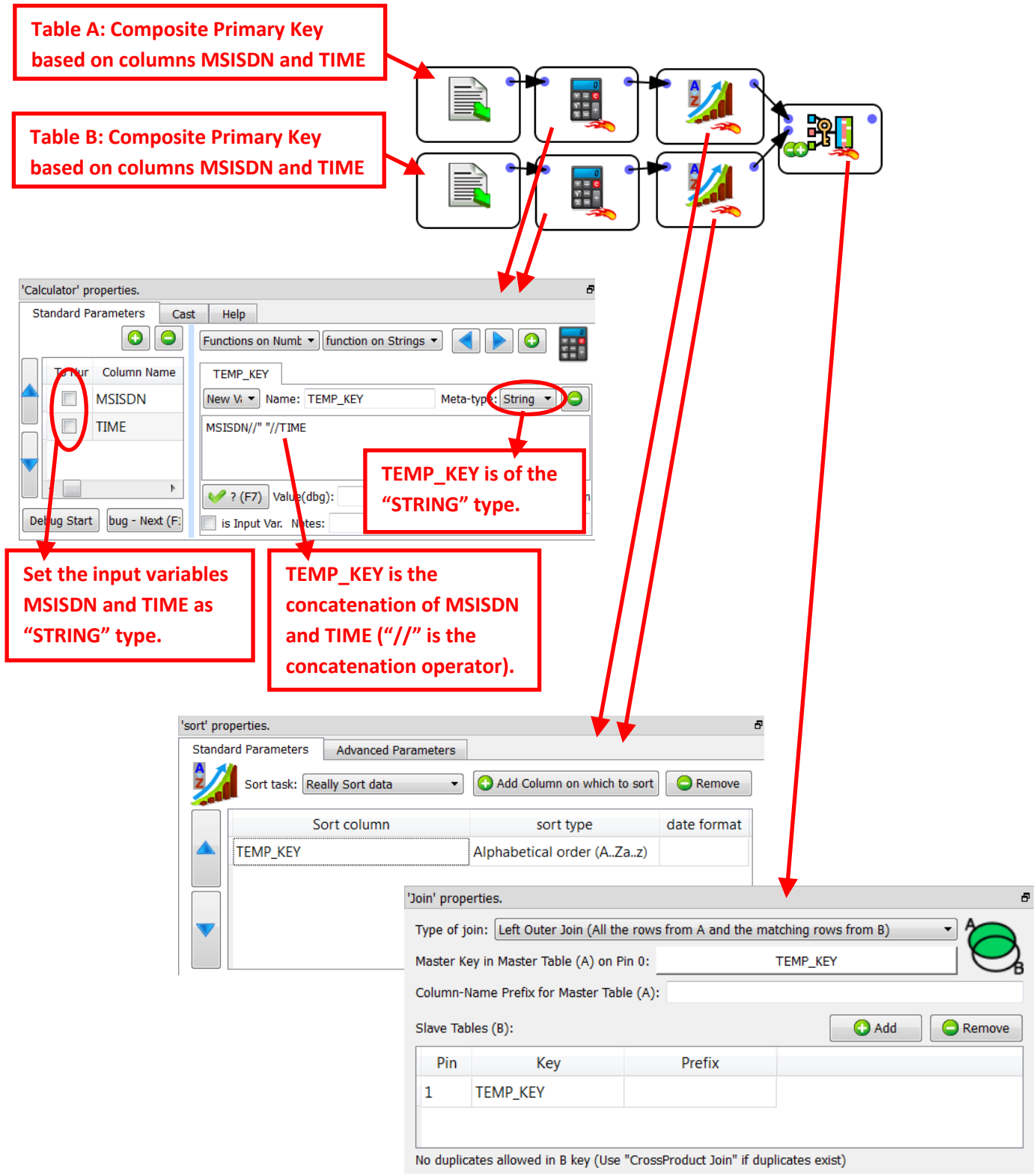
Unfortunately, during our tests we noticed that the "Status" column returned by the "upsertOleDBAction" some contains the string "NO_STATUS": This means that the "DBPROP_IMultipleResults=**TRUE**" option is not supported by the Oracle OleDB driver and all the returned status indicators are invalid.

5.1.8. Composite Primary Keys

When you are computing a join between several columns, sometime, the primary key is composed of the several columns (instead of, classically, ONE column). This is called a COMPOSITE primary key.
Most Anatella actions only support "normal" primary keys (i.e. primary keys that are defined using one column). Anatella can handle composite primary keys but you need an extra step: you must create one temporary new column that is the concatenation of all the columns inside the COMPOSITE primary key (and use this new, temporary column instead of the columns inside the COMPOSITE primary key).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example: Let's assume that we want to compute a join between two tables: table A and table B. The COMPOSITE primary key inside table A is defined using the columns MSISDN and TIME (and the same inside table B). We will have:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.1.9. Internet access

Inside Anatella, most of the actions that are connecting to a remote cloud service are actually using a small utily named "cURL" to manage the internet connection.

As an example, the following actions are using cURL to connect to the internet:

The DownloadAndUpload action (see section 5.22.1), The Multiple DownloadAndUpload action (see section 5.22.2), The generic REST API call action (see section 5.22.3), The VAT check action (see section 5.22.4), The Tableau Publish action (see section 5.22.5), The S3ListFilesInBucket action (see section 5.22.6), The S3DownloadFile action (see section 5.22.8), The S3UploadFile action (see section 5.22.9).

Using the "cURL" tool to access internet services has several advantages:

- cURL supports a wide range of internet protols directly "out of the box".

The "cURL" engine (currently included inside Anatella) supports the following protocols:

curl 7.60.0 (i386-pc-win32) libcurl/7.60.0 OpenSSL/1.1.0h (WinSSL) zlib/1.2.11 brotli/1.0.4 WinIDN libssh2/1.8.0 nghttp2/1.32.0

Release-Date: 2018-05-16

Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtsp scp sftp smb smbs smtp smtps telnet tftp

Features: AsynchDNS IDN IPv6 Largefile SSPI Kerberos SPNEGO NTLM SSL libz brotli TLS-SRP HTTP2 proxy HTTPS-proxy MultiSSL

- cURL support internet access through a proxy server.

You'll find more documentation about cURL here: https://curl.haxx.se/docs/manpage.html

5.1.9.1. List of cURL error codes and their explanation

As an output of each internet connection through cURL, you get an "Error Code" that give some explanation about the success/failure of the connection. The error code "zero" means: No error. The Other Error codes are (This is an extract from https://curl.haxx.se/libcurl/c/libcurl-errors.html):

ERROR CODE	SHORT DESCRIPTION	LONG DESCRIPTION
0	OK	All fine. Proceed as usual.
1	UNSUPPORTED_PROTOCOL	The URL you passed to libcurl used a protocol that this "curl.exe" does not support. it can be a misspelled protocol string or just a protocol that curl has no code for.
2	FAILED_INIT	Very early initialization code failed. This is likely to be an internal error or problem, or a resource problem where something fundamental couldn't get done at init time.
3	URL_MALFORMAT	The URL was not properly formatted.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4	NOT_BUILT_IN	A requested feature, protocol or option was not found built-in in this curl.exe. This means that a feature or option was not enabled or explicitly disabled when curl.exe was built and in order to get it to function you have to get a download another “curl.exe” executable.
5	COULDNT_RESOLVE_PROXY	Couldn't resolve proxy. The given proxy host could not be resolved.
6	COULDNT_RESOLVE_HOST	Couldn't resolve host. The given remote host was not resolved.
7	COULDNT_CONNECT	Failed to connect() to host or proxy.
8	FTP_WEIRD_SERVER_REPLY	The server sent data libcurl couldn't parse. This error code is used for more than just FTP and is aliased as CURLE_WEIRD_SERVER_REPLY since 7.51.0.
9	REMOTE_ACCESS_DENIED	We were denied access to the resource given in the URL. For FTP, this occurs while trying to change to the remote directory.
10	FTP_ACCEPT_FAILED	While waiting for the server to connect back when an active FTP session is used, an error code was sent over the control connection or similar.
11	FTP_WEIRD_PASS_REPLY	After having sent the FTP password to the server, libcurl expects a proper reply. This error code indicates that an unexpected code was returned.
12	FTP_ACCEPT_TIMEOUT	During an active FTP session while waiting for the server to connect, the CURLOPT_ACCEPTTIMEOUT_MS (or the internal default) timeout expired.
13	FTP_WEIRD_PASV_REPLY	libcurl failed to get a sensible result back from the server as a response to either a PASV or a EPSV command. The server is flawed.
14	FTP_WEIRD_227_FORMAT	FTP servers return a 227-line as a response to a PASV command. If libcurl fails to parse that line, this return code is passed back.
15	FTP_CANT_GET_HOST	An internal failure to lookup the host used for the new connection.
16	HTTP2	A problem was detected in the HTTP2 framing layer. This is somewhat generic and can be one out of several problems, see the error buffer for details.
17	FTP_COULDNT_SET_TYPE	Received an error when trying to set the transfer mode to binary or ASCII.
18	PARTIAL_FILE	A file transfer was shorter or larger than expected. This happens when the server first reports an expected transfer size, and then delivers data that doesn't match the previously given size.
19	FTP_COULDNT_RETR_FILE	This was either a weird reply to a 'RETR' command or a zero byte transfer complete.
21	QUOTE_ERROR	When sending custom "QUOTE" commands to the remote server, one of the commands returned an error code that was 400 or higher (for FTP) or otherwise indicated unsuccessful completion of the command.
22	HTTP_RETURNED_ERROR	This is returned if CURLOPT_FAILONERROR is set TRUE and the HTTP server returns an error code that is >= 400.
23	WRITE_ERROR	An error occurred when writing received data to a local file, or an error was returned to libcurl from a write callback.
25	UPLOAD_FAILED	Failed starting the upload. For FTP, the server typically denied the STOR command. The error buffer usually contains the server's explanation for this.
26	READ_ERROR	There was a problem reading a local file or an error returned by the read callback.
27	OUT_OF_MEMORY	A memory allocation request failed. This is serious badness and things are severely screwed up if this ever occurs.
28	OPERATION_TIMEDOUT	Operation timeout. The specified time-out period was reached according to the conditions.
30	FTP_PORT_FAILED	The FTP PORT command returned error. This mostly happens when you haven't specified a good enough address for libcurl to use. See CURLOPT_FTPPORT.
31	FTP_COULDNT_USE_REST	The FTP REST command returned error. This should never happen if the server is sane.
33	RANGE_ERROR	The server does not support or accept range requests.
34	HTTP_POST_ERROR	This is an odd error that mainly occurs due to internal confusion.
35	SSL_CONNECT_ERROR	A problem occurred somewhere in the SSL/TLS handshake. You really want the error buffer and read the message there as it pinpoints the problem slightly more. Could be certificates (file formats, paths, permissions), passwords, and others.
36	BAD_DOWNLOAD_RESUME	The download could not be resumed because the specified offset was out of the file boundary.
37	FILE_COULDNT_READ_FILE	A file given with FILE:// couldn't be opened. Most likely because the file path doesn't identify an existing file. Did you check file permissions?

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

38	LDAP_CANNOT_BIND	LDAP cannot bind. LDAP bind operation failed.
39	LDAP_SEARCH_FAILED	LDAP search failed.
41	FUNCTION_NOT_FOUND	Function not found. A required zlib function was not found.
42	ABORTED_BY_CALLBACK	Aborted by callback. A callback returned "abort" to libcurl.
43	BAD_FUNCTION_ARGUMENT	Internal error. A function was called with a bad parameter.
45	INTERFACE_FAILED	Interface error. A specified outgoing interface could not be used. Set which interface to use for outgoing connections' source IP address with CURLOPT_INTERFACE.
47	TOO_MANY_REDIRECTS	Too many redirects. When following redirects, libcurl hit the maximum amount. Set your limit with CURLOPT_MAXREDIRS.
48	UNKNOWN_OPTION	An option passed to libcurl is not recognized/known. Refer to the appropriate documentation. This is most likely a problem in the program that uses libcurl. The error buffer might contain more specific information about which exact option it concerns.
49	TELNET_OPTION_SYNTAX	A telnet option string was Illegally formatted.
51	PEER_FAILED_VERIFICATION	The remote server's SSL certificate or SSH md5 fingerprint was deemed not OK.
52	GOT_NOTHING	Nothing was returned from the server, and under the circumstances, getting nothing is considered an error.
53	SSL_ENGINE_NOTFOUND	The specified crypto engine wasn't found.
54	SSL_ENGINE_SETFAILED	Failed setting the selected SSL crypto engine as default!
55	SEND_ERROR	Failed sending network data.
56	RECV_ERROR	Failure with receiving network data.
58	SSL_CERTPROBLEM	problem with the local client certificate.
59	SSL_CIPHER	Couldn't use specified cipher.
60	SSL_CACERT	Peer certificate cannot be authenticated with known CA certificates.
61	BAD_CONTENT_ENCODING	Unrecognized transfer encoding.
62	LDAP_INVALID_URL	Invalid LDAP URL.
63	FILESIZE_EXCEEDED	Maximum file size exceeded.
64	USE_SSL_FAILED	Requested FTP SSL level failed.
65	SEND_FAIL_REWIND	When doing a send operation curl had to rewind the data to retransmit, but the rewinding operation failed.
66	SSL_ENGINE_INITFAILED	Initiating the SSL Engine failed.
67	LOGIN_DENIED	The remote server denied curl to login (Added in 7.13.1)
68	TFTP_NOTFOUND	File not found on TFTP server.
69	TFTP_PERM	Permission problem on TFTP server.
70	REMOTE_DISK_FULL	Out of disk space on the server.
71	TFTP_ILLEGAL	Illegal TFTP operation.
72	TFTP_UNKNOWNID	Unknown TFTP transfer ID.
73	REMOTE_FILE_EXISTS	File already exists and will not be overwritten.
74	TFTP_NOSUCHUSER	This error should never be returned by a properly functioning TFTP server.
75	CONV_FAILED	Character conversion failed.
76	CONV_REQD	Caller must register conversion callbacks.
77	SSL_CACERT_BADFILE	Problem with reading the SSL CA cert (path? access rights?)
78	REMOTE_FILE_NOT_FOUND	The resource referenced in the URL does not exist.
79	SSH	An unspecified error occurred during the SSH session.
80	SSL_SHUTDOWN_FAILED	Failed to shut down the SSL connection.
81	AGAIN	Socket is not ready for send/recv wait till it's ready and try again. This return code is only returned from curl_easy_recv and curl_easy_send (Added in 7.18.2)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

82	SSL_CRL_BADFILE	Failed to load CRL file (Added in 7.19.0)
83	SSL_ISSUER_ERROR	Issuer check failed (Added in 7.19.0)
84	FTP_PRET_FAILED	The FTP server does not understand the PRET command at all or does not support the given argument. Be careful when using CURLOPT_CUSTOMREQUEST, a custom LIST command will be sent with PRET CMD before PASV as well. (Added in 7.20.0)
85	RTSP_CSEQ_ERROR	Mismatch of RTSP CSeq numbers.
86	RTSP_SESSION_ERROR	Mismatch of RTSP Session Identifiers.
87	FTP_BAD_FILE_LIST	Unable to parse FTP file list (during FTP wildcard downloading).
88	CHUNK_FAILED	Chunk callback reported error.
89	NO_CONNECTION_AVAILABLE	(For internal use only, will never be returned by libcurl) No connection available, the session will be queued. (added in 7.30.0)
90	SSL_PINNEDPUBKEYNOTMATCH	Failed to match the pinned key specified with CURLOPT_PINNEDPUBLICKEY.
91	SSL_INVALIDCERTSTATUS	Status returned failure when asked with CURLOPT_SSL_VERIFYSTATUS.
92	HTTP2_STREAM	Stream error in the HTTP/2 framing layer.

5.1.9.2. Internet access through a PROXY

The "cURL" engine (i.e. the engine used inside Anatella to do nearly all internet connections) also supports internet access through a PROXY server.

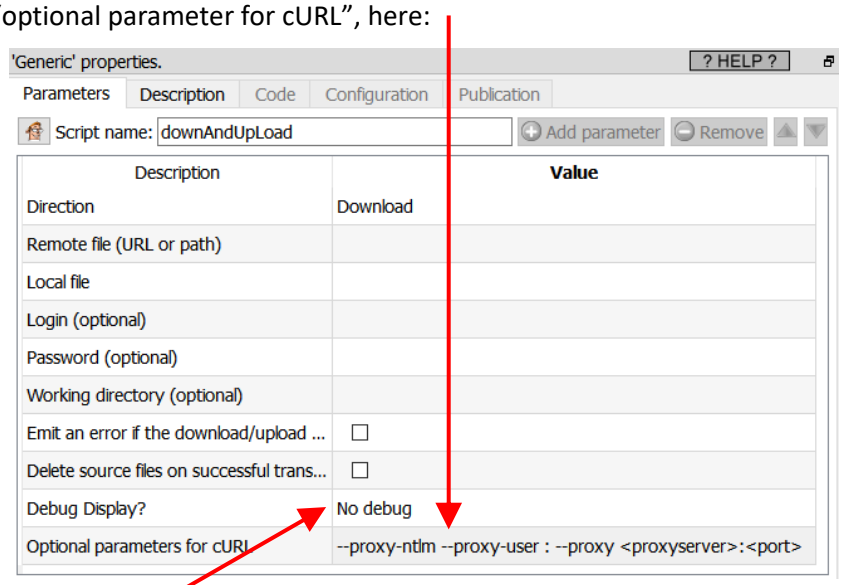
Most of the time, the PROXY server wants to authenticate you using your MS-Windows login session: In this common situation, cURL must use you MS-Windows credentials to login into the PROXY server. To use this type of authentification mechanism, you'll use the following cURL optional parameters:

```
--proxy-ntlm --proxy-user : --proxy <proxyserver>:<port>
```

Another, less common, mechanism to connect to the PROXY server is to use the following cURL optional parameters:

```
--proxy --proxy-user <login>:<pass> --proxy <proxyserver>:<port>
```

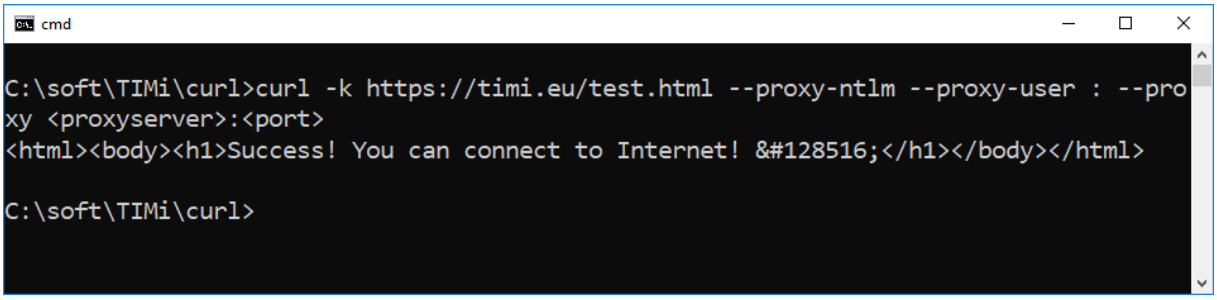
You enter the "optional parameter for cURL", here:



To test/debug if your parameters are correct, you can change the "Debug Display" option to the "Verbose" mode, here

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

An easier, alternative way to test if your (PROXY-related) cURL parameters are correct is to use cURL inside a command-line prompt (i.e. press [Win]+[R] and run “cmd”): For example:



You’ll find more documentation about the different possible parameters related to PROXY servers inside cURL (e.g. "--proxy-ntlm","--proxy-user","--proxy", “--proxy-anyauth”, “--negotiate”), here:
https://curl.haxx.se/docs/manpage.html

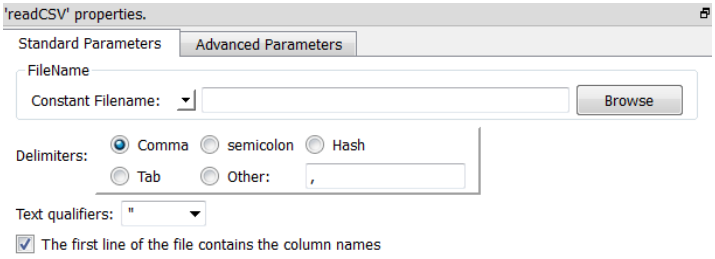
To see the exact proxy settings that are currently used on your machine, you can type in a DOS/Shell command line: `netsh winhttp show proxy`

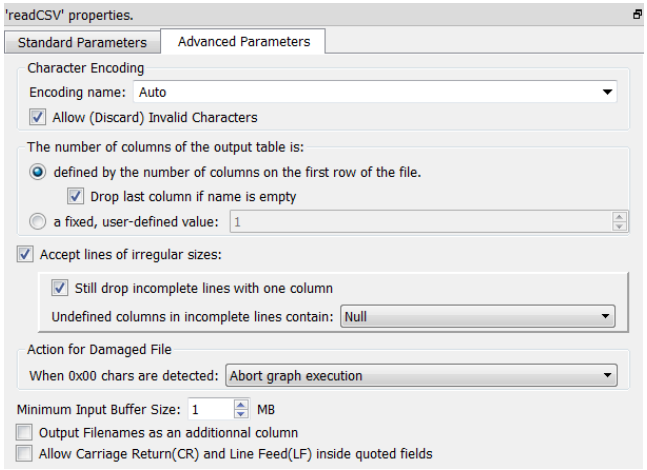
5.2. Input Actions

5.2.1. CSV file reader

Icon:

Property window:





Short description:
Reads a Text/CSV file

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the Text/CSV file (i.e. you can use relative path, wildcards, and Javascript to specify your filename).

When reading a Text/CSV file, the first operation that Anatella does is to decode the characters contained inside your Text/CSV file to obtain Unicode characters. Anatella supports many different character encodings. To decode characters, Anatella uses the most extensive library about character encodings currently available (i.e. it uses the “iconv” library).

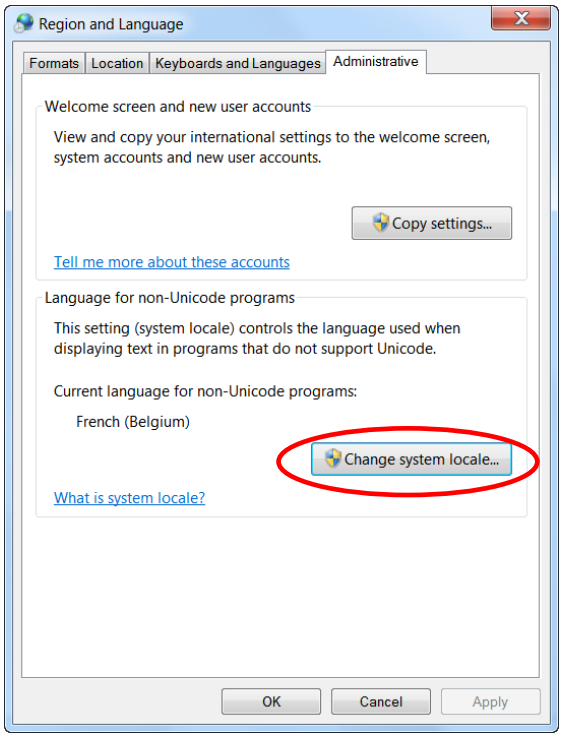
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The supported character encodings are (this is a **non-limitative** list):
UTF16, UTF16LE, UTF16BE, UTF8, CP1252, CP819 (aka ISO-8859-1 or LATIN1), SHIFT-JIS, BIG5, GBK, CP1251 (Cyrillic), JAVA, etc.
Anatella currently supports nearly all known encodings (even the most exotic ones found inside very old servers).

If the Text/CSV file contains a BOM (Byte-Order-Mark), then Anatella will always use the character encoding that is specified inside the BOM (this takes precedence over any other user's settings, including the "Auto" character encoding setting).

When the "Auto" character encoding is selected (i.e. this is the default choice), Anatella uses the "Local" character encoding that is defined inside the MSWindows operating system. Under MSWindows, the local/currently active "Character Encoding" is named the "Code Page" (the "Code page" is a common synonym for "Character Encoding"): More information about this subject here:
http://en.wikipedia.org/wiki/Windows_code_page

To change the active "Code Page" under MSWindows: Open the "Region and Language" settings inside the "Control Panel", go to the "Administrative" panel and click on the "Change system locale" button: Here is a screenshot:



In opposition to many other ETL tools, all the strings inside Anatella are handled in "true" UTF-16 format. This "true" UTF-16 support guarantees you complete conformance to the universal standard in string manipulations routines (for case insensitive sort, for example).

If the extension of the Text/CSV file is RAR, ZIP, GZ or LZO then Anatella will transparently decompress the file in memory. Anatella chooses the (de)compression technique to use based on the filename extension. When Anatella uses compressed file formats, it does NOT decompress the files on the Hard Drive: Anatella decompresses the data "on-the-fly" in central core RAM memory, thus reducing:

- the load on the hard drive
- the hard-drive consumption required to do the analysis.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Usually, for classical "real world" databases, the compression/ratio of CSV file is around 90-95%. For example, the "classical Census-Income database" is originally 100MB and after compression (using WinRar) it's only 4MB.

The ability to natively read compressed files is important when you are working on a distributed file system, with a central network drive shared by all the Anatella users. For example, when Anatella reads the "classical Census-Income database", there will be only 4MB of data that goes "through the network cables", instead of 100MB for another ETL that is not able to work on compressed files. Thus, Anatella is the only ETL tool that reduces substantially the load on your computer network.

Using a central repository on one shared drive is a good idea to prevent duplication of the data (to have only one version of the "truth"). You should avoid data-duplication because, if the same (supposedly) data is present on different location, there will always be a moment where all the different duplicated copies will be "out of synchronization" (i.e. different). If several analysts are working on different "out of synch." Data (i.e. different version of the "truth"), then they might arrive to different opposing conclusions, giving you (in the worst case) contradictory advices about your business.

To summarize:

1. Anatella is one of the few ETL that *fully* supports standard Unicode characters.
2. Keeping only one central data repository ensures consistency between the different analyzes made by your team.
3. The unique compression technology of Anatella allows you to substantially reduce the load on your computer network and thus, to easily work with one central data repository.

5.2.1.1. Reading corrupted CSV/Text files.

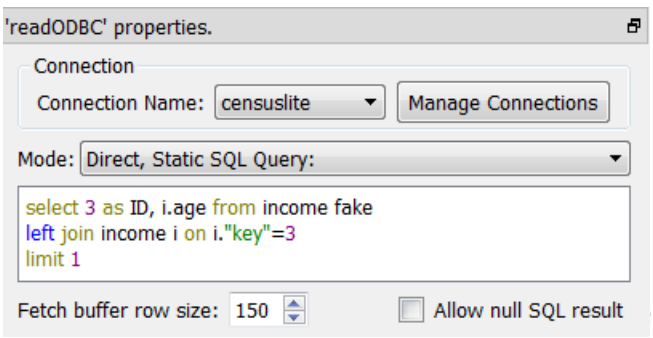
When transferring large .csv/.txt files (e.g. 100 GB text files) over an unrealiable computer network, it might sometime happen that some bytes of the file did not arrived correctly at the destination. In such situation, you will have a text file with a "hole" in it. The "hole" is typically composed of the ascii-character number zero (this character is also sometime referred as the "unicode code point zero") and it's usually 1 or 2 MB long. The same type of "holes" are also sometime happening when storing text files on some poor-quality USB keys. When you read such "damaged/corrupted file", you have four options:

- Abort the data-transformation graph if the ascii-character number zero is found inside the text file (this is the default option).
- Skip all the rows that contains the ascii-character number zero: This allows to easily "jump" above the large "holes" inside the text files, to only keep the correct rows. This option is very useful to still read text files corrupted by a bad network transfer or a bad USB key.
- Remove all the ascii-character number zero from the file and proceed as usual (this last option does not discard any row or column). This typically allows handling of erroneous text files produced by SAP.
- Replace the ascii-character number zero with a valid character (e.g. the "space" character).

5.2.2. Generic ODBC reader

Icon:

Property window:



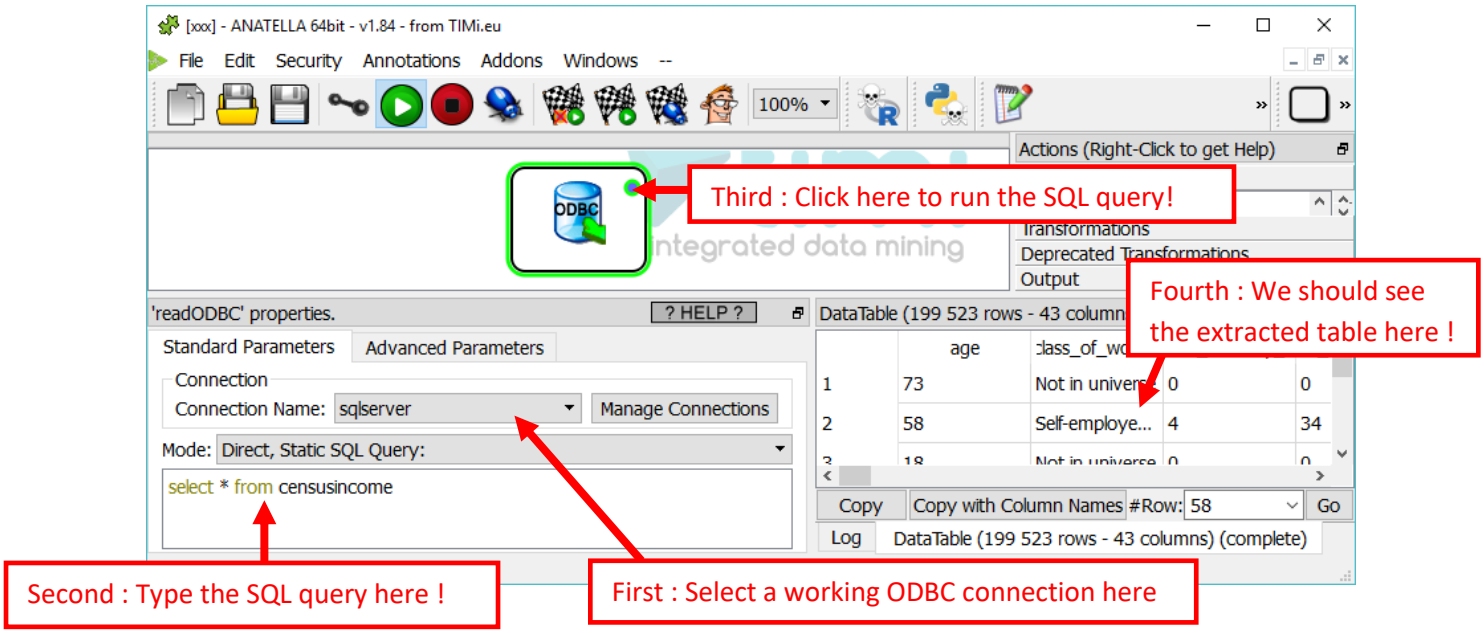
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Reads a table from an ODBC datasource.
Launch the execution any free-form SQL statement(s).

Long Description:
ODBC is a generic technique to access the content of almost any relational databases.
ODBC is the oldest technique available to access data contained into relational databases.
All the databases have an ODBC driver.

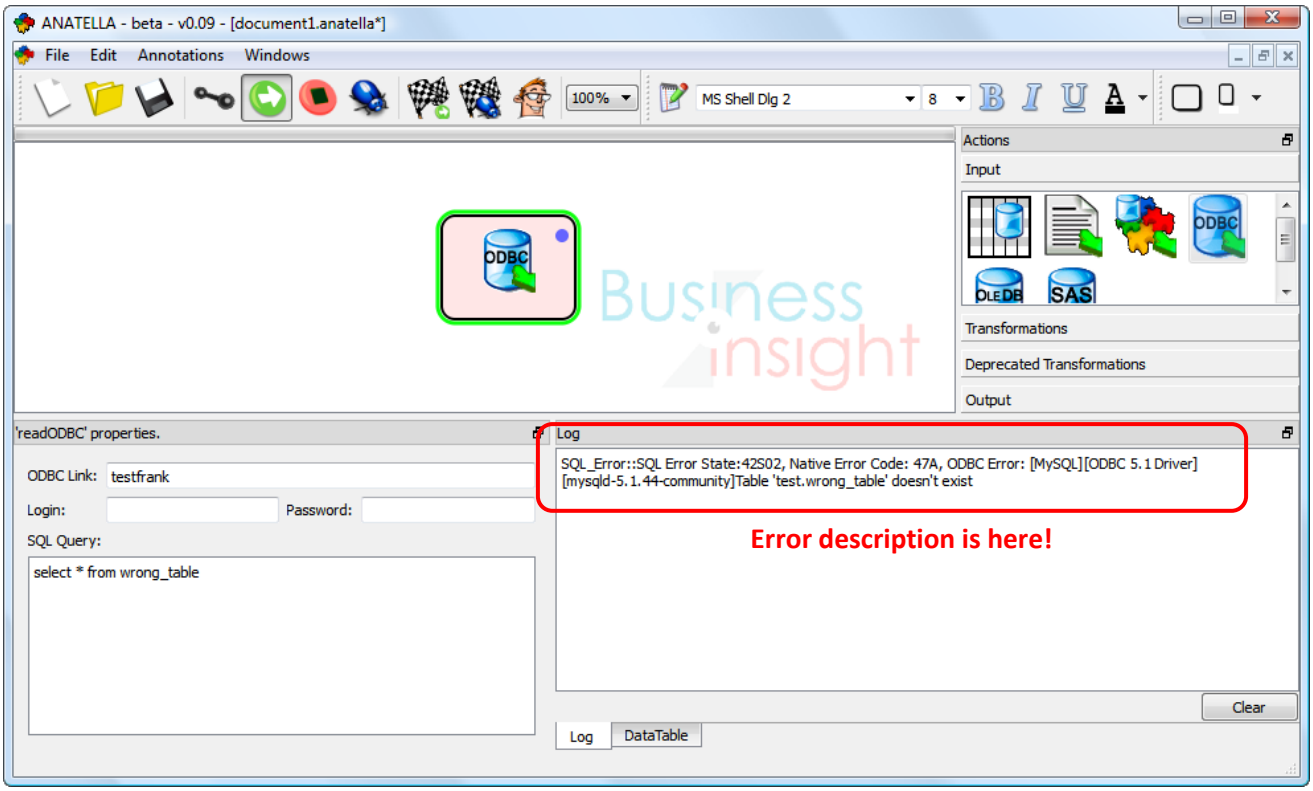
Before executing the

ReadODBC Action, you must first define an ODBC connection: See the section 5.1.6. to know how to create an ODBC connection from Anatella to your database. Once you have a working ODBC connection, you can directly execute any SQL command: For example: Let's extract the table "censusincome" from your database:



What happens if your SQL query has some error in it? Don't worry! Anatella will give you a complete description of the error inside the "Log window":

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



The ODBC reader Action submits your SQL statments to the database and, then, waits for the output rows that the database can (optionally) return (some SQL statements do not return any results). This means that, in particular:

1. If you are experiencing very long running time, it means that the database engine has difficulties processing your SQL statement (…and Anatella is innocent! ☺). Try simplifying your SQL statements (adding "top 100" usually helps). The queries including JOIN's are usually too CPU-intensive (on large tables) and, very often, it's faster to compute the joins with Anatella.

2. Some database engines have strong limitations on the number of tables that can be included inside a JOIN statement (e.g. Access2010 is limited to 5 tables, so it's best to extract one table at a time and compute the JOIN with Anatella). If the SQL statement fails because of some limitations of the database engine, Anatella will display an ODBC error message explaining the nature of the error, allowing you to track and resolve the error.

3. The ODBC error messages are produced by your database. Please, refer to your database documentation for more information about the reported errors.

There are 6 operating modes for the

ReadODBC Action:

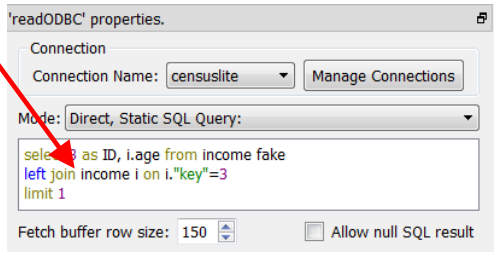
1. Direct, Static SQL Query
2. Direct, Static SQL Query (each statement executed in its own query)
3. SQL Queries computed using JavaScript code
4. SQL Queries from Input Pin
5. SQL Queries from Input Pin + Input Pin Data
6. SQL Queries from Input Pin + Input Pin Data ; Returns "_Status" only

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

These 6 operating modes are described in the remainder of this section.

Operating Mode 1: Direct, Static SQL Query

This text box can include many different SQL statements (the statements are typically separated from each other with a ";" character):

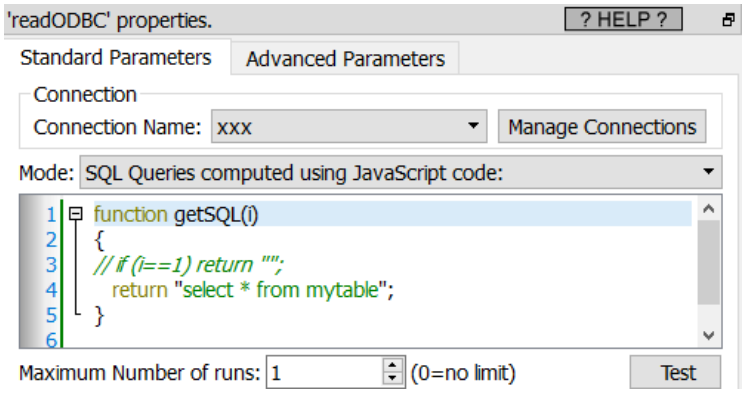


In "Operating Mode 1", the whole content of the text box is sent in "one go" to the ODBC driver for execution. Unfortunately, some databases (e.g. Teradata) are limited: i.e. They cannot run more than one SQL statement at-a-time. This means that, for example, if your text box contains two SQL statements, the Teradata database will return an error and the graph execution will stop. To bypass this limitation of the database, there now exists an "Operating Mode 2" inside Anatella.

Operating Mode 2: Direct, Static SQL Query (each statement executed in its own query)

When using "Operating Mode 2", Anatella splits the content of the text box in many different SQL statements (currently the "split procedure" is primitive: Anatella just splits the text box simply looking for the ";" character). Each SQL statement is then sent one-by-one to the ODBC driver. This allows to bypass the limitation of some ODBC drivers that are only able to execute one SQL statement for each string that they receive.

Operating Mode 3: SQL Queries computed using JavaScript code



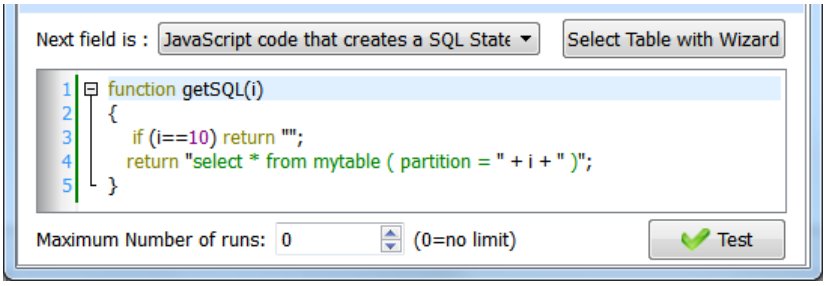
This third operating mode allows you to execute SQL statements that are generated using a given JavaScript code. This has several benefits over simply writing directly a "constant" SQL statement:

1. You can use inside your JavaScript code some Global Variables. This allows you to parameterize your data-transformation-graph (see section 5.1.5 about Global Variables).
2. Sometime, it's easier to write the (JavaScript) code that generates a very complex and long SQL statement than to try to write directly the statement.
3. When your SQL statement is a union of many different statements, it's easier to use a JavaScript code.
   For example, instead of writing:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
select * from mytable (partition=0)
union all
select * from mytable (partition=1)
union all
select * from mytable (partition=2)
union all
select * from mytable (partition=3)
union all
select * from mytable (partition=4)
union all
select * from mytable (partition=5)
union all
select * from mytable (partition=6)
union all
select * from mytable (partition=7)
union all
select * from mytable (partition=8)
union all
select * from mytable (partition=9)
```

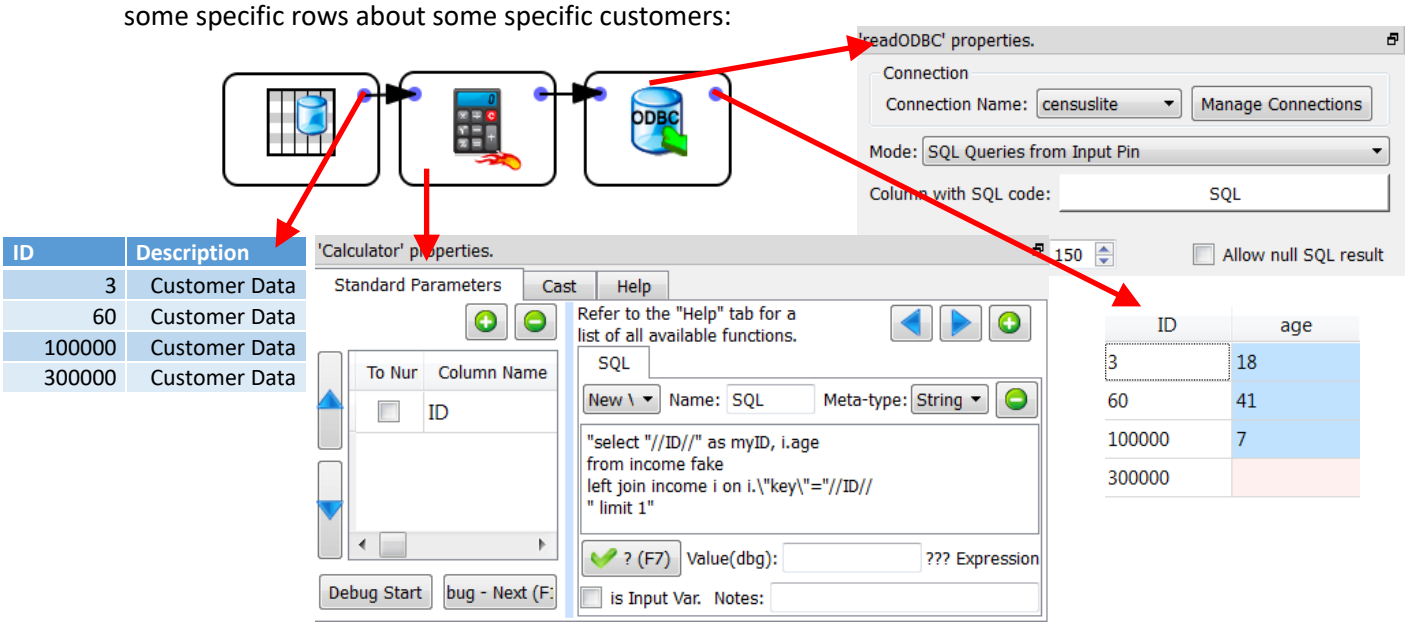
… you can simply write:



In this way, the "union" is performed by Anatella (and not by the database engine). This allows more flexibility because there are nearly no restrictions in Anatella on the "union" operation (as opposed to the union performed by the database).

Operating Mode 4: SQL Queries from Input Pin
Operating Mode 5: SQL Queries from Input Pin + Input Pin Data

For example, these last 2 modes are usefull when you want to extract from the relational database some specific rows about some specific customers:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Operating Mode 6: SQL Queries from Input Pin + Input Pin Data ; Returns "_Status" only

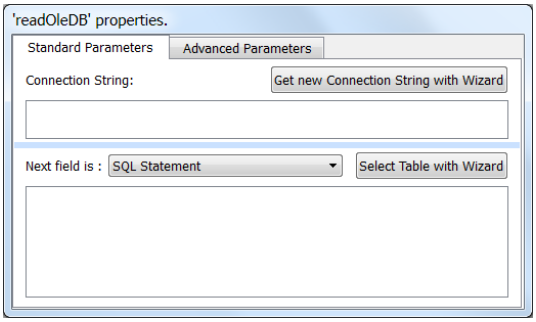
This mode forwards the input table to the output pin, execute the SQL command given in input and adds a new column named "_status". The content of the column "_status" is a number code:

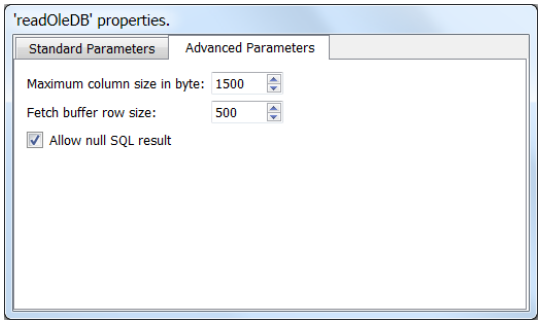
"_status" code	Description
0	No error + the SQL command returned some data
1	No error + the SQL command did not return any data
2	SQL command is Still executing?
3	The SQL command that was received was empty?
4	Error detected

5.2.3. Generic OleDB reader

Icon:

Property window:





Short description:
Reads a table from an OleDB datasource

Long Description:
OleDB is a generic technique to access the content of almost any relational databases.

All OleDB providers are required to produce as output fully compliant Unicode Strings.

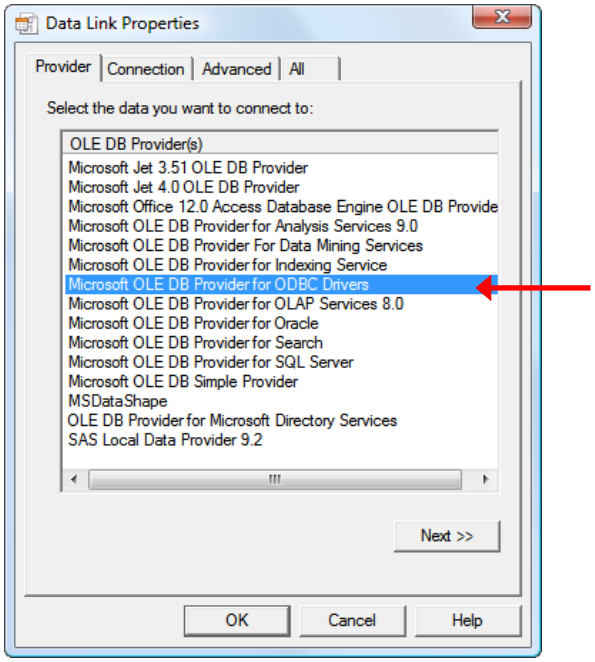
With the proper OleDB driver installed on your computer, you can access these databases:

SAS, AS/400, VSAM, VSAM-VSE, VSAM-MVS, dBase, Acceler8-DB, Microsoft SQL Server, ALLBASE, Btrieve, C-ISAM/D-ISAM, CorVision, DB2, IBM DB2/400 on iSeries (AS/400), Enscribe, IDMS, IMAGE, IMS/DB, Informix, Informix OnLine Dynamic Server, Ingres/Ingres II, Jasmine, jBASE, MUMPS, NonStop SQL/MP, ObjectStore, Oracle, QueryObject, Rdb, Red Brick, RMS, Sybase, SQLite, Firebird/Interbase, MySQL, ADABAS, Approach, Btrieve, DataFlex, DBMS (CODASYL), DMS II (CODASYL), DMS 2200 (CODASYL), Domino, FoxPro, IMS, Lotus, Micro Focus, Microsoft Access, Microsoft Excel, Paradox, PowerFlex, PostgreSQL, Centura, Datacom, IDMS, OS/390 sequential files, Pervasive SQL, Progress, SAP, Advantage Database Server, ADDS, D3, General Automation, Mentor, mvBase, mvEnterprise, Pick, Reality, Reality/X, Sequoia, Unidata, Universe, Ultimate, UltPlus, SQLBase, Essbase, Peoplesoft, Lawson, Active Directory Provider, Analysis Services Provider, Commerce Server Provider, Provider for Internet Publishing, Index Server Provider, SNA Server, Office documents, Teradata, OpenLink Virtuoso, Microsoft Exchange 5.5 and 2000. MAPI compliant sources, CodeBase Server, Clipper, XML, HTML tables, LINC II, MCP Data Files, Successware Engine, Apollo Database Server, Outlook 2000.

Let's give a small example of usage. The first thing that you want to obtain when working with an OleDB datasource is an "OleDB connection string". An "OleDB connection string" contains all the information required to access the database: the type of database (Access, SQLServer, MySQL, Oracle,...), the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

database files and name, your login and password. Click on the Get new Connection String with Wizard button: A (Microsoft-Generated) wizard open:



This wizard lists all the OleDB providers installed on your computer.

If you can't see an OleDB driver that you just installed, it means that you are using Anatella 64-bit and you installed some 32-bit OleDB drivers (or vice versa).

Be sure to install 64-bit OleDB drivers to be able to use Anatella 64-bit

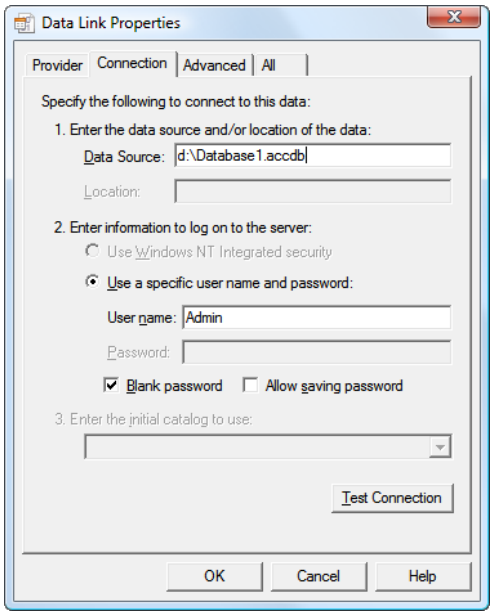
(or install 32-bit OleDB drivers, if you want to use Anatella 32-bit).

OleDB drivers are different from ODBC drivers: i.e. If you installed on your machine some ODBC drivers, it does NOT mean that you also installed at the same time OldDB drivers: These are different.

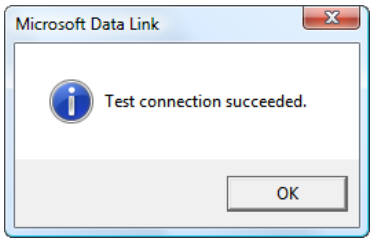
For example, if you want to read an Access database, you should select the "*Microsoft Office 12 Access Database Engine OleDB provider*" (you can also use the "*Microsoft Jet 4.0 OleDEB provider*", for older Access versions). In this example, we will open an Access 2007 database: click on "*Microsoft Office 12 Access Database Engine OleDB provider*" and then click on the "next" button. The next tabs depend on the chosen driver and might vary from one driver to the other.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Fill-in the “Data source” field:

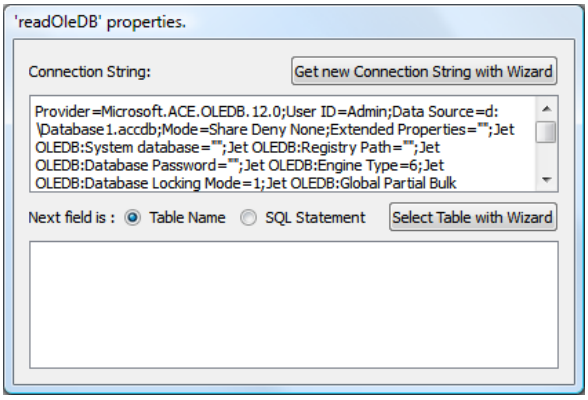


When you click the “Test Connection” button, you should have:



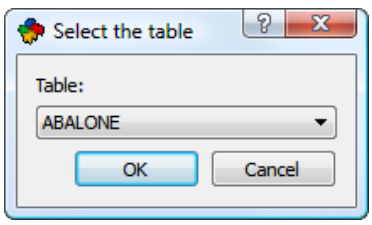
Close the test-connection window and click Ok in the “Data-link Properties” window.

Let’s go back in Anatella. The “OleDB connection string” box is now setup properly:



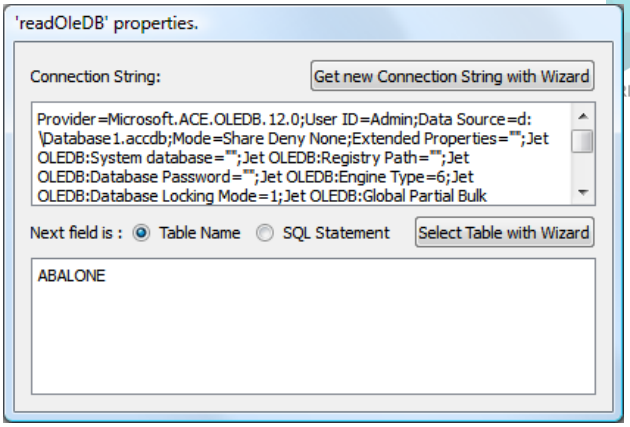
Click on the

button. A list of table inside your database appears:



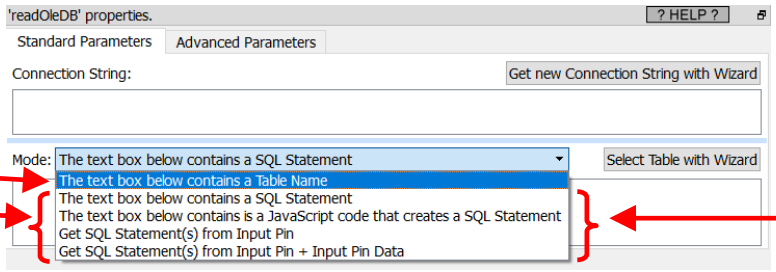
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Click the "Ok" button. You obtain:

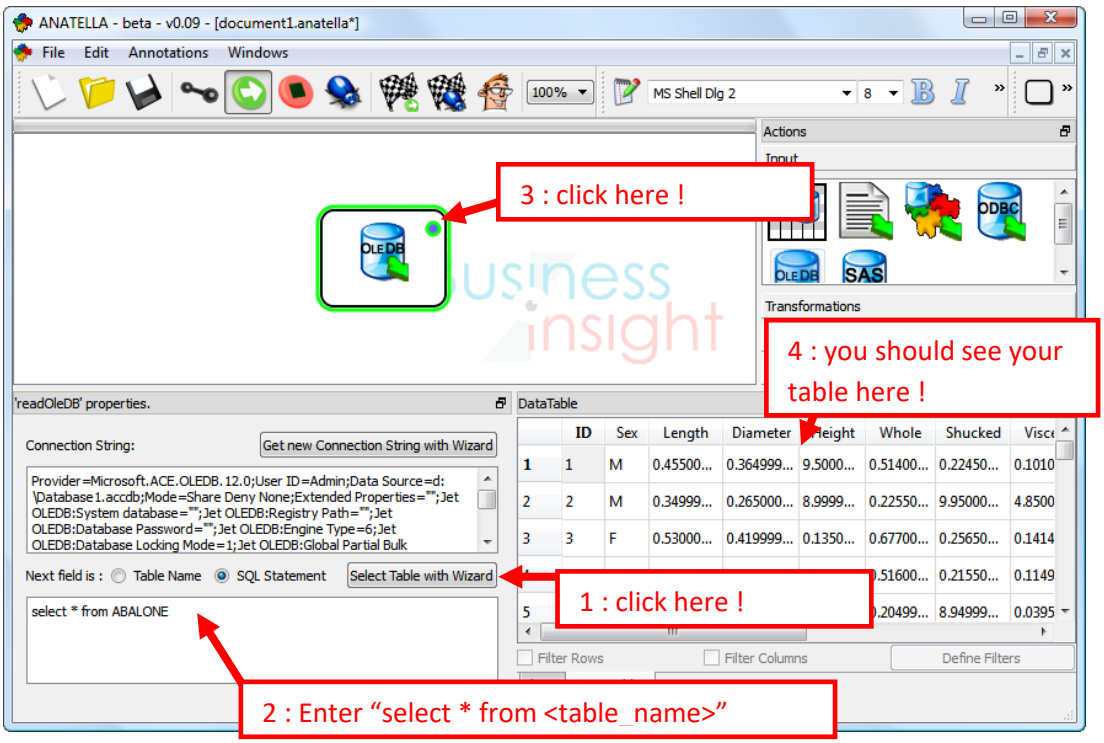


An OleDB provider can work using two different access modes:

1. Table mode.
2. SQL mode.



Depending on the available OleDB provider, both access modes can be implemented or only one. Usually, the fastest access mode is "*Table mode*" (the speed entirely depends on the OleDB database driver) but it's not always available. The only way to know if you can use the "*Table mode*" is to test it: Click on the output pin of the "OleDB reader" (Don't forget to switch to "Run Mode" before clicking on the pin). If it fails, then, you have no other choice than to use (one of) the "*SQL mode*" access:



When using the "*SQL mode*" access, you can define your SQL statement in 4 different ways:

These 4 ways of defining your SQL statement are the same as for the readODBC action and are described in more details in section 5.2.2.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The OleDB reader Action submits SQL script to the database and, then, waits for the output rows that the database can (optionally) return (some SQL statements do not return any results). This means that, in particular:

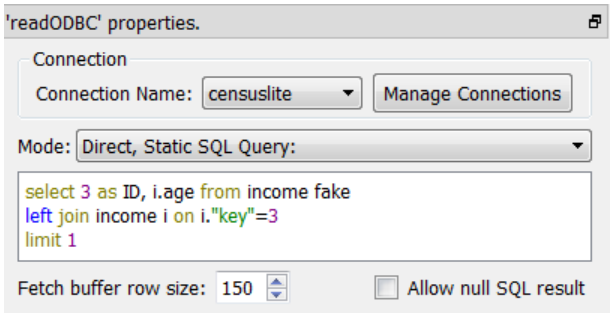
1. If you are experiencing very long running time, it means that the database engine has difficulties processing your SQL statement (…and Anatella is innocent! ☺). Try simplifying your SQL statements (adding "top 100" usually helps).

2. Some database engine have strong limitations on the number of tables that can be included inside a JOIN statement (e.g. Access2010 is limited to 5 tables, so it's best to extract one table at a time and compute the JOIN with Anatella). If the SQL statement fails because of some limitations of the database engine, Anatella will display an OleDB error message explaining the nature of the error, allowing you to track and resolve the error. The content of the OleDB error message is produced by your database. Please, refer to your database documentation for more information about the reported errors.

5.2.4. SQLite Reader – GIS support

Icon:

Property window:



Short description:
Launch the execution of SQLite SQL statement(s) based on (several) SQLite database(s).

Long Description:
SQLite files are self-contained, serverless, zero-configuration, transactional SQL databases.
SQLite is the most widely deployed SQL database engine in the world.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. The SQLite engine runs directly inside the Anatella process.

The SQLite engine reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is 100% cross-platform. Inside SQLite, All transactions are ACID (Atomicity, Consistency, Isolation, Durability) even if interrupted by system crashes or power failures. This make SQLite databases particularly resilient and ***reliable***. These features make SQLite a popular choice as a convenient file format to exchange data between different applications.

Although the SQLite engine is one of the ***fastest*** database engine available, you should NOT use it to perform "heavy" data transformations on Big Data. If you need to perform large and complex data transformations, avoid using the SQLite engine because it's a lot slower (and less scalable) than the standard Anatella data transformations (i.e. the Anatella Actions). The SQLite engine has been included

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

inside Anatella mainly for the purpose of "easy data exchange" between applications (and not to perform heavy-duty data transformations using SQL).

If you need to perfom large data transformations, you should always use the Anatella Engine rather than the SQLite Engine. There exists however one data-transformation-operation where the SQLite engine *might* be faster than the standard Anatella engine: It's the rowFilter Action. Inside Anatella, the rowFilter Action always performs a "full table scan": It reads all the rows of the input table and outputs only the desired rows (This is because the .gel_anatella file do not contain any INDEXING structure of any kind). If you need to output a very small number of rows, it might be faster to create an INDEX on the table to "filter" and use this INDEX to find the desired rows to output (This avoids reading the whole table because you only need to read the INDEX data and the selected rows). With the SQLite engine, you can create INDEXES on tables and use these INDEXES (inside a SQL statement) as a substitute to the rowFilter Action. Of course, if the rowFilter Action outputs anyway 95% of the rows of the input table, this substitution makes no sense (because, in such a case, a "full table scan" approach is more efficient than an INDEX-based approach). One last alternative is to use a columnar ".**c**gel_anatella" file. Columnar files are also able to filter rows very efficiently: See the example named "Read the rows for which 'Acquisition Date>=2011'" inside section 5.26.3. for more information about this subject.

SQLite databases are limited in size to 140 terabytes.

An SQLite database is not intended to be an enterprise database engine. It is not designed to compete with Oracle, Teradata or PostgreSQL. Use SQLite in situations where simplicity of administration, implementation, and maintenance are more important than the countless complex features that enterprise database engines provide. **SQlite is a "small" engine (with a limited set of features) that is easy to administrate**.

The major limitation of SQLite is the relatively smaller number of simultaneous users able to **write** inside a SQLite database. SQLite supports an unlimited number of simultaneous **readers**, but it will only allow one **writer** at any instant in time. This is usually not a problem because each application write its "changes" into the database quickly and then moves on, and no lock lasts for more than a few dozen milliseconds. But there are some applications that require more concurrency, and those applications may need to seek a different solution.

This makes SQLite databases particularly well suited for enterprise datawarehouse that are populated using batch processing with Anatella. Such datawarehouses are infrequently changed (and only in "batch": thus there usually exists only one **writer**: Anatella) but, on the other hand, these datawarehouses are typically accessed daily by hundreds of users (there are thus many **readers**) to:

- Create reports and plot charts (using OLAP reporting tools such as Tableau, Quickview, Mondrian, Business Objectics, etc.)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- analyse website logs,
- analyse sports statistics,
- compile programming metrics,
- analyse experimental results.
- Etc.

Since SQLite supports an unlimited number of simultaneous readers, it's particularly well-suited for such use-cases.

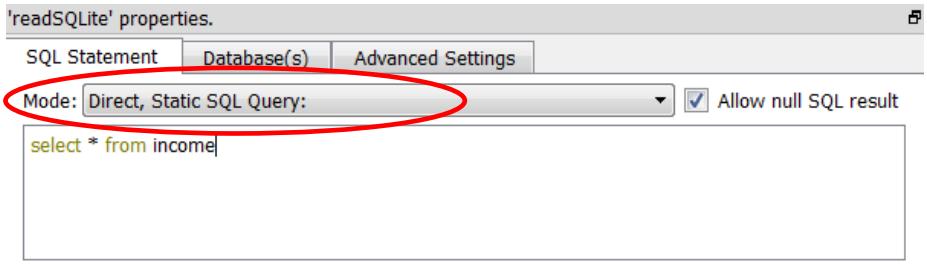
According to the developpers, the three main advantages of SQLite over other solutions are: "***SQLite is Small, Fast & Reliable***".

You can drag&drop a .SQLite file or a .db3_file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding SQLReader Action inside the Anatella graph.

There are 4 operating modes for the SQLiteReader Action:

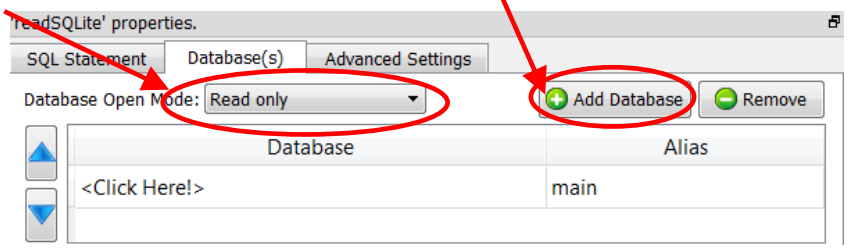
1. Direct, Static SQL Query
2. SQL Queries computed using JavaScript code
3. SQL Queries from Input Pin
4. SQL Queries from Input Pin + Input Pin Data

These are the same 4 operating modes that have been described inside the section 5.2.3. about the OleDB Reader. You can select the operating mode here:



The SQLite database engine is able to compute the results of SQL queries that ares mixing different tables from different databases (i.e. from different files). This is very handy because it allows you to gain some precious computing- time because you don't need to merge all the required tables inside one unique database file before using them: You can directly work on the "source" database files without doing any data "copy". Click the "Add Database" button to add the different databases (i.e. the different files) that are required to write the required SQL statements.

Inside the "Database(s)" tab, you can also select which mode is used to open a connection to a SQLite database:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The "Database Open Modes" are:

1. **"Read Only" Mode:** If the database does not already exist, an error is returned.
2. **"Read & Write" Mode:** The database is opened for reading and writing if possible, or reading only if the file is write protected by the operating system. In either case the database must already exist, otherwise an error is returned.
3. **"Read & Write – Create if not Exists" Mode:** The database is opened for reading and writing, and it is created if it does not already exist.

When opening SQLite databases, you should always favour the "Read Only" mode since the SQLite database engine allows an unlimited number of "readers" but only one writer at any instant in time. When a database is opened in "Read & Write" Mode, all readers must wait until the "write" operations are complete (unless you are using the "WAL" locking mode. A writer that uses the "WAL" mode does not block the readers but it slows them down radically).

The SQLite engine runs directly inside the Anatella process (it's an "embedded" SQL database engine). Some complex SQL queries might consume a large amount of RAM memory. To prevent the SQLite engine to use all the memory available for the Anatella process (i.e. 2GB RAM if running inside 32 bit Windows), you can set an upper bound on the memory used by the SQLite engine (see the parameter "Max Memory" inside the "Advanced Settings" tab).

The default value for the parameter "*Max number of Columns in query*" is 2000. You can change this value to a higher number. However, a larger number might slow down radically the time required to prepare the SQL statements (Because there are places in the SQLite code generator that are using algorithms that are $O(N^2)$ where N is the number of columns.).

When a SQLite database is opened in "Read & Write" mode, you can run SQL commands to change the content of the database. All these SQL commands are included inside one unique transaction (unless you manually add some "COMMIT; BEGIN TRANSACTION;" in the middle of your SQL code). There are basically 3 different ways used by the SQLite engine to handle transactions:

1. All rollback mechanisms are disabled (i.e. "JOURNAL_MODE=OFF"). This is very insecure but it's also the fastest mode.
2. Standard rollback mode (i.e. "JOURNAL_MODE=DELETE"): We create a JOURNAL file that contains all the required information to undo the changes made to the database, if a rollback is required.
3. New Write-Ahead-Logging (ie. "JOURNAL_MODE=WAL"): During a transaction, we write all required changes inside a separate WAL file without modifying the database (Multiple transactions can be appended to the end of a single WAL file). At one point in time, we move all the transactions back into the database (This is called a "*checkpoint*").

For most operations, "Write-Ahead-Logging" (i.e. "WAL" mode) is usually a lot faster than the old Standard Rollback Mode (i.e. the "JOURNAL_MODE=DELETE"). You'll find more information about this subject here: http://www.sqlite.org/draft/wal.html

Using the "Locking Mode" parameter inside the "Advanced Settings" tab, you can select which mechanism (i.e. JOURNAL_MODE=OFF, DELETE or WAL) is used to handle your transactions.

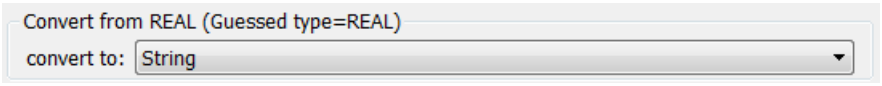
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If your SQL statement is not returning any rows (i.e. it's not a "SELECT" statement but it's rather a "TRUNCATE", "CREATE INDEX", etc. statement), you need to check the "*Allow NULL SQL result*" checkbox otherwise Anatella will abort with an error message.

All normal SQL database engines use static, rigid typing. With static typing, the datatype of a value is determined by the column in which the value is stored. Static typing is also used inside Anatella.

In opposition, in SQLite, all cells inside a table can, potentially, have a different datatype (In this regard, SQLite is very similar to MS-Excel). This means that you can have a column that is filled with floating point values (i.e. the "Guessed Datatype" of the column is "REAL") and suddenly, in the middle of the column, you find a String! Aaargh! There can be several solutions to this annoying situation:

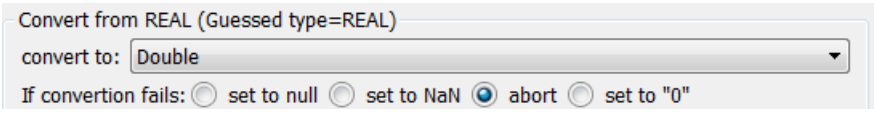
- You declare inside Anatella that the whole column contains Strings:

  In which case, you also need to specify how to convert your numbers into strings using the following formatting parameter:

  Please refer to section 5.5.1.3. for more information about this formatting parameter.

- You try to cast (i.e. convert) the String to a floating point value.
  In which case, you need to specify how to react if the conversion fails:

In the above example, Anatella guessed that the column datatype is "floating point" (i.e. "Double").

How is Anatella guessing the datatype of each column?

The result of the guess can either be: TEXT, INTEGER or REAL.
Anatella analyses how the columns were declared inside the "CREATE TABLE" statement:

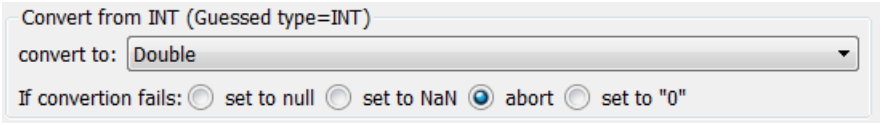
1. If there are no declared type for that column, we look at the first row of data:
   2.1. If the cell data type is INTEGER the Guessed Type is INTEGER
   2.2. If the cell data type is REAL the Guessed Type is REAL
   2.3. Otherwise the Guessed Type is TEXT
2. If the declared type contains the string "INT" then:
   4.1. If the declared type contains the string "POINT" then the Guessed Type is REAL
   4.2. Otherwise the Guessed Type is INT
3. If the declared type contains the string "CHAR", "CLOB", "TEXT", "STRING" then the Guessed Type is TEXT
4. If the declared type contains the string "REAL", "FLOA", "DOUB" then the Guessed Type is REAL
5. If the declared type contains the string "BOOL" then the Guessed Type is INT
6. Otherwise, the Guessed Type is TEXT

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

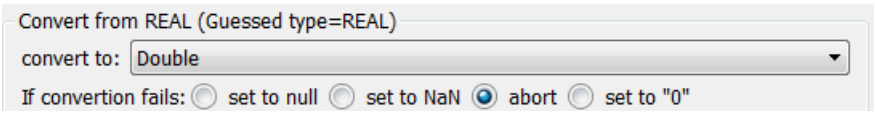
Anatella attempts to guess the type of each of the column. The result of the guess can be:

- **A column seem to be filled with TEXT**
  You can control the data-type conversion of this column using this parameter:

- **A column seem to be filled with INTEGER numbers:**
  You can control the data-type conversion of this column using this parameter:



- **A column seem to be filled with REAL (floating point numbers)**
  You can control the data-type conversion of this column using this parameter:



5.2.4.1. GIS Support

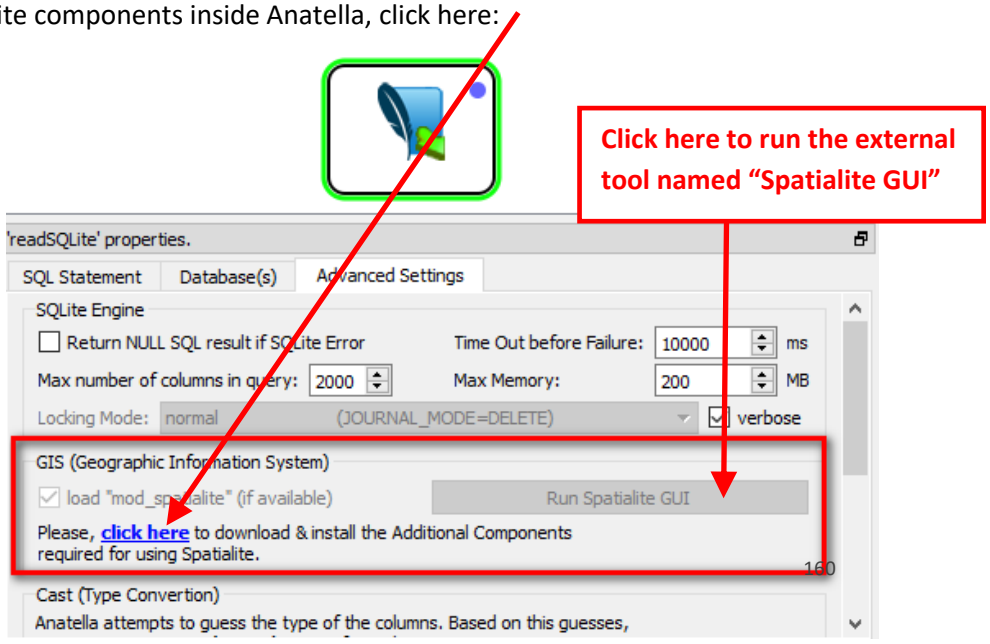
The SQLite engine integrated inside Anatella supports the Spatialite Exentions to do advanced GIS queries on your GIS databases.

The Spatialite library offers mostly the same set of GIS functionalities that are available in PostgreSQL + PostGIS (but faster), Oracle Spatialite or SQL Server. The spatialite library is mostly OGC-SFS compliant (i.e. it's compliant with the "Open Geospatial Consortium" standard).

When installing the additional Spatialite components inside Anatella, you also get:

- a large documentation (151 pages PDF) on how to use spatialite (with QGIS) to make GIS operations (e.g. geographical joins). For example, with spatialite (and the GIS data from the Government), we can:
  - Count the number of "mail boxes" in a given radius around any given agency or Point-Of-Sale.
  - Estimate the average area (in square meters) of these houses (e.g. to get an estimate of household revenue)
- An additional external tool named "Spatialite GUI" that is optimized to allow you to easily design GIS queries interactively.

To install the additional Spatialite components inside Anatella, click here:

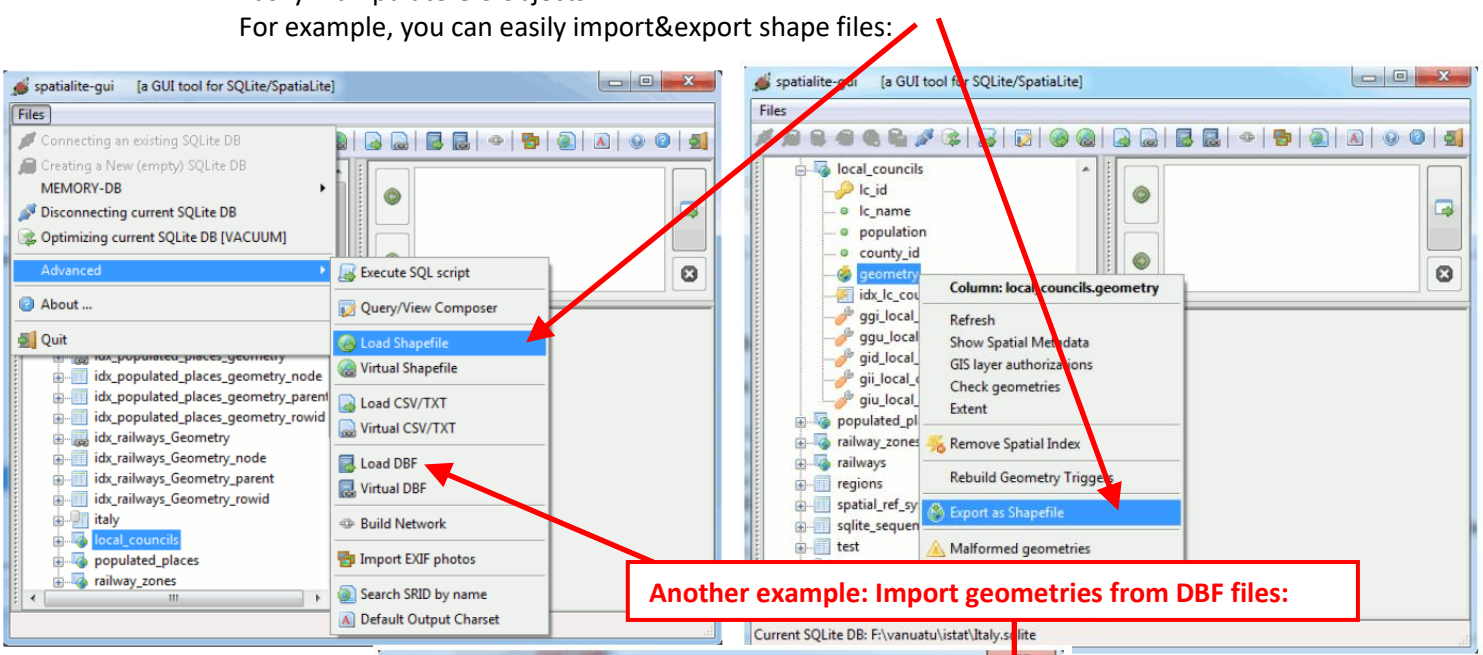


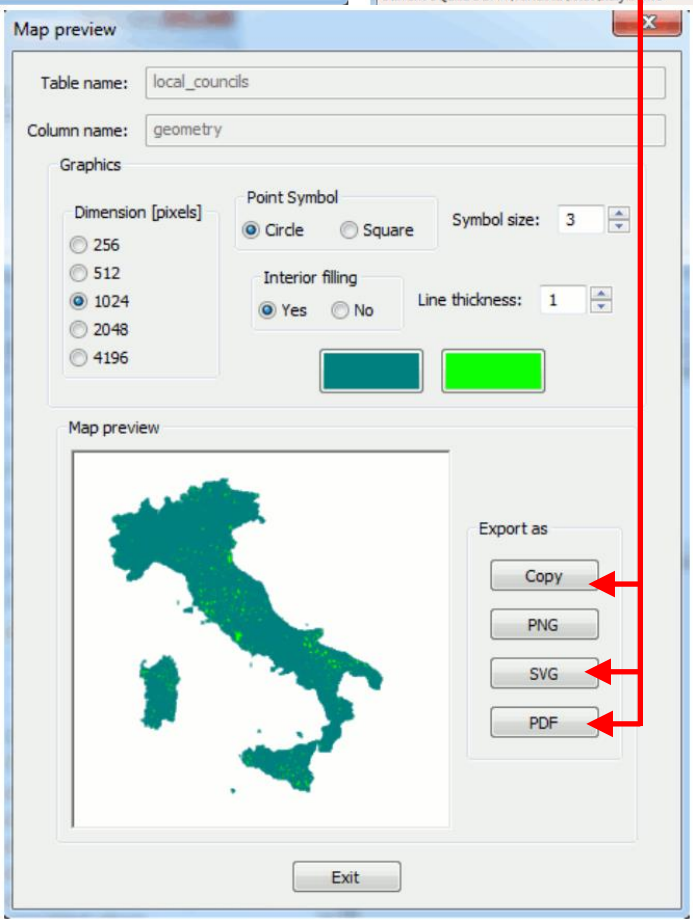
The complete list of functions available inside the Spatialite extention is given here:
http://www.gaia-gis.it/gaia-sins/spatialite-sql-4.3.0.html

To test if the Spatialite extention has been correctly loaded inside Anatella, run the following SQL command: `select spatialite_version()`

The external tool named "Spatialite GUI" (available here) allows you to:

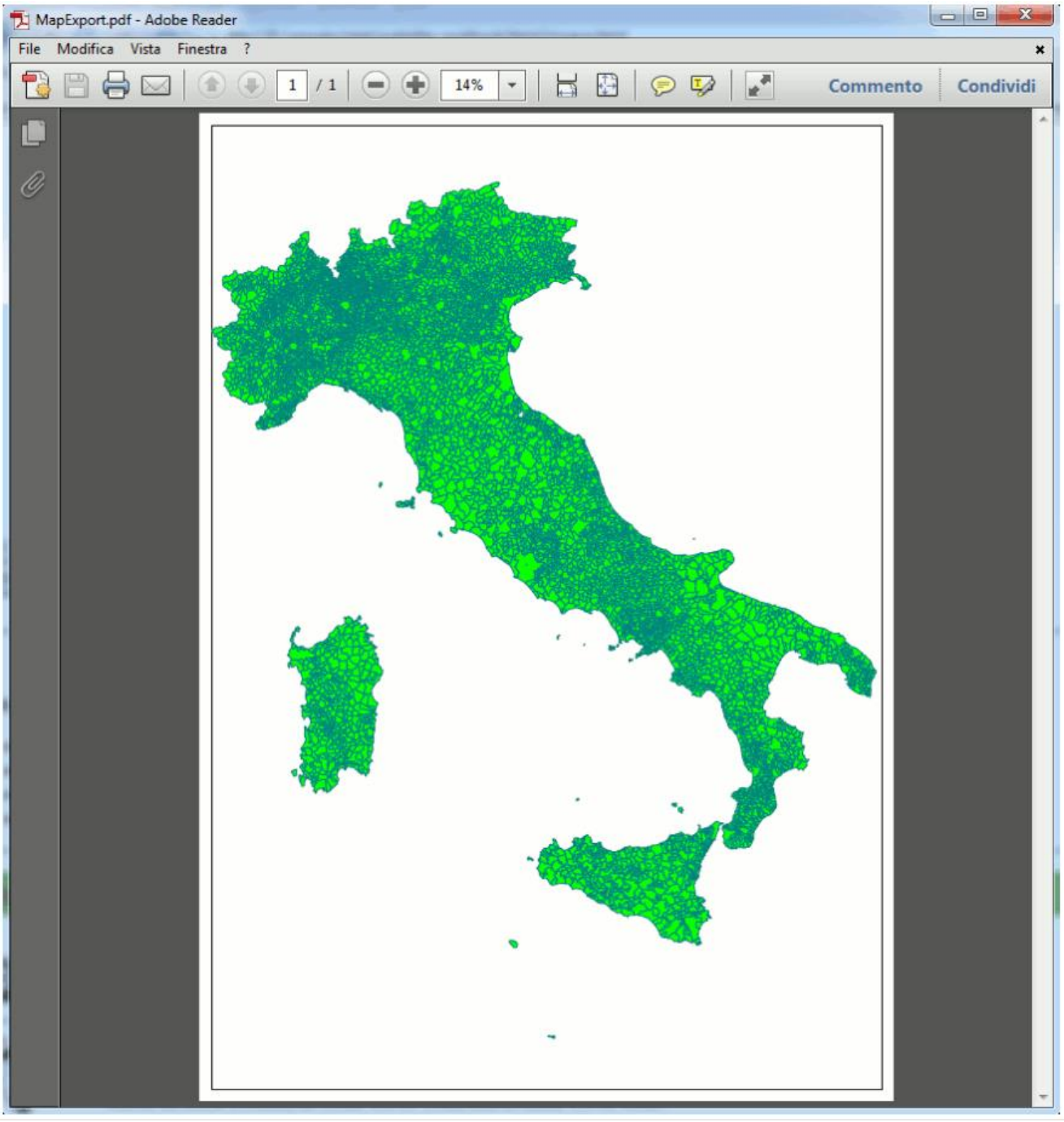
- Easily create GIS queries interactively
- Easily manipulate GIS objects.
  For example, you can easily import&export shape files:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is an example of PDF exportation:



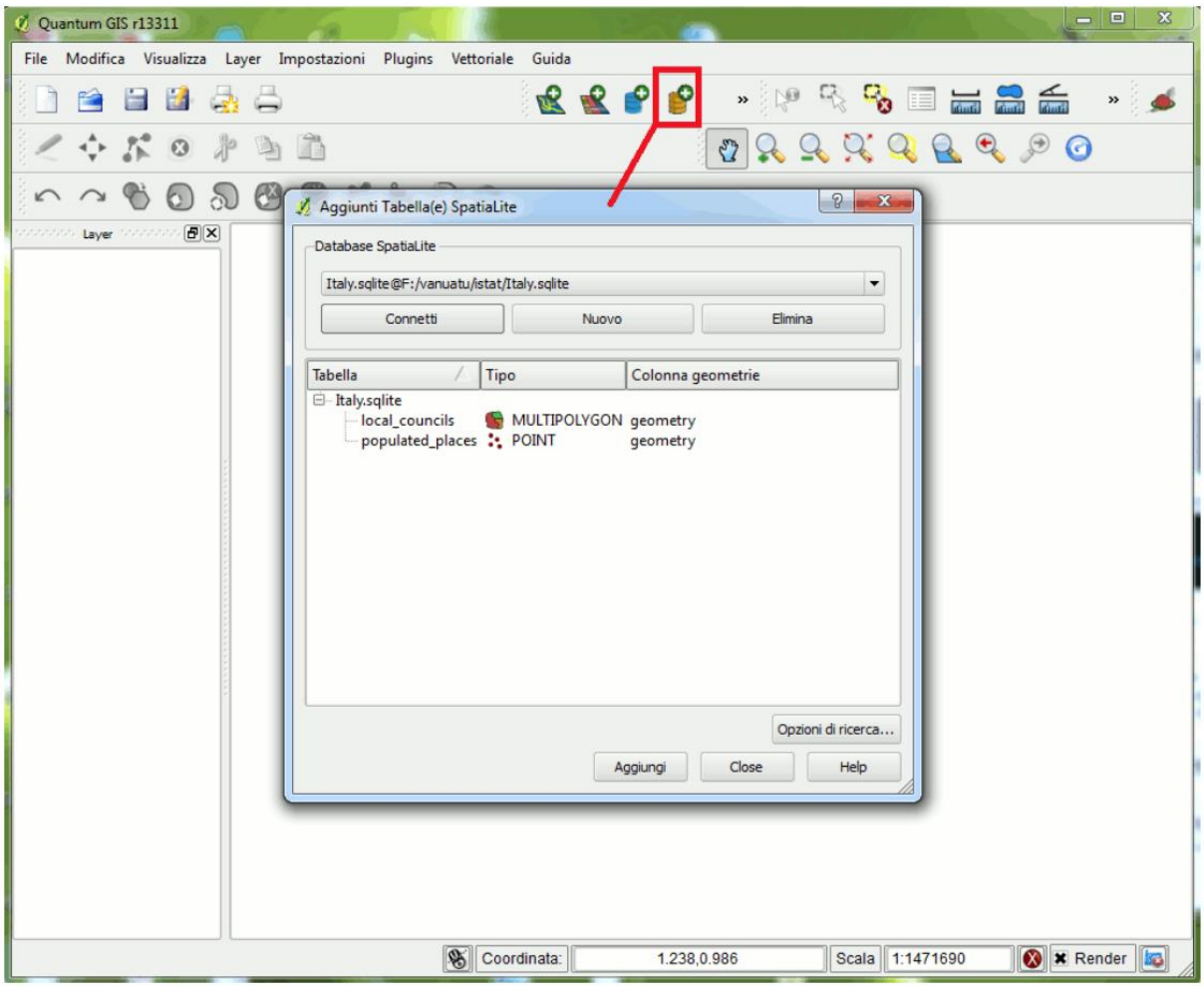
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is an example of SGV exportation:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

…but maybe the easiest way to interactively see your geometries is to use the "Quantum GIS" software (also named QGIS). QGIS is a really popular and widespread desktop GIS app: You can download the latest QGIS from: http://www.qgis.org/

QGIS contains an internal data provider supporting SpatiaLite: So, interacting with any SpatiaLite's DataBase using a classic desktop GIS is simple and easy: Here is an example of using a Spatialite DataBase inside QGIS:



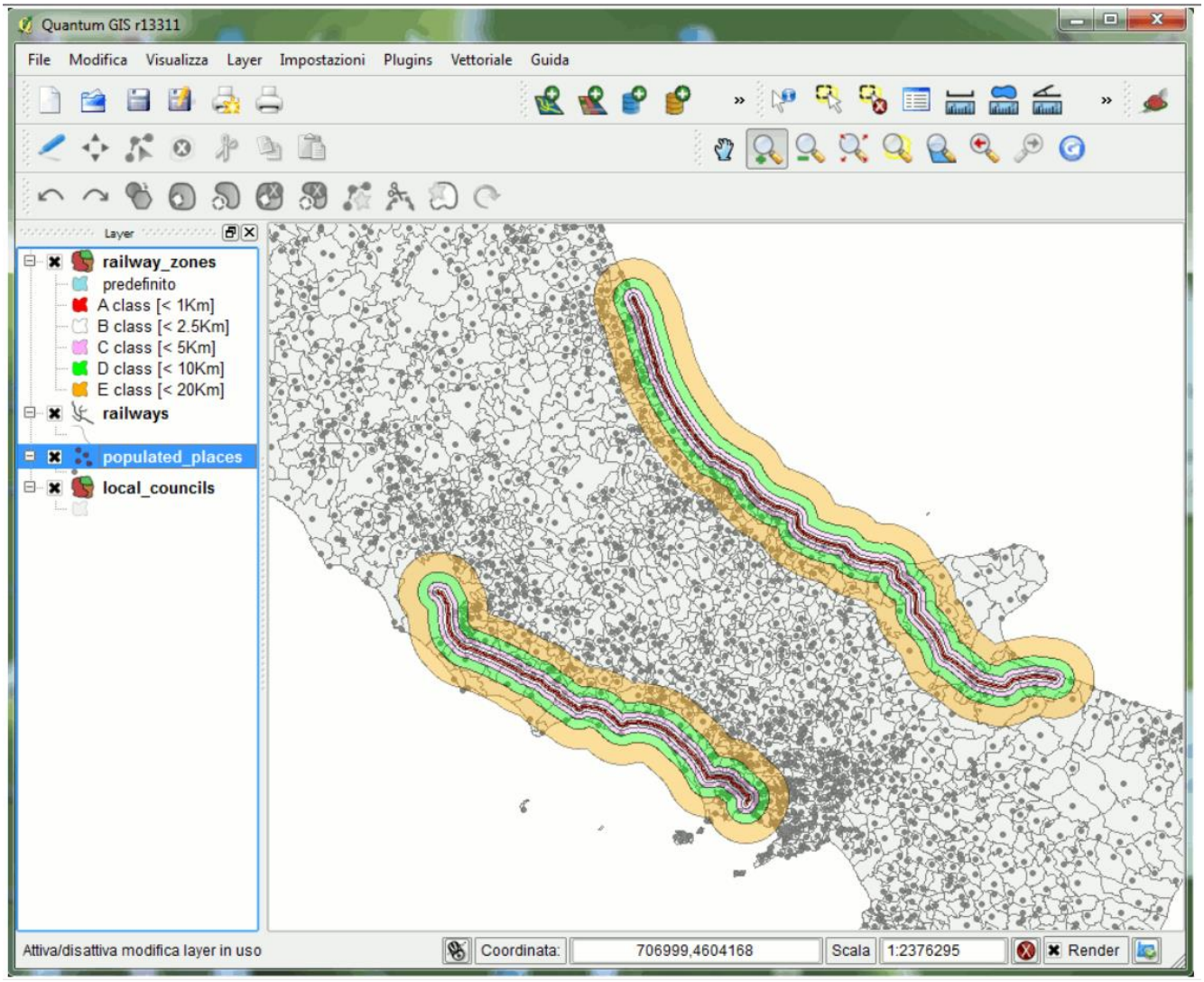
You simply have to connect the SpatiaLite's DB, then choose the layer(s) that you intend to use.
Please note: According to DBMS terminology you are accustomed to handle tables. But in the GIS own jargon the term layers is very often used to identify exactly the same thing.

Below is an example where we used QGIS to represent graphically the proximity to two different railway lines in Italy. The "proximity" to the railway line is divided into 5 classes that are defined this way:

Class	Min. distance	Max. distance
A-class	0 Km	1 Km
B-class	1 Km	2.5 Km
C-class	2.5 Km	5 Km
D-class	5 Km	10 Km
E-class	10 Km	20 Km

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

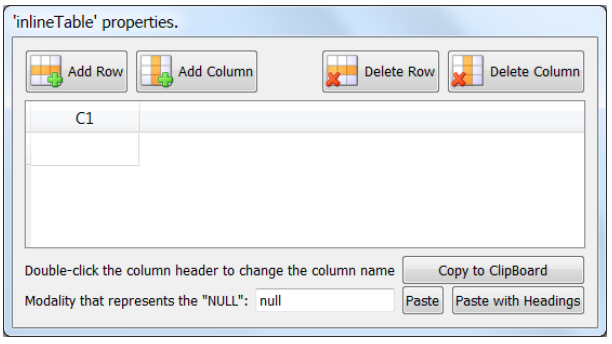
Here are the 5 classes (i.e. the A,B,C,D,E classes) represented inside QGIS, for visual inspection:



5.2.5. In-Line table

Icon:

Property window:



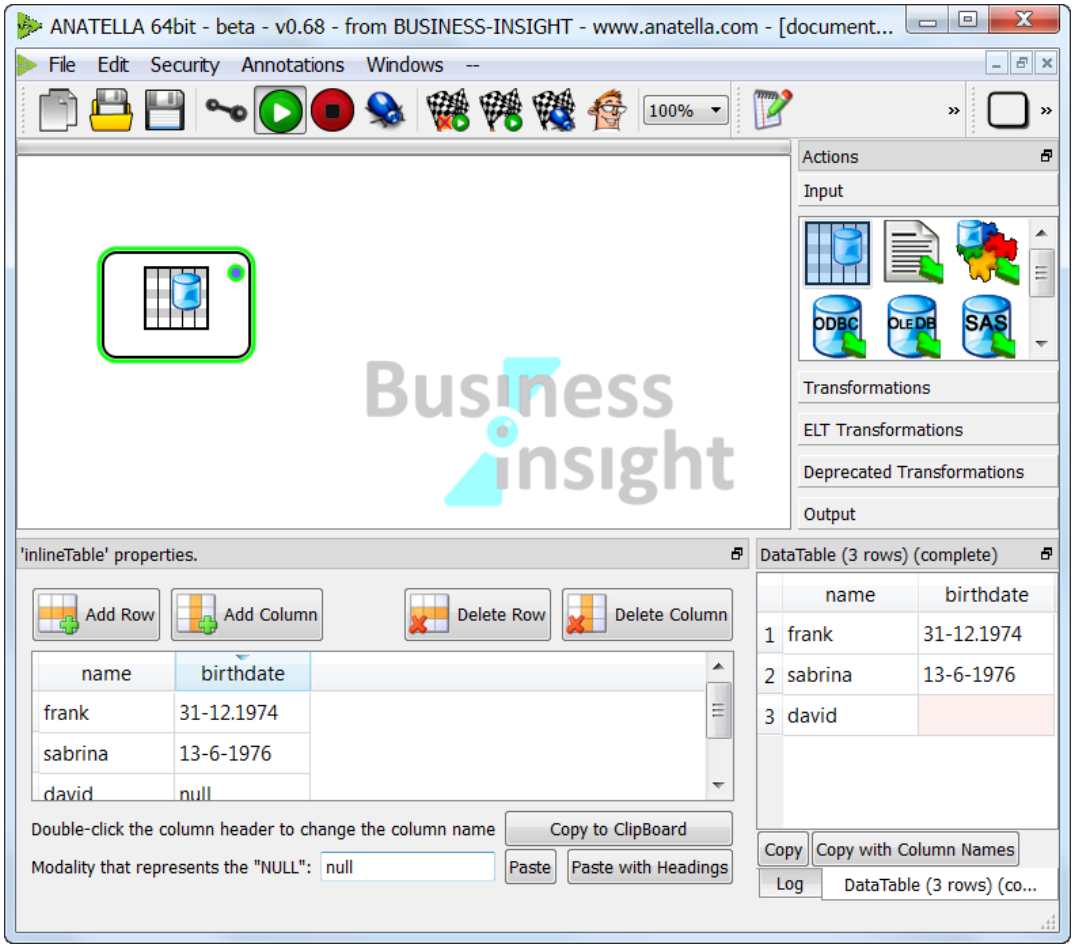
Short description:
Allows the user to input a small table “by hand”.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Allows the user to input a small table “by hand”. The table data is stored directly inside the .anatella file (rather than inside an external .txt or .gel_anatella file).

Click on the “Copy to Clipboard” to export the selected cells of the Inline-Table to MS-Excel.
Click on the “Paste from Clipboard” to import the whole Inline-Table from MS-Excel (or from the result DataTable panel of Anatella).

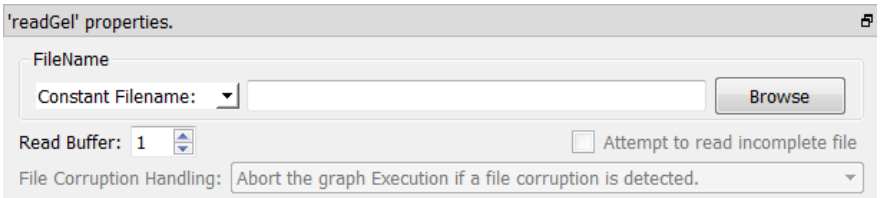
Double-click the header of a column to change the column’s name.
Here is an example:



5.2.6. Anatella “Gel” file reader (row-based Storage)

Icon:

Property window:



Short description:
Reads a table from a .gel_anatella file

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

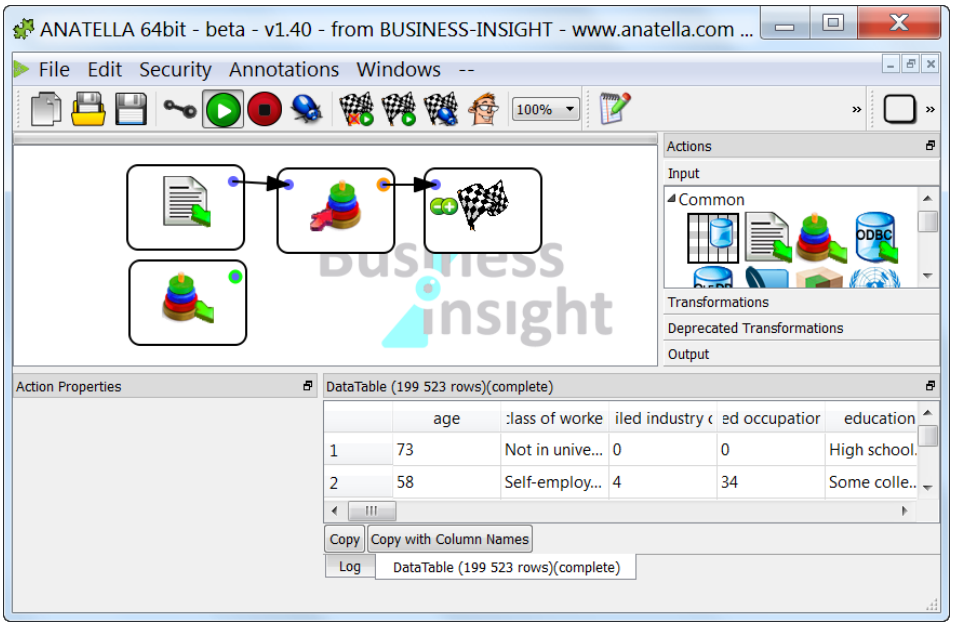
Long Description:
Reads a table from a ".gel_anatella" file. See section 5.1.1. to have more information on how to specify the filename of the ".gel_anatella" file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

You can connect to the input pin of the GelFile Reader a table containing (many) filenames. Typically, this input table will be computed using the fileListFromObsDate Action (see section 5.22.5). Anatella reads all the corresponding "Gel files" one after the other (This is more or less equivalent to the Append Action). There is a limitation: The different "Gel Files" that are read in this way must all have exactly the same meta-data.

The Anatella "HD cache" system is automatically creating "Gel Files" when you click on the output pins of the different Actions. These "Gel Files" are also useful when you want to exchange data with other Anatella processes.

You can drag&drop a .gel_anatella file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadGel Action inside the Anatella graph.

Here is an example:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

An ".gel_anatella" file contains:

- The table data
- The following meta-data:
  - For each column: the name
  - The columns on which the table is sorted and the type of sort used.
  - For each column: the content type:
    - Alpha numeric (i.e. the "String" or "Unknown" type inside Anatella)
    - Numeric (i.e. "Floating-point numbers": the "Float" type inside Anatella)
    - Integer Number (i.e. the "Key" type inside Anatella)
    - Date
    - Pure Binary data (this might contains '\0' chars, for example)
    - Image
    - Sound
    - Unknown
  - For the columns containing a date: the data-format
- A flag that indicates if the ".gel_anatella" file is complete: If a user interrupted the data transformation that was computing the ".gel_anatella" file (or if the data transformation aborted due to an error), then we can obtain an "incomplete" ".gel_anatella" file meaning that the file should most certainly contains some more rows that have not been computed and saved.

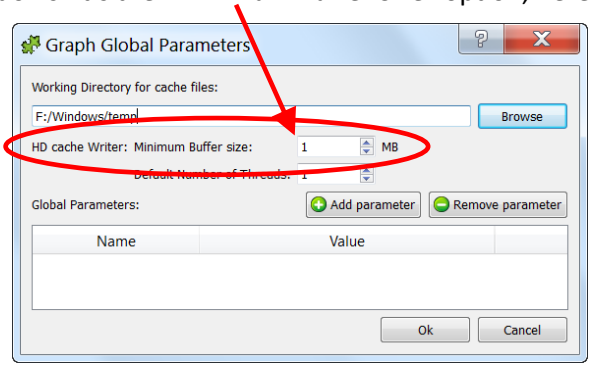
5.2.6.1. Reading files with the "complete" meta-data-flag set to false.

You can force Anatella to read incomplete ".gel_anatella" files (i.e. a file with the "complete" meta-data flag set to false) by cheking the "*Attempt to read incomplete file*" checkbox. To be able to modify this checkbox, you first need to switch to "Expert User Mode": see section 6 about "Expert User Mode" (Press CTRL-U anytime to switch the user mode).

5.2.6.2. About asynchronous (and synchronous) I/O algorithms

The data inside a ".gel_anatella" file is compressed using block-based compression algorithm.
This means that the algorithm used to read a ".gel_anatella" file is the following:

1. Extract/Read one data-block out of the hard drive. The data-block size is defined when you create the ".gel_anatella" file, using the GelWriter Action. You can select the data-block size at the moment of creating the ".gel_anatella" file (i.e. When you read the file, it's too late: i.e. You don't have any control on the data-block size anymore) using the "Graph Global Parameters" Windows: It's the "Minimum Buffer Size" option, here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

It can happen that you are forced to open a large number of ".gel_anatella" files simultaneously (for example, when using the mergeSortInput Action: see section 5.2.15.). Keep in mind that each opened ".gel_anatella" file uses (by default) around 1 MB of RAM (…and it uses even more RAM if you set the "Read Buffer" parameter greater than one). Thus, to open one thousand ".gel_anatella" files simultaneously, you need at least 1GB of RAM: This is already a lot of RAM on a small 32-bit server and might lead to some crashes.

Using a larger block size than 1MB means that the compression is (very slightly) better and the reading&writing speed is also slightly better (faster).

2. Validate that the data-block is not corrupted: Anatella compute a checksum on each data-block.

3. Decompress the data-block to get some "data rows" and send these rows to the connected Actions as output.

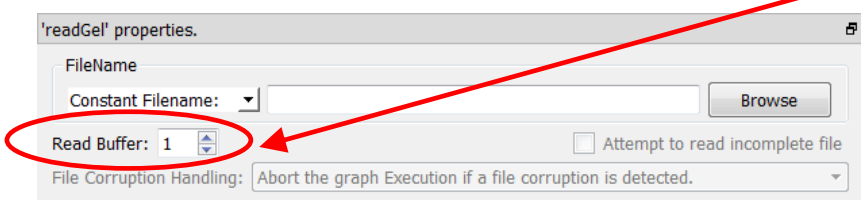
4. Wait for the Actions connected to the ouput of the GelFileReader Action to use/consume all the rows from the current data-block. Once all the rows from the "current block" are used, go back to step 1.

In terms of speed, the above algorithm used to read the ".gel_anatella" file is not very efficient because, it's a "synchronous (i.e. blocking)" I/O algorithm: i.e. When the Actions connected to the ouput of the GelFileReader Actions are requesting some more rows (i.e. when they just consumed all the current rows from the current data block), the data-transformation:

- Blocks until the extraction of the next data-block out the Hard Drive is completed.
- Blocks until the validation of the next data-block is completed.
- Blocks until the decompression of the next data-block is completed.

…and then (and only then) the data-transformation unblocks and you receive the next rows.

It would be nice to have a better algorithm that is an asynchronous (i.e. non-bloking) I/O algorithm: When you set the value "2" for the "Read Buffer" parameter from the GelFileReader Action, Anatella will use an asynchronous (i.e. non-bloking) I/O algorithm to read the ".gel_anatella" file (this implies using several threads in parallel). This means that all the extraction/validation/decompression tasks will be performed continuously "*as a background thread*" so that, when the Actions connected to the ouput of the GelFileReader Actions are requesting some more rows (e.g. from the next data-block), these rows are always directly available in RAM, ready to be used (i.e. and the data-transformation-process does not have to stupidly wait for the extraction/validation/decompression tasks to be finished). The "*background thread*" continuously produces new rows (so that they are always directly available when requested). These rows are stored inside one of the (many) "Read Buffers". The quantity of "Read Buffers" is set here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When the parameter “Read Buffer” is:

- ***“1”:***
  - Anatella uses a synchronous (i.e. bloking) I/O algorithm.
  - This is the best option if you want to reduce to the minimum the RAM consumption (at the cost of speed). By default, the GelFileReader Action uses 1MB of RAM.
- ***“2”:***
  - Anatella use an asynchronous (i.e. non-bloking) I/O algorithm.
  - Lectures from the Hard Drive are faster but RAM comsumption is two times higher. With the default settings, the GelFileReader Action now uses 2MB of RAM.
  - This is usually the best option when your “.gel_anatella files” are stored on a local hard drive, with constant guaranteed access speed.
- **“3 and higher”:**
  - Anatella use an asynchronous (i.e. non-bloking) I/O algorithm.
  - Lectures from the Hard Drive are faster but RAM comsumption is higher (by default, the RAM consumption is 1MB multiplied by the quantity of “Read Buffers”).
  - This is the best option when your “.gel_anatella files” are stored on a distant hard drive on a remote server with an unconstant, varying computer network speed (e.g. when the Anatella server is connected to a common-grade 100Mbit/sec or 1Gbit/sec computer network).

    A large value for the quantity of “Read Buffers” allows to cope with a sudden&brief “speed drop” of your network drive. Let’s assume that your remote drive stops working for a very brief moment of time (e.g. there are some “network drops”). During this second, the Actions connected to the ouput of the GelFileReader Action won’t stop working because they will use the rows that have been pre-loaded into the N “Read Buffers” (with N, a very large number). As soon as the network drive works again, the “*background thread*” directly fills-in the N “Read Buffers” to the maximum, so that, when the next “drive-speed-drop” occurs, the data-transformation-computations can still continue without interruption. A large N value means that Anatella can compute its data-transformation at full speed, even on a very poor, unstable computer network.

5.2.6.3. Reading Corrupted “.gel_anatella” files

When some bytes of some data-block inside a “.gel_anatella” file are corrupted, Anatella won’t be able to decompress the corrupted data-block anymore. When this happens, you can choose to, either:

- Stop the data transformation with an error (This is the default behavior).

- Discard the corrupted data-block (and all the rows that it contains), still continuing to execute the data transformation (but with a warning inside the log window). This is handy when you still want to continue the computations when there are only a few thousands rows that are “lost” out of several billion rows.

  To be able to select this option, you need to switch to “Expert User Mode”: see section 6 about “Expert User Mode” (Press CTRL-U anytime to switch to expert user mode).

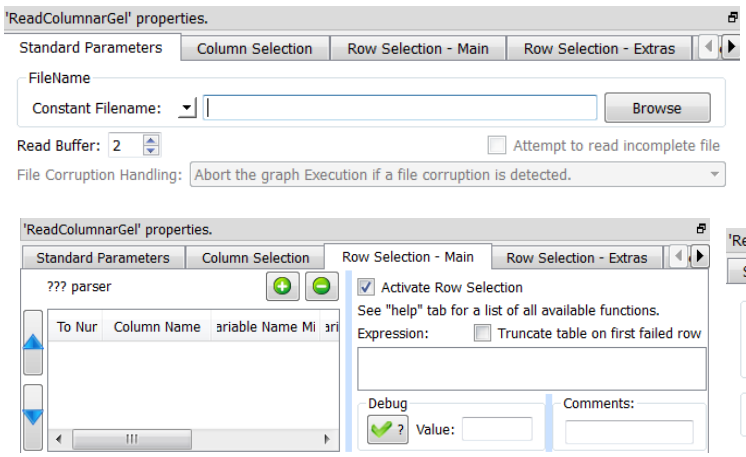
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The quantity of lost rows depends on the data-block size. You can change the data-block size inside the "Graph Global Parameters" Windows: it's the "Minimum Buffer Size" option (see above).

5.2.7. Anatella "Columnar Gel" file reader (column-based storage)

Icon:

Property window:



Short description:
Reads a table from a "**.c**gel_anatella" file.

Long Description:
Reads a table from a ".cgel_anatella" file (and from the associated "column set" data files "*.NNN.cs_anatella"). See section 5.1.1. to have more information on how to specify the filename of the ".cgel_anatella" file (i.e. You can use relative path, wildcards, and Javascript to specify your filename). You can connect to the input pin of the ColumnarGelFile Reader a table containing (many) filenames.

You can drag&drop a ".cgel_anatella" file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadGel Action inside the Anatella graph.

Anatella possesses two highly-efficient proprietary file formats that allows you to handle with ease any "Big Data" problem. These two files formats are:

- **".gel_anatella" files:** Optimized for speed and for low RAM consumption. Ideal when processing *all the columns and all the rows* inside a table. Since the ".gel_anatella" files have relatively low RAM consumption, this means that you can simultaneously open thousands of them (for example, when using the mergeSortInput Action: see section 5.2.15.).
- **".cgel_anatella" files:** Optimized to have the best speed and the mimimum quantity of I/O transfer. To minimize the quantity of bytes extracted from the Hard drive, you can parameter

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

the ColumnarGelFile Reader to read a (small) subset of the columns and a (small) subset of the rows: i.e. The smaller the subset, the higher the processing speed.

The Columnar Gel files have the same set of great features as the simpler ".gel_anatella" files: More precisely:

- The Columnar Gel files contain the same meta-datas as inside a simple ".gel_anatella" file (i.e. To remind you, these meta-data are: the column's names, column's type: Key, Float or Unknown/String, the sorting flags, the "complete" flag), plus some more meta-data that allows to only extract out of the hard drive a subset of the columns and a subset of the rows (to reduce the required I/O and gain speed).
- All the data inside the files are compressed. In opposition to the simpler ".gel_anatella" file (that uses only one generic data compression algorithm), inside the "cgel_anatella" columnar gel file, we use different compression algorithms for the different data types, achieving a (slightly) better compression.
- All I/O algorithms are asynchronous (i.e. non-bloking) I/O algorithm:
    - Inside the ColumnarGelWriter Action, we have an asynchronous (i.e. non-bloking) I/O algorithm to create the ".cgel_anatella" files and the ".cs_anatella" files. Furthermore, we can decide to use many threads/CPU's to create our files, to still increase writing speed.
    - Inside the ColumnarGelFile Reader, we have an asynchronous (i.e. non-bloking) I/O algorithm to read the ".cgel_anatella" files and the ".cs_anatella" files (

  See the section 5.2.6.2. about asynchronous (and synchronous) I/O algorithms.
  Asynchronous I/O algorithms allows very fast reading speed.
- It's possible to read "incomplete" columnar gel files: See section 5.2.6.1. for more information about this subject.
- It's possible to read "corrupted" columnar gel files: See section 5.2.6.3. for more information about this subject.

As you can see the ".cgel_anatella" Columnar Gel files seems to improve on all aspects compared to the simpler ".gel_anatella" files. The ".gel_anatella" files have still the "upper hand" in the following situations:

- When the number of columns is large (>300), the RAM consumption required to read&write columnar gel files might be prohibitive. This means that, for most predictive datamining tasks (that requires a large number of columns), you'll still use the simpler, row-based ".gel_anatella" files. For classical Business-Intelligence tasks, we are usually using a small quantity of columns (out of many) and thus the ".cgel_anatella" Columnar Gel files are usually better.
- When you need to read many data tables simultaneously (i.e. when the number of simultaneously opened data-file is above 40: For example, when using the mergeSortInput Action), it's better to use the simple ".gel_anatella" files (rather than the ".cgel_anatella" columnar gel files) because the simple ".gel_anatella" files require a lot less RAM to operate.

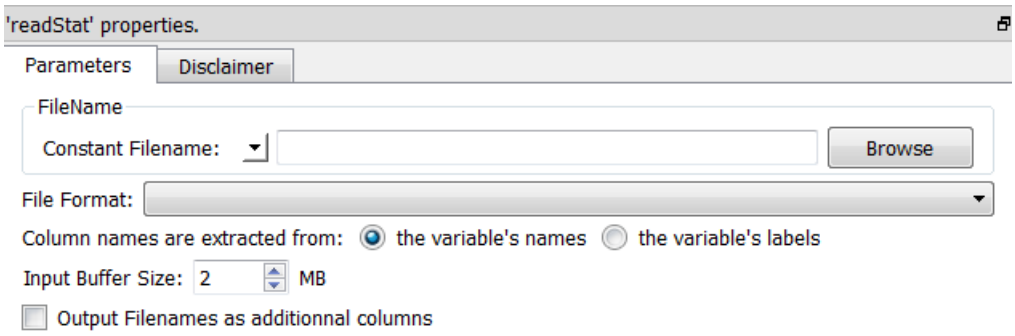
A complete explanation on the proper usage of all the parameters of the ColumnarGelFile Reader is given the section 5.26.3. about the ColumnarGelWriter Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.8. SAS (.sas7bdat), SPSS (.sav and .por) and STATA (.dta) File Reader

Icon:

Property window:



Short description:
Reads a table from a SAS (.sas7bdat), SPSS (.sav and .por) or STATA (.dta) File.

Long Description:
See section 5.1.1. to have more information on how to specify the filename of the ".cgel_anatella" file (i.e. You can use relative path, wildcards, and Javascript to specify your filename). You can also connect to the input pin of the readStat Action a table containing (many) filenames.

You can drag&drop a ".sas7bdat" file, a ".sav" file, a ".por" file or a ".dta" file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadStat Action inside the Anatella graph.

The readStat Action supersedes the old readSAS Action that was the only choice available in the older Anatella versions (older than v1.38). The readStat Action has many advantages compared to the old readSAS Action:

1. It's about three times faster.
2. It does not require you to install any SAS OleDB drivers on your computer in order to operate. This means that you can now always read your .sas7bdat files, even without requiring any "administrative" privileges from your IT department (because "administrative" privileges are required to install the SAS OleDB driver).

For faster processing speed, the readStat Action uses an asynchronous (i.e. non-bloking) I/O algorithm (See the section 5.2.6.2. about asynchronous I/O algorithms).

Sometime, the readStat Action does not correctly extract the dates stored inside some "date fields" inside a .SAS7BDAT file (i.e. you see a number instead of the actual date and time). When this happens, you need to perform an extra step to convert these

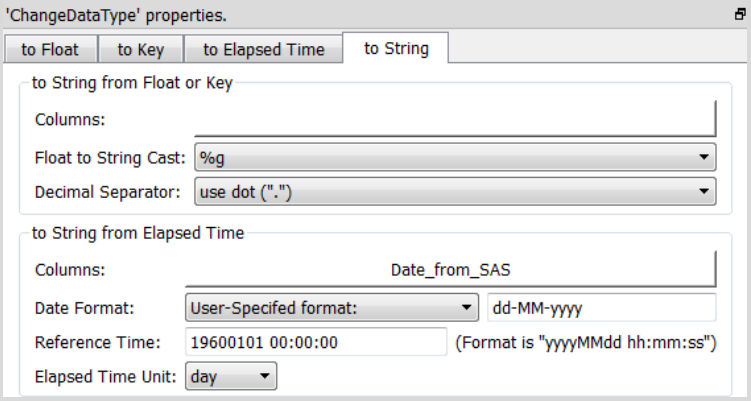
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

dates to “normal” Anatella dates: i.e. Use the “*to String from Elapsed Time*” option of the ChangeDataType Action with these parameters:

* Reference Time: 19600101 00:00:00

* Elapsed Time Unit: day

Here is a screenshot:



DISCLAIMER.

This Action allows the extraction of the data stored inside files originating from various commercial statistical systems (namely SAS, SPSS and Stata). The supported files are currently of extension: .sas7bdat (from SAS), .sav (from SPSS), .por (from SPSS) and .dta (from Stata). These files we be referred inside the rest of this section as the "Data Files".

The format of the Data Files belongs to their respective owner. These formats are proprietary and undocumented. Various students and coders all around the world have tried to decipher the internal structures of these proprietary formats. The end-result of their work is an open-source library (named "ReadStat") that we used inside this Anatella Action to decrypt&read the Data Files. Since we had no access to any kind of documentation (official or not) related to the formats of the Data Files, **we cannot offer any kind of guarantee on the proper extraction of the data stored inside the Data Files**. This means that:

● The safest way to extract data from SAS is to use the old

readSAS Action (but it requires the SAS OleDB drivers installed on your computer and it's quite slow: It's approximatively three times slower than this Action).

● The safest way to extract data from the SPSS or Stata environment is to export your data as simple text files (comma separated) and read them back inside Anatella using the "readCSV" Action.

To validate that the data were properly extracted from your “Data Files”, you should check:

● For SAS files: The character encoding: If some (accentuated) characters are incorrect or missing, you might want to use another character encoding (e.g. use the "ISO-8859-16" character encoding rather than the default "UTF-8" character encoding).

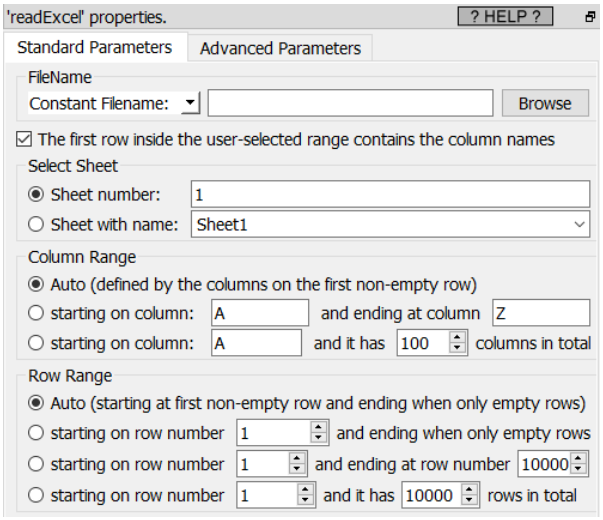
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

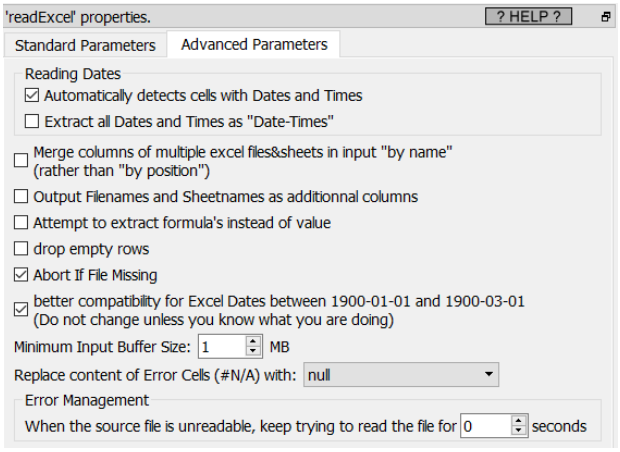
- For SAS files: SAS handles NaN and Null floating-point-numbers in a strange way. Validate that the Nulls are indeed extracted as Nulls (and not NaNs) and vice-versa.
- It's always a good ideas to validate the "Date" variables because they are always handled using the most "exotic" ways.

5.2.9. Excel File Reader

Icon:

Property window:





Short description:

Reads the newer Microsoft Excel .xlsx file.

Long Description:

The readExcel action only reads the newer excel file formats: i.e. the files with the extentions .xls**x** and .xlsm. If you need to import into Anatella some data stored into the older excel file format (from the older version of Excel before 2003; i.e. the files with the extensions .xls or .xlt), you should use the readExcelOld action (see section 5.2.33. for more details about this action). Don't worry about selecting the or the action: Just drag&drop your Excel file inside the Anatella window and Anatella will take care of the rest.

See section 5.1.1 to have more information on how to specify the filename of the .xlsx files (i.e. You can use the input pin, a relative path, wildcards, and Javascript to specify your filename(s)).

You can drag&drop a .xls**x** file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadExcel Action inside the Anatella graph.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

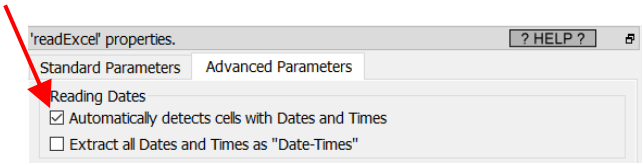
Each different cell inside a .xlsx file can have a different Data-Type (i.e. it can be String, Double, Date, etc.). Furthermore, there is no way to a-priori "guess" the most common type used inside a specific column inside an Excel sheet. For these reasons, all the cells inside an Excel file are imported inside Anatella as simple "Strings".

A Small note about the way Excel handles dates

Excel uses two different ways to store the dates&times inside the .xlsx files: Dates can either be stored as Strings or as Numbers. The dates that are stored as Strings are always imported inside Anatella without any difficulties.

By default, Anatella automatically recognize the "dates&times" that are stored as number in MS-Excel .xlsx files and directly import them in a human-readable-format ("yyyyMMdd hh:mm:ss" or "yyyyMMdd" or "hh:mm:ss"). Nothing to worry about.

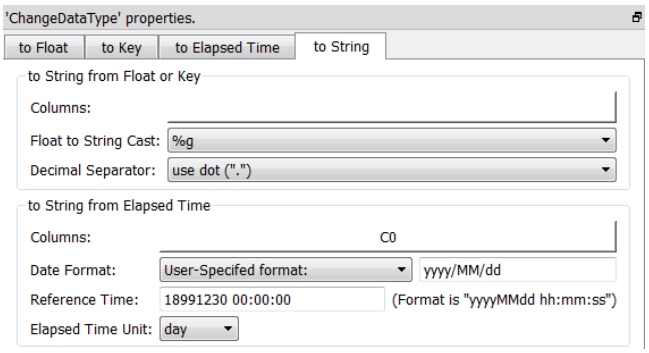
If you want to avoid this automatic conversion to a "normal, human-readable" dates (and keep the dates as "numbers"), you can uncheck this checkbox:



To convert the dates as "numbers" from Excel back to "normal, human-readable" Anatella dates, you can use the "*to String from Elapsed Time*" option of the ChangeDataType Action. Use these parameters:

- Reference Time: 18991230 00:00:00
- Elapsed Time Unit: day.

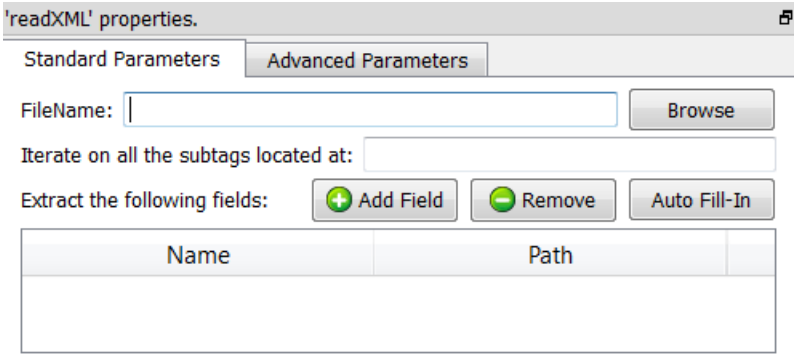
Here is a screenshot:



5.2.10. XML/HTML File Reader

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Reads a (compressed) XML/HTML file.

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the XML/HTML file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

You can drag&drop a .XML file, a .HTML file or a .HTM file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding

ReadXML Action inside the Anatella graph.

The XML/HTML reader included inside Anatella is stream-oriented. This means that Anatella reads the XML/HTML file "chunk-by-chunk" (when all data from one chunk has been extracted, Anatella loads the next chunk). This is in opposition to almost all other XML/HTML extraction engine (Almost all engines require to load the whole XML file in RAM memory before starting data extraction). This means that, in opposition to other XML/HTML extraction engine:

- There are no limits to the size of the XML/HTML file that Anatella can read&parse.
- With Anatella, Data Extraction from XML/HTML only requires a small (and constant) amount of RAM.
- With Anatella, data Extraction from XML/HTML is very fast.

If the extension of the XML file is RAR, ZIP, GZ or LZO then Anatella will transparently decompress the XML file in RAM memory. Anatella chooses the (de)compression technique to use based on the filename extension. When Anatella uses compressed file formats, it does NOT decompress the files on the Hard Drive: Anatella decompresses the data "on-the-fly" in central core RAM memory, thus reducing:

- the load on the hard drive
- the hard-drive space consumption required to do the analysis.

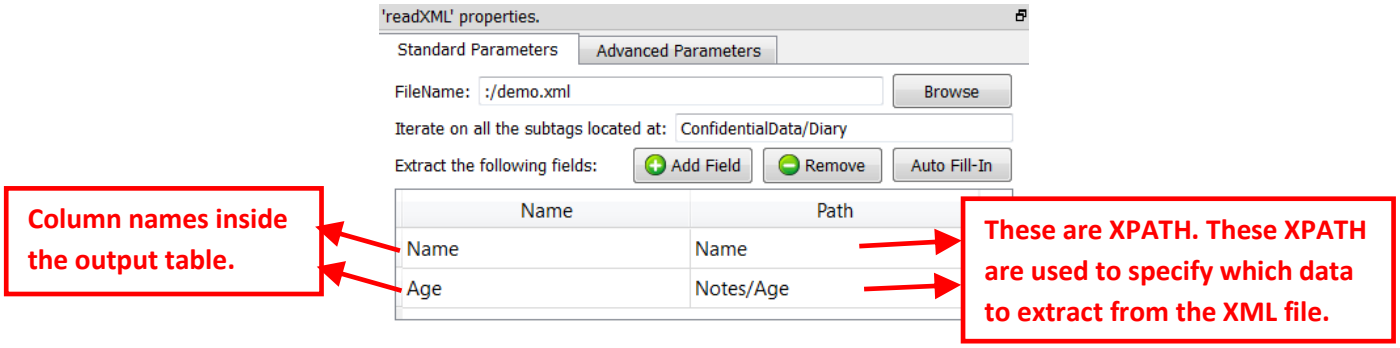
Usually, for classical "real world" XML/HTML files, the compression/ratio is above 90-95%. Thus, it makes a lot of sense to compress all your XML/HTML files.

Let's assume that you have the following XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ConfidentialData>
<Revision>3</Revision>
<Diary>
      <Contact>
            <Name>Frank</Name>
            <Address street="villers" town="ath" zip="7812"/>
            <Notes><Age>38</Age></Notes>
            <Skill>Coding</Skill>
            <Skill>Datamining</Skill>
      </Contact>
      <Contact>
            <Name>Sabrina</Name>
            <Address street="jeanne" town="ixelles" zip="1050"/>
            <Notes><Age>36</Age></Notes>
            <Skill>Coding</Skill>
      </Contact>
</Diary>
</ConfidentialData>
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We want to extract the name and the age of each of our contact.
We'll use the following settings:



We'll get as output the following table:

Name	Age
Frank	38
Sabrina	36

It can sometime be difficult to manually write the different XPATHs required for extraction. This is why Anatella has an "Auto Fill-In" button: After you finished entering the "*Iterate on all subtags located at*" parameter, you can click the "*Auto Fill-In*" button. Anatella will analyzes the first 100 `<Contact>` tags (You can change this number using the "*Number of Rows to Analyze for Auto Fill-In*" parameter inside the "*Advanced Parameters*" tab) and extract all the different XPATH to all the different data contained inside these first 100 tags. For the above example, Anatella will find the following XPATHs:

Name
Address/street
Address/town
Address/zip
Notes/Age
Skill[0]
Skill[1]

By default, the character encoding used by Anatella to decipher the content of the XML file is found automatically based on the BOM (Byte Order Mark) of the .XML file or based on the declaration on the first row of the XML file (e.g. "`<?xml version="1.0" encoding="UTF-8"?>`" is for UTF-8 character encoding). It can happen that the BOM is missing and the XML declaration is incorrect (The XML file is using another encoding than the one specified on the first row of the XML file). In such (common) situation, you can manually specify (using the "*Encoding Name*" parameter inside the "*Advanced Parameters*" tab) which character encoding Anatella must use to decipher the XML file.

5.2.10.1. HTML Extraction – A Simple Example

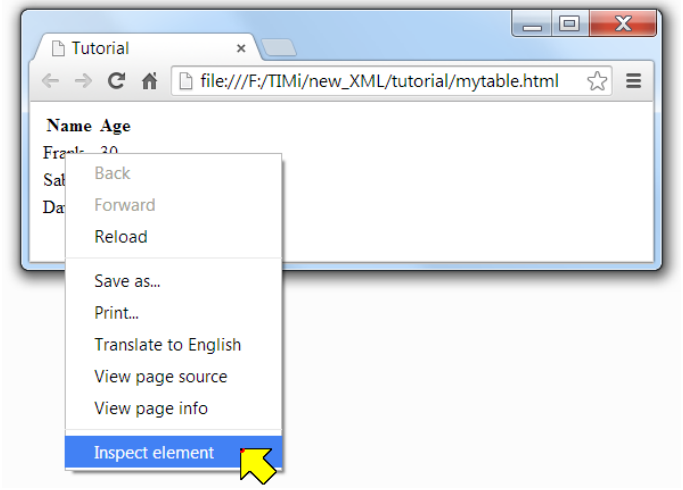
Let's assume that we have the following HTML file:

```
<html>
<head><title>Tutorial</title></head>
<body>
  <div>
    <class>
      <table>
        <tr><th>Name</th><th>Age</th></tr>
        <tr><td>Frank</td><td>30</td></tr>
        <tr><td>Sabrina</td><td>25</td></tr>
        <tr><td>David</td><td>26</td></tr>
      </table>
    </class>
  </div>
</body>
```

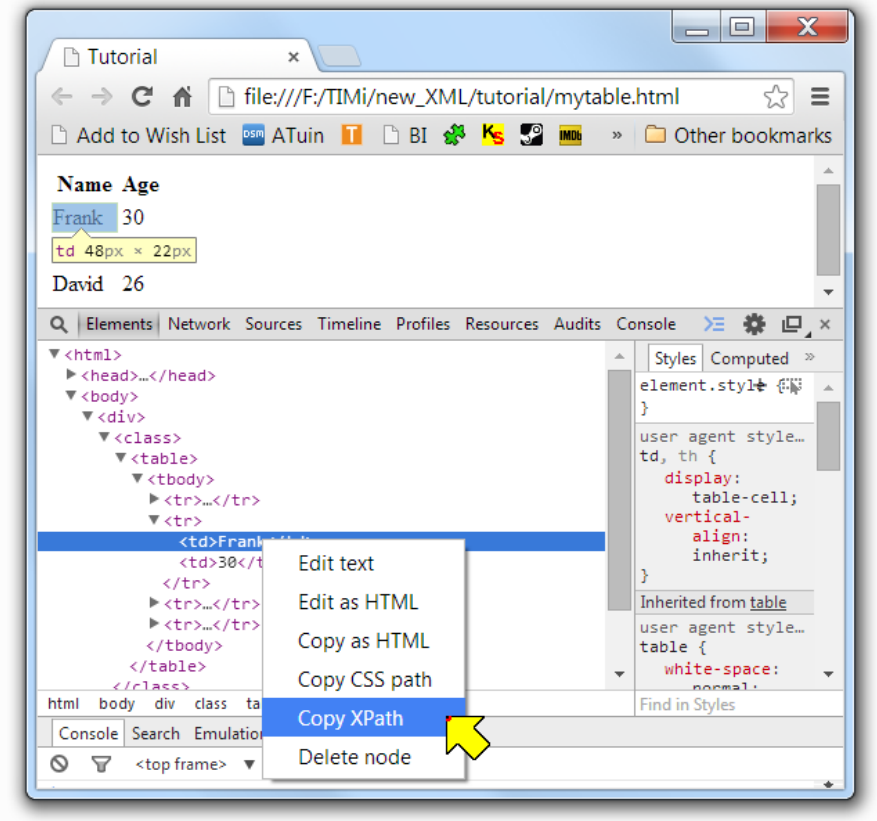
When displayed inside a browser, this HTML file looks like this:

Name	Age
Frank	30
Sabrina	25
David	26

We want to import the above table inside Anatella. First, we need to find the XPATH that gives the location of the table inside the HTML document. To do so, open the HTML file inside a Browser (e.g. inside "Chrome"), right-click on the first cell of the table and select "*Inspect Element*": See the screenshot below:



You should now see the following window:



Right-click the required XML tag (i.e. the tag that contains "Frank") and select "*Copy XPath*" in the cotext menu. The clipboard now contains the following XPath:

```
/html/body/div/table/tbody/tr[2]/td[1]
```

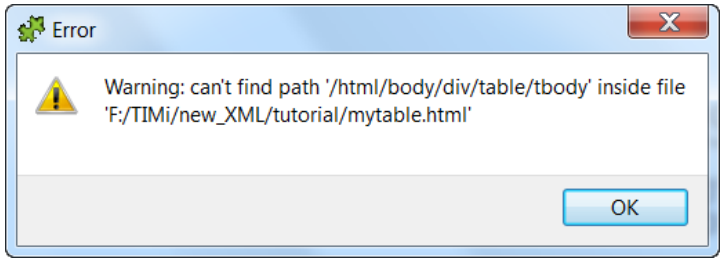
Looking at the above XPath, you can see that the XPath that represents the start of the table is:

```
/html/body/div/table/tbody
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Open Anatella, add a

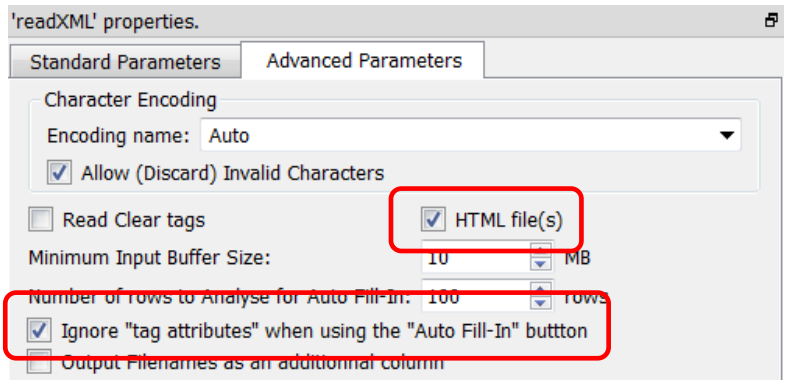
ReadXML Action inside the graph, open the "Properties window" of this ReadXML Action, paste the XPath there, and click the "*Auto Fill-In*" button: You get:



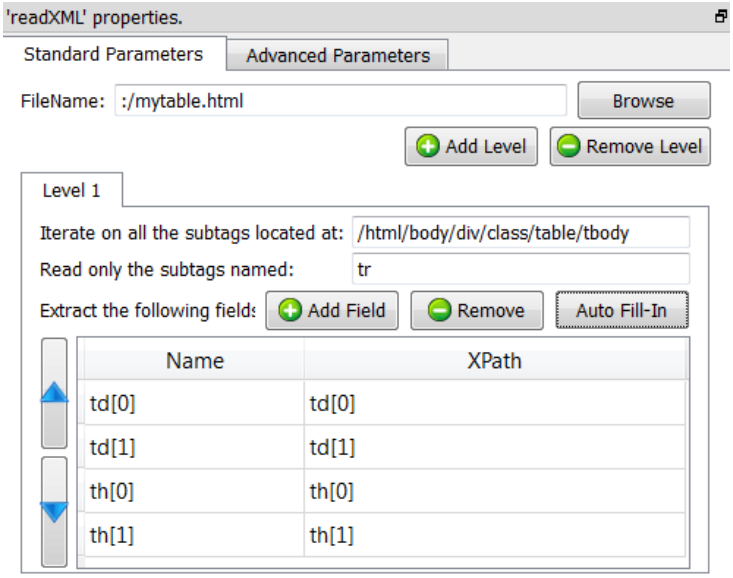
The above error message is normal because there is no "tbody" tags inside the HTML file (although a "tbody" tag appears inside your XPath expression). All the browsers always add "tbody" tags inside XPath expressions for compatibility reasons. To get around this annoying behavior from the browsers, open the "Advanced Parameters" panel and:

- Enable the check box "HTML file(s)": In addition to properly handle "tbody" tags, Anatella now also properly handles "br", "img", "link",… tags that all have a special behavior in HTML.
- Optional: Enable the check box "Ignore tags attributes when using the "Auto Fill-In" button.

You should have:

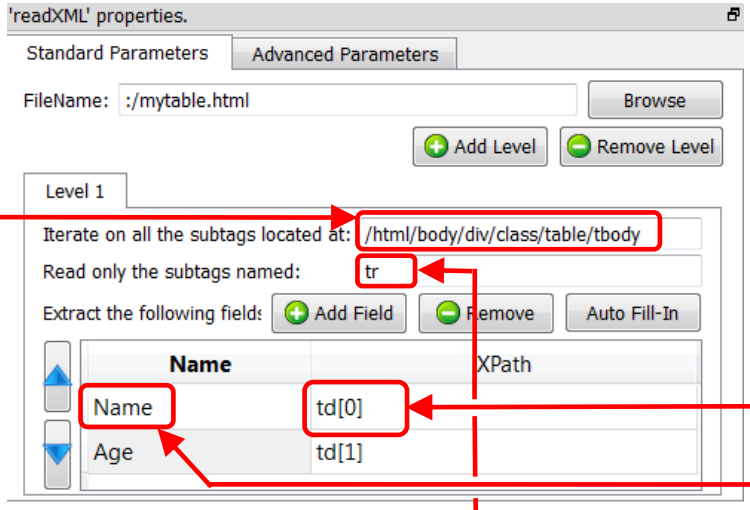


Go the "Standard Parameters" panel and click again the "Auto Fill-In" button: You now get:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Discard the last 2 rows (i.e: the last 2 extractions - You don't need them) and rename "by hand" the first 2 rows: You now have:



Looking at the above window, we can say that the complete XPath to get the different Name's is:

`/html/body/div/table/tbody` + `/tr`[<iteration_counter>]/ + `td[0]`

Run the extraction (i.e. click the output pin of the ReadXML action): You get:

DataTable (4 rows)(complete)

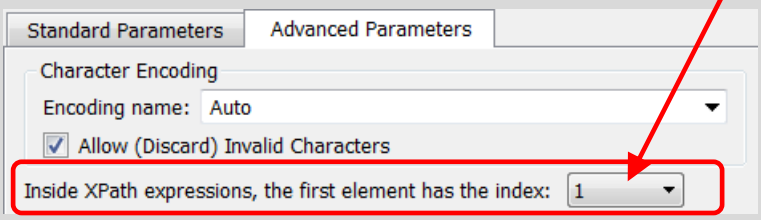
	Name	Age
1		
2	Frank	30
3	Sabrina	25
4	David	26

DataTable (3 rows)(complete)

	Name	Age
1	Frank	30
2	Sabrina	25
3	David	26

You should still add a simple FilterRow Action to remove the "empty rows" (e.g. use the following expression: "strlen(Name)>0"). You finally get:

Inside Anatella, by default, the XPath indexes are zero-based (everything is zero-based in Anatella). This means that, inside Anatella, "`td[0]`" is equivalent to "`td`". Unfortunately, HTML browsers are using indexes inside XPath expression that are "one-based" (i.e. for HTML browsers, "`td[1]`" is equivalent to "`td`"). Thus, to be able to directly copy/paste XPath expressions from your browser into Anatella, please verify that you changed the Anatella's setting to use "one-based" XPath expression: Click here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.10.2. HTML Extraction – A More Complex Example

Let's assume that we have the following HTML file:

```
<html>
<head><title>Tutorial 2</title>
<style type="text/css">
.number {text-align: right; }
.internalTable {margin-left:50px; width:100%; }
</style>
</head>
<body>
<table style="border: 1px solid black;">
  <tr><th>Client</th><th> Purchased_Item</th><th> Price  </th></tr>
  <tr><td colspan="2">Frank<br>
    <table class="internalTable">
      <tr><td>Ipad</td><td class="number">100</td></tr>
      <tr><td>Book</td><td class="number">10</td></tr>
    </table>
  </td></tr>
  <tr><td colspan="2">Sabrina<br>
    <table class="internalTable">
      <tr><td>Cup</td><td class="number">5</td></tr>
      <tr><td>Spoon</td><td class="number">3</td></tr>
      <tr><td>Teat</td><td class="number">10</td></tr>
    </table>
  </td></tr>
</table>
</body>
```

Client	Purchased_Item	Price
Frank		
	Ipad	100
	Book	10
Sabrina		
	Cup	5
	Spoon	3
	Teat	10

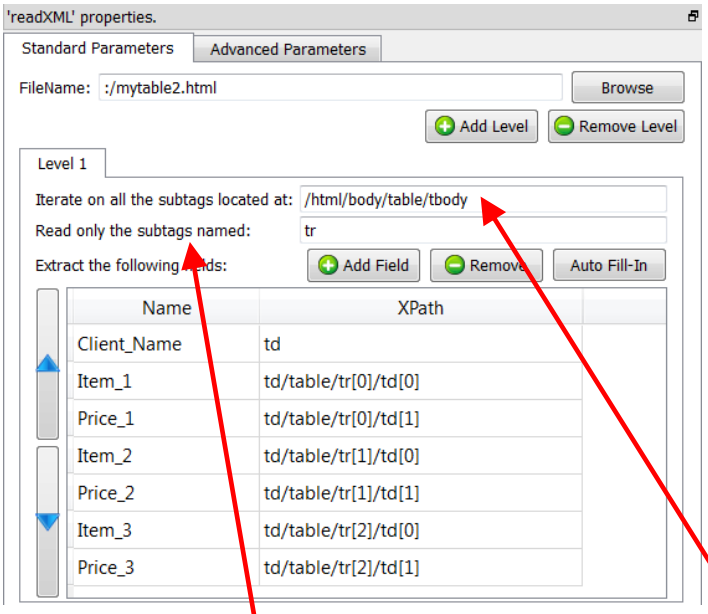
There are two "iteration levels" inside this HTML file:

- The first iteration is about the Clients
- The second level of iteration is about the transactions that each client commited. The second level is "embedded" inside the first level.

To extract this HTML, you have 2 solutions.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here are the parameters for the first solution:



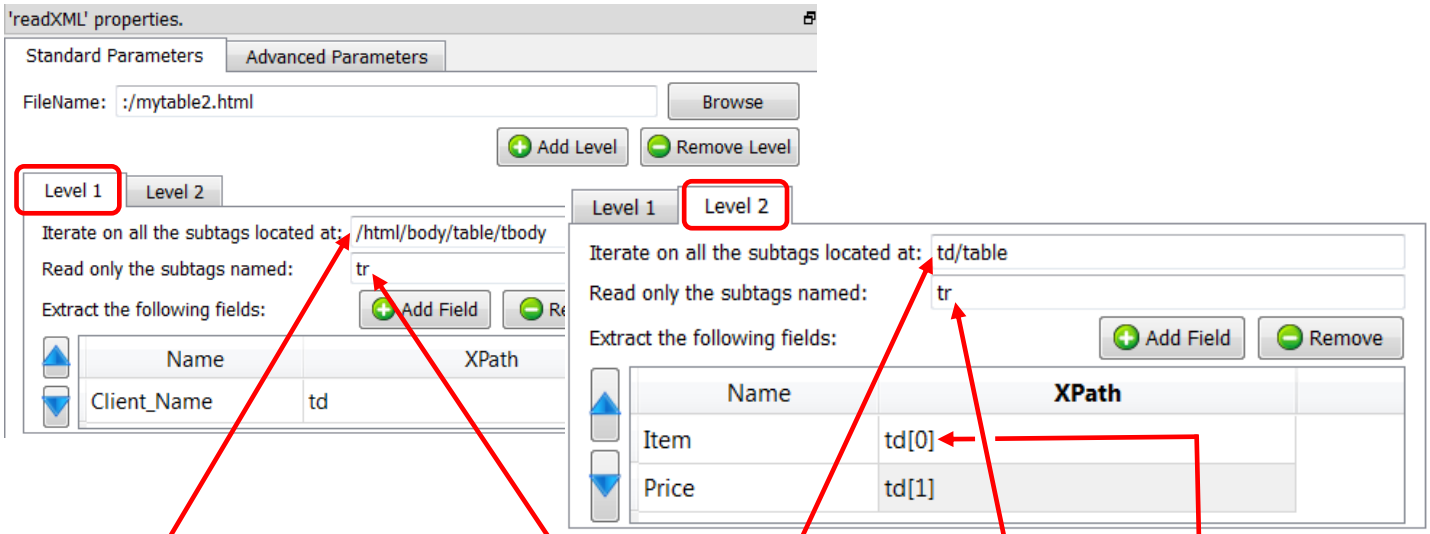
The parameter "*Read only the subtags named:*" is optional.
If left blank, Anatella simply iterates on all the subtags found here .
E.g. for the above example, you can let this parameter blank: It does not change anything.

The output of the first solution is (after removing the empy row):

DataTable (2 rows)(complete)

	Client_Name	Item_1	Price_1	Item_2	Price_2	Item_3	Price_3
1	Frank	Ipad	100	Book	10		
2	Sabrina	Cup	5	Spoon	3	Teat	10

This first solution is not very good because the number of "Purchased items" extracted is limited to 3.
Here are the parameters for the second solution: Please note that we are now defining 2 levels:



The complete XPath to get the name of the first item purchased by the first customer is:

```
/html/body/table/tbody  +  /tr/  +  td/table  +  /tr/  +  td[0]
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The output of this second solution is (after removing the empy row):

	Client_Name	Item	Price
1	Frank	Ipad	100
2	Frank	Book	10
3	Sabrina	Cup	5
4	Sabrina	Spoon	3
5	Sabrina	Teat	10

This second solution is better than the first one because it does not impose any restriction on the number of transactions that each client commited (and, also, it looks more like the original HTML file).

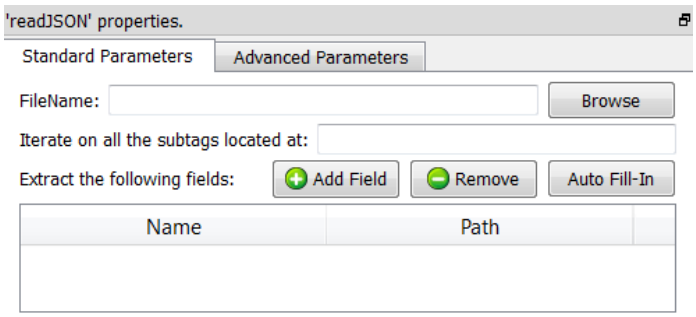
Anatella allows you to define as many levels as you like, so that you can easily extract any data from any HTML file, whatever the size and structure. Furthermore, it's very easy to find the right extraction parameters for the

ReadXML Action (i.e. the right XPATHs) because you can directly use the XPath expressions generated by Chrome, Firefox or IE (i.e. nearly all other HTML parsers that are based on XPath expressions can't use XPath expressions generated by Chrome, Firefox or IE because they have problems with <tbody> tags).

5.2.11. JSON File Reader

Icon:

Property window:



Short description:
Reads a (comrpessed) JSON file.

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the .JSON file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can drag&drop a .JSON file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding

ReadJSON Action inside the Anatella graph.

The JSON reader included inside Anatella is stream-oriented. This means that Anatella reads the JSON file "chunk-by-chunk" (when all data from one chunk has been extracted, Anatella loads the next chunk). This is in opposition to all other JSON parsers (Almost all parsers require to load the whole JSON file in memory before starting data extraction). This means that, in opposition to other JSON extraction engine:

- There are no limits to the size of the JSON file that Anatella can read&parse.
- With Anatella, Data Extraction from JSON only requires a small (and constant) amount of RAM.
- With Anatella, data Extraction from JSON is very fast.

If the extension of the JSON file is RAR, ZIP, GZ or LZO then Anatella will transparently decompress the JSON file in RAM memory. Anatella chooses the (de)compression technique to use based on the filename extension. When Anatella uses compressed file formats, it does NOT decompress the files on the Hard Drive: Anatella decompresses the data "on-the-fly" in central core RAM memory, thus reducing:

- the load on the hard drive
- the hard-drive consumption required to do the analysis.

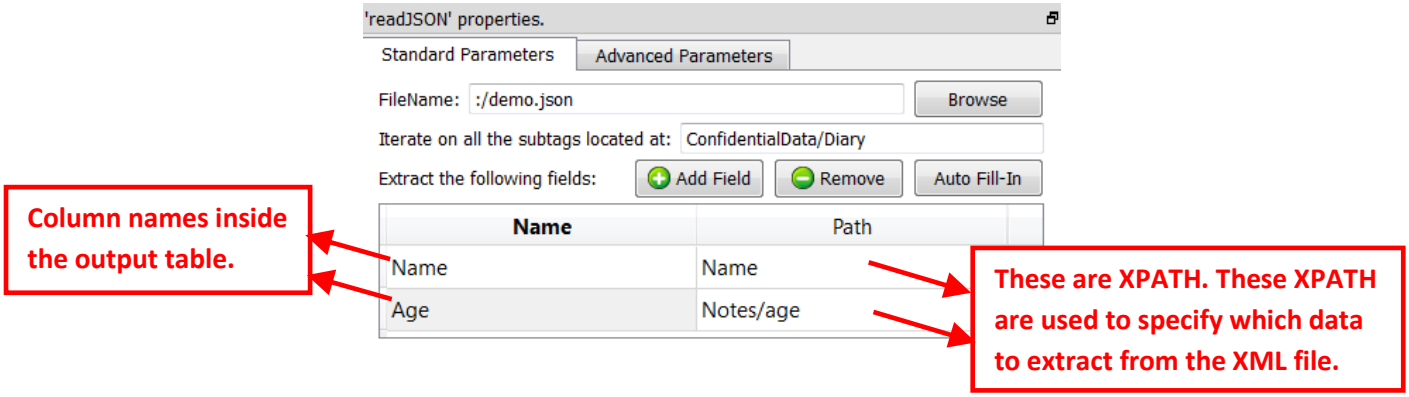
Usually, for classical "real world" JSON files, the compression/ratio is around 90%. Thus, it makes a lot of sense to compress all your JSON files.
Let's assume that we have the following JSON file:

```
{ "ConfidentialData":
  { "Revision":3,
    "Diary":
    [
      { "Name":"Frank",
        "Address":{"street":"villers","town": "ath", "zip":7812},
        "Notes": { "age": 38 },
        "Skills": [ "Coding", "Datamining" ]
      },
      { "Name":"Sabrina",
        "Address":{"street":"jeanne","town":"ixelles", "zip":1050},
        "Notes": { "age": 36 },
        "Skills": [ "Coding" ]
      }
    ]
  }
}
```

We want to extract the name and age of each of your contact.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We'll use the following settings:



We'll get as output the following table:

Name	Age
Frank	38
Sabrina	36

It can sometime be difficult to manually write the different required XPATHs. This is why Anatella has an "Auto Fill-In" button: After you finished entering the "*Iterate on all subtags located at*" parameter, you can click the "*Auto Fill-In*" button. Anatella will analyzes the first 100 entries in the "`Diary`" (You can change this number using the "*Number of Rows to Analyze for Auto Fill-In*" parameter inside the "*Advanced Parameters*" tab) and extract all the different XPATH to all the different data contained inside these first 100 entries. For the above example, Anatella will find the following XPATHs:

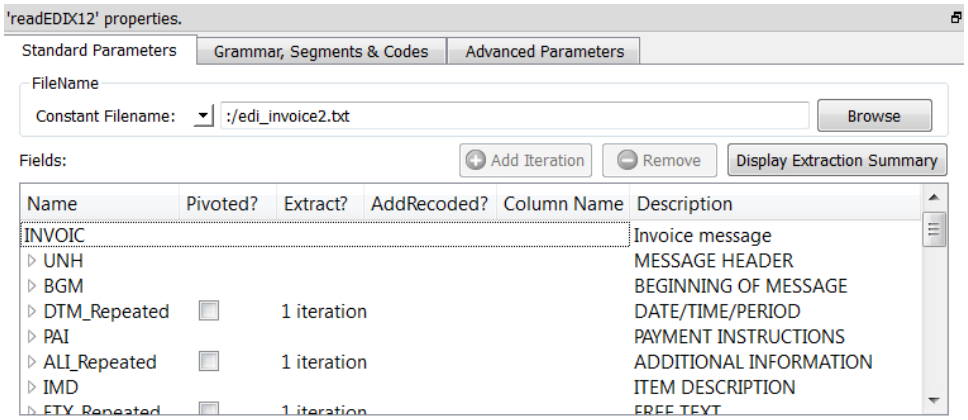
Name
Address/street
Address/town
Address/zip
Notes/age
Skills/[0]
Skills/[1]

5.2.12. Edi-Fact / X12 Reader

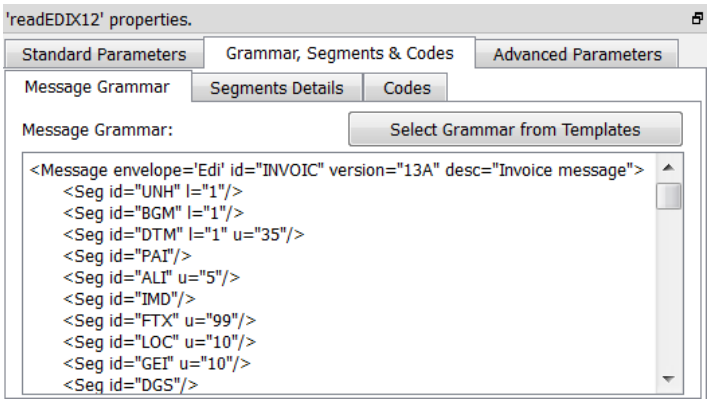
Icon:

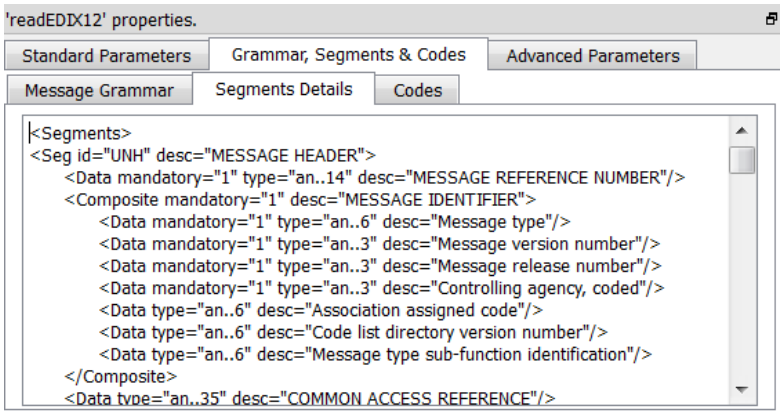
Property window:

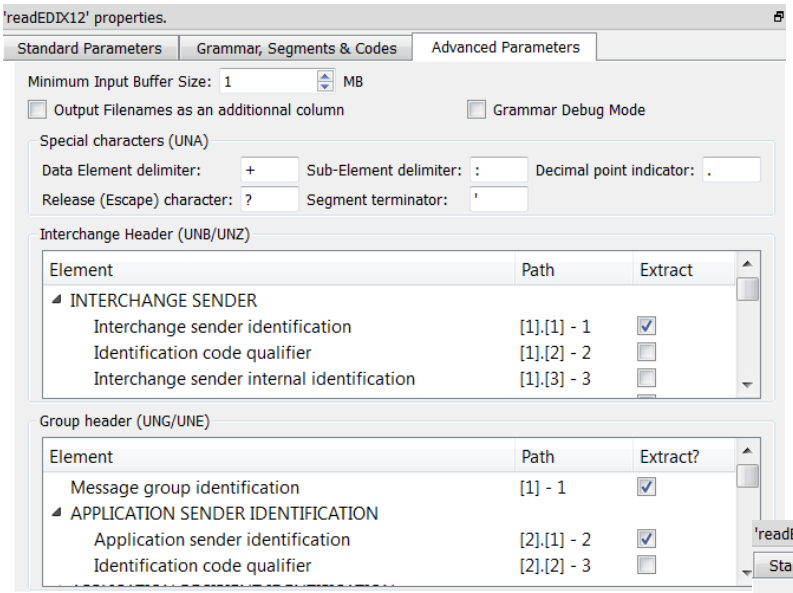
EDI mode:

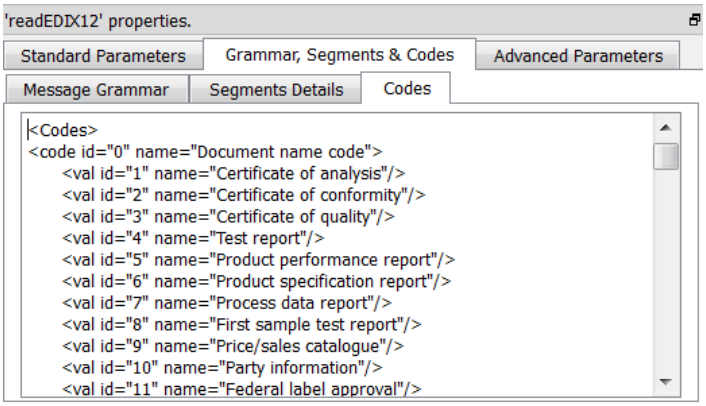


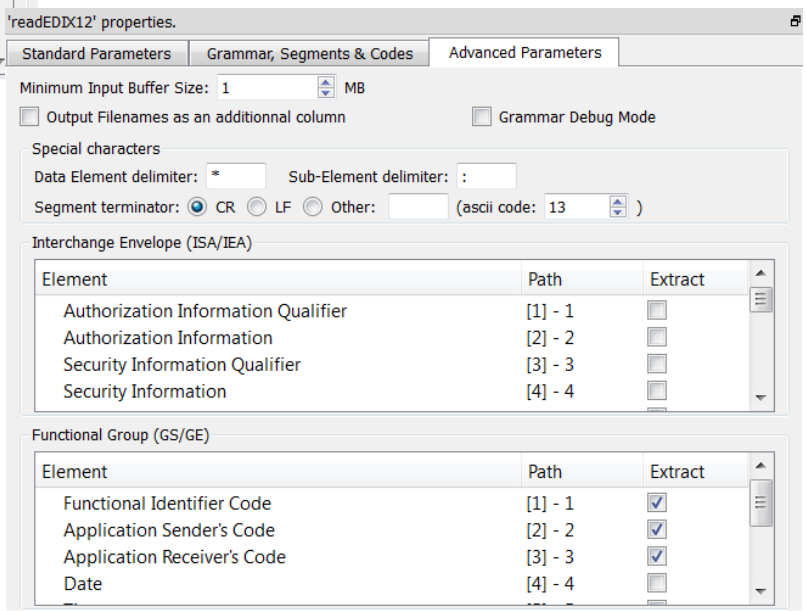
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu


'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
Message Grammar
Segments Details
Codes
Message Grammar:
Select Grammar from Templates
<Message envelope='Edi' id="INVOIC" version="13A" desc="Invoice message">
<Seg id="UNH" l="1"/>
<Seg id="BGM" l="1"/>
<Seg id="DTM" l="1" u="35"/>
<Seg id="PAI"/>
<Seg id="ALI" u="5"/>
<Seg id="IMD"/>
<Seg id="FTX" u="99"/>
<Seg id="LOC" u="10"/>
<Seg id="GEI" u="10"/>
<Seg id="DGS"/>

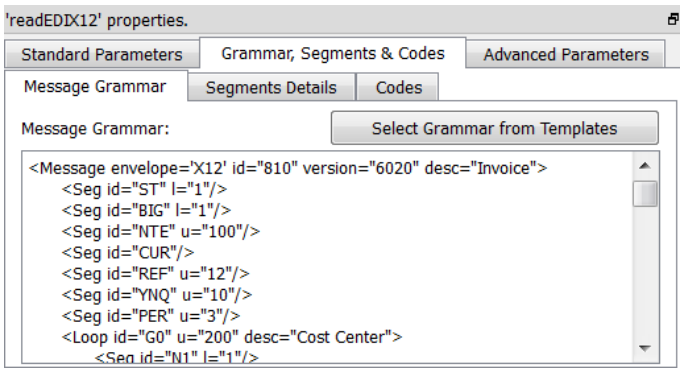

'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
Message Grammar
Segments Details
Codes
<Segments>
<Seg id="UNH" desc="MESSAGE HEADER">
<Data mandatory="1" type="an..14" desc="MESSAGE REFERENCE NUMBER"/>
<Composite mandatory="1" desc="MESSAGE IDENTIFIER">
<Data mandatory="1" type="an..6" desc="Message type"/>
<Data mandatory="1" type="an..3" desc="Message version number"/>
<Data mandatory="1" type="an..3" desc="Message release number"/>
<Data mandatory="1" type="an..3" desc="Controlling agency, coded"/>
<Data type="an..6" desc="Association assigned code"/>
<Data type="an..6" desc="Code list directory version number"/>
<Data type="an..6" desc="Message type sub-function identification"/>
</Composite>
<Data type="an..35" desc="COMMON ACCESS REFERENCE"/>

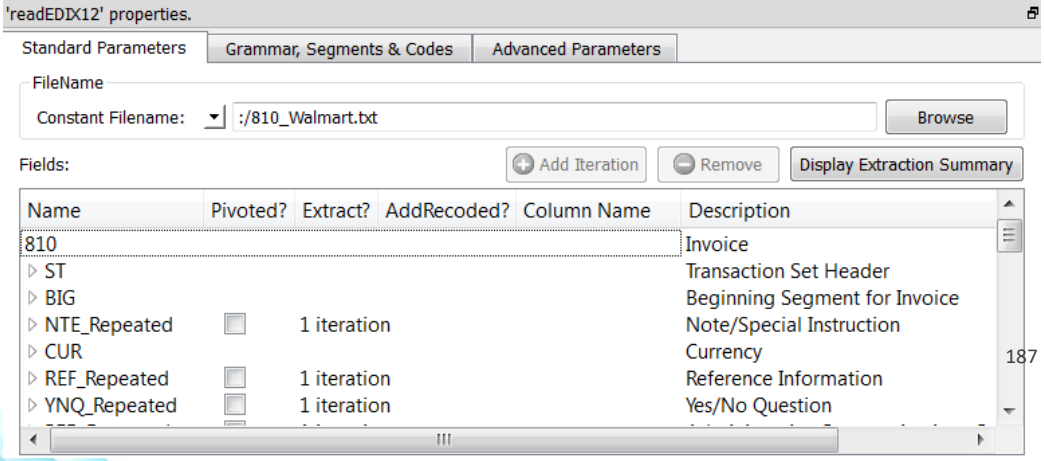

'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
Minimum Input Buffer Size: 1 MB
Output Filenames as an additionnal column
Grammar Debug Mode
Special characters (UNA)
Data Element delimiter: +
Sub-Element delimiter: :
Decimal point indicator: .
Release (Escape) character: ?
Segment terminator: '
Interchange Header (UNB/UNZ)
Element
Path
Extract
INTERCHANGE SENDER
Interchange sender identification
[1].[1] - 1
Identification code qualifier
[1].[2] - 2
Interchange sender internal identification
[1].[3] - 3
Group header (UNG/UNE)
Element
Path
Extract?
Message group identification
[1] - 1
APPLICATION SENDER IDENTIFICATION
Application sender identification
[2].[1] - 2
Identification code qualifier
[2].[2] - 3


'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
Message Grammar
Segments Details
Codes
<Codes>
<code id="0" name="Document name code">
<val id="1" name="Certificate of analysis"/>
<val id="2" name="Certificate of conformity"/>
<val id="3" name="Certificate of quality"/>
<val id="4" name="Test report"/>
<val id="5" name="Product performance report"/>
<val id="6" name="Product specification report"/>
<val id="7" name="Process data report"/>
<val id="8" name="First sample test report"/>
<val id="9" name="Price/sales catalogue"/>
<val id="10" name="Party information"/>
<val id="11" name="Federal label approval"/>


'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
Minimum Input Buffer Size: 1 MB
Output Filenames as an additionnal column
Grammar Debug Mode
Special characters
Data Element delimiter: *
Sub-Element delimiter: :
Segment terminator: CR LF Other: (ascii code: 13)
Interchange Envelope (ISA/IEA)
Element
Path
Extract
Authorization Information Qualifier
[1] - 1
Authorization Information
[2] - 2
Security Information Qualifier
[3] - 3
Security Information
[4] - 4
Functional Group (GS/GE)
Element
Path
Extract
Functional Identifier Code
[1] - 1
Application Sender's Code
[2] - 2
Application Receiver's Code
[3] - 3
Date
[4] - 4

X12 mode:


'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
Message Grammar
Segments Details
Codes
Message Grammar:
Select Grammar from Templates
<Message envelope='X12' id="810" version="6020" desc="Invoice">
<Seg id="ST" l="1"/>
<Seg id="BIG" l="1"/>
<Seg id="NTE" u="100"/>
<Seg id="CUR"/>
<Seg id="REF" u="12"/>
<Seg id="YNQ" u="10"/>
<Seg id="PER" u="3"/>
<Loop id="G0" u="200" desc="Cost Center">
<Seg id="N1" l="1"/>


'readEDIX12' properties.
Standard Parameters
Grammar, Segments & Codes
Advanced Parameters
FileName
Constant Filename: :/810_Walmart.txt
Browse
Fields:
Add Iteration
Remove
Display Extraction Summary
Name
Pivoted?
Extract?
AddRecoded?
Column Name
Description
810
Invoice
ST
Transaction Set Header
BIG
Beginning Segment for Invoice
NTE_Repeated
1 iteration
Note/Special Instruction
CUR
Currency
REF_Repeated
1 iteration
Reference Information
YNQ_Repeated
1 iteration
Yes/No Question

Short description:
Extract data from (compressed) EDI/X12 files.

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the EDI or X12 file (i.e. You can read the filenames from the input pin. You can also use relative path, wildcards, and Javascript to specify your filenames).

You can drag&drop a .x12 file or .edi file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadEdiX12 Action inside the Anatella graph.

The EDI/X12 reader included inside Anatella is stream-oriented. This means that Anatella reads the EDI/X12 file "chunk-by-chunk" (when all data from one chunk has been extracted, Anatella loads the next chunk). This means that:

- There are no limits to the size of the EDI/X12 file that Anatella can read&parse.
- With Anatella, Data Extraction from EDI/X12 only requires a small (and constant) amount of RAM.
- With Anatella, data Extraction from EDI/X12 is very fast.

If the extension of the EDI/X12 file is RAR, ZIP, GZ or LZO then Anatella will transparently decompress the XML file in RAM memory. Anatella chooses the (de)compression technique to use based on the filename extension. When Anatella uses compressed file formats, it does NOT decompress the files on the Hard Drive: Anatella decompresses the data "on-the-fly" in central core RAM memory, thus reducing:

- the load on the hard drive
- the hard-drive consumption required to do the analysis.

Usually, for classical "real world" EDI/X12 files, the compression/ratio is around 90%. Thus, it makes a lot of sense to compress your EDI/X12 files.

5.2.12.1. Background information about EDI/X12 files

EDI/X12 files are text files that contains a tree data structure (much like XML or JSON). The ReadEdiX12 Action creates a simple table from the EDI/X12 files: i.e. it 'flattens" the tree-structure to obtain a simple table.

An EDI/X12 file is composed of different "Segments". Each Segment is separated from the next one using a special character: the "Segment Terminator".

	Default Segment Terminator
EDIFACT	Quote '
X12	Caret ^ or Line Feed (LF) or both

Each segment is composed of several "Data Elements". The different "Data Elements" are separated using a special character: the "Data Element Delimiter".

	Default Data Element Delimiter
EDIFACT	Plus +
X12	Star *

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The first “Data Element” of a segment contains the Segment Name. Some “Data Elements” (but not all) are composed of several (sub)fields. These special “Data Elements” are named “Composite Data Elements”. Each field inside a “Composite Data Element” is is separated from the next field using a special character: the “Sub-Element Delimiter”.

	Default Sub-Element Delimiter
EDIFACT	colon :
X12	colon :

Here is an example of EDI/X12 file (we added a CR/LF after each segment to make it easier to read):

EdiFact	X12
The following example is an EDI file that describes a PURCHASE ORDER. It contains the following segments: UNH, BGM, NAD, LIN, QTY, PRI, UNS, CNT, UNT. UNH+SSDD1+ORDERS:D:03B:UN:EAN008’ BGM+220+4768+9’ DTM+137:20140930:102’ NAD+BY+541234::9++XYZ Company+123 Church Street+Brussels+BX+1050+BE’ LIN+1+1+ID-12AB+VN’ QTY+1:100:EA’ PRI+AAA:2765’ UNS+S’ CNT+2:100’ UNT+10+SSDD1’	The following example is an EDI file that describes a PURCHASE ORDER (850). It contains the following segments: ST, BEG, N1, N3, N4, PO1, STT, SE. ST*850*54001^ BEG*00*SA*4768*65*20140930^ N1*SO*XYZ Company^ N3*123 Church Street^ N4*Brussels*BE*1050^ PO1*1*100*EA*27.65**VN*ID-12AB^ CTT*1*100^ SE*9*54001^

Inside an EDI file, the NAD segment contains an address.
In the example above the addess is:

XYZ Company
123 Church Street
1050 Brussels – BE

If the street name contains a special characters (such as the ‘, the + or the :), we need to “escape” it (so that it’s not interpreted as a “Segment Terminator”, a “Data Element Separator” or a “Sub-Element Separator”). The default “Escape” character is the “Question Mark”(?).
For example, if the street name is:

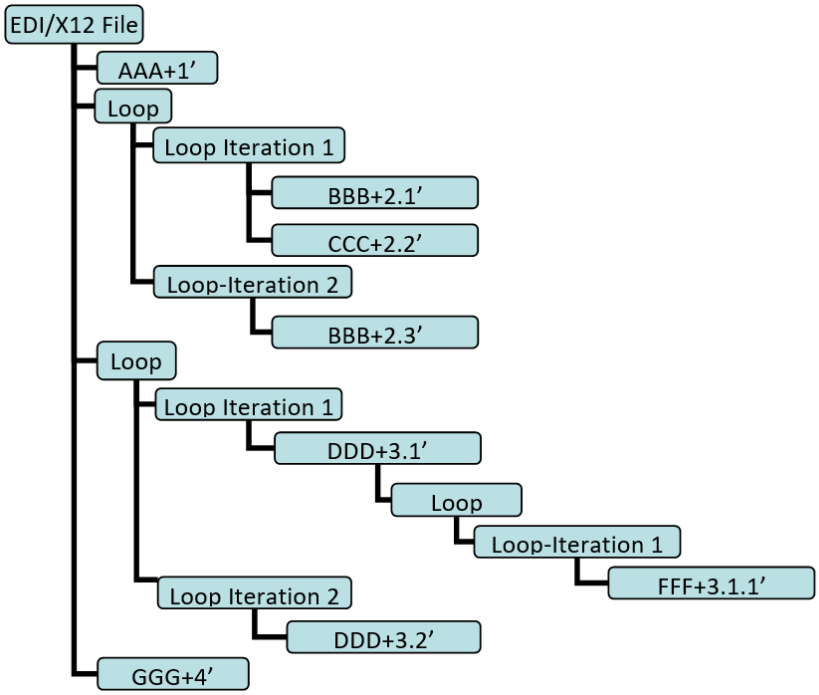
2+ Is it Bob? Street

… the EDI file contains:

2?+ Is it Bob?? Street

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's now illustrate the tree structure of an EDI/X12 file.
Let's create an EDI/X12 file that represents the following tree:



The corresponding EDI/X12 file is (we added CR/LF and spaces after each segment to make it easier to read):

EDI	X12
AAA+1'	AAA*1^
BBB+2.1'	BBB*2.1^
CCC+2.2'	CCC*2.2^
BBB+2.3'	BBB*2.3^
DDD+3.1'	DDD*3.1^
FFF+3.1.1'	FFF*3.1.1^
DDD+3.2'	DDD*3.2^
HHH+4'	HHH*4^

The corresponding XML or JSON file is:

XML File	JSON File
<File> <AAA v='1'/> <Loop> <Iteration> <BBB v='2.1'/> <CCC v='2.2'/> </Iteration> <Iteration> <BBB v='2.3'/> </Iteration> </Loop>	{ "File": { "AAA": ["1"], "Loop": [{ "BBB": ["2.1"], "CCC", ["2.2"] }, { "BBB": ["2.3"] }], "Loop":

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

<Loop> <Iteration> <DDD v='3.1'/> <Loop> <Iteration> <FFF v='3.1.1'/> </Iteration> </Loop> </Iteration> <Iteration> <DDD v='3.2'/> </Iteration> </Loop> <HHH v='4'/> </File>	[{ "DDD": ["3.1"], "Loop": [{ "FFF": ["3.1.1"] }] }, { "DDD": ["3.2"] }], "HHH" : ["4"] } }

A difficulty with EDI/X12 files is that there are no tags inside the file that "close" a level: i.e. There are no "</loop>" tags (i.e. no "end-of-loop" tags), no "</iteration>" tags (no "end-of-iteration" tags) (…or in JSON: There are no "]" or "}").

Inside an EDI/X12 file, to know if you need to either "go down" a hierarchical level, either "close" a hierarchical level (i.e. "go up") or either start a new iteration at the same hierarchical level, you need to refer to the grammar of the EDI/X12 file. Without the grammar, you cannot properly decipher an EDI/X12 file. Here is the grammar used to parse the above file (note that this grammar refers to the segments "EEE" and "GGG" but the example file did not contain any such segments, since they are optional):

```
<Message envelope='Edi'>
  <Seg id='AAA'/>
  <Loop u='-1'>
      <Seg id='BBB'/>
      <Seg id='CCC'/>
  </Loop>
  <Loop u='-1'>
      <Seg id='DDD'/>
      <Loop u='-1'>
          <Seg id='FFF'/>
          <Seg id='GGG'/>
      </Loop>
      <Seg id='EEE'/>
  </Loop>
  <Seg id='HHH'/>
</Message>
```

The grammar contains two type of tags: "Seg" tags and "Loop" tags.

The "Seg" tags indicate that the EDI/X12 file might contain here a specific segment, with a specific name. The attributes of a "Seg" tag are:

- "id": the name of the segment

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- "l" (optional: if omitted, l=0): if l=1, then the segment if mandatory (i.e. it must appear inside the EDI/X12 file). "l" stands of "lower bound".
- "u" (optional: if omitted, u=1): The number of time this segment might be repeated inside the ED/XC12 file. If u=-1, then an unlimited number of repetition is authorized. "u" stands of "upper bound".

The "Loop" tag indicates that the EDI/X12 file might contain here a loop (i.e. a repetition of several segments). "Loops" are sometime refered as "Segment groups" inside Edi-Fact documentation. You can have a Loop within a Loop. The attributes of a "Loop" tag are:

- "id": the name of the loop. Usefull only for debugging purposes. It's not used to parse the EDI/X12 file. It's also used when saving the .anatella file (The XPath expressions used to describe the fields to extract out of the EDI/X12 file are using the loop's names).
- "l" (optional: if omitted, l=0): if l=1, then the loop if mandatory (i.e. the loop must appear inside the EDI/X12) file.
- "u" (optional: if omitted, u=1): The number of time this loop might be repeated inside the ED/XC12 file. If u=-1, then an unlimited number of repetition is authorized.

The first segment inside a Loop is special: It marks:

- The start of the loop (i.e. We "go down" one hierarchical level).
- The start of a new iteration of the loop (i.e. We stay at the same hierarchical level and we start a new iteration of the loop)

For example, if we have the following grammar:

```
<Seg id='AAA'>
<Loop id='G0'>
    <Seg id='BBB'/>
    <Seg id='CCC'/>
</Loop>
<Seg id='CCC'/>
```

… and the following EDI files:

AAA+1' CCC+3'	AAA+1' { BBB+2.1' CCC+2.2' CCC+3'	AAA+1' { BBB+2.1' CCC+2.2' { BBB+2.3' CCC+2.4' CCC+3'
We don't enter the loop at all because we din't find the 'BBB' segment that marks the start of the loop.	We enter the loop "G0" only one time because, for the second iteration, we din't find the 'BBB' segment. In other words, the 'BBB' segment (that marks the start of the second iteration) is missing.	We make two interations of the loop "G0".

The following grammar:

```
<Seg id='AAA'>
<Seg id='BBB'>
<Seg id='CCC'>
```

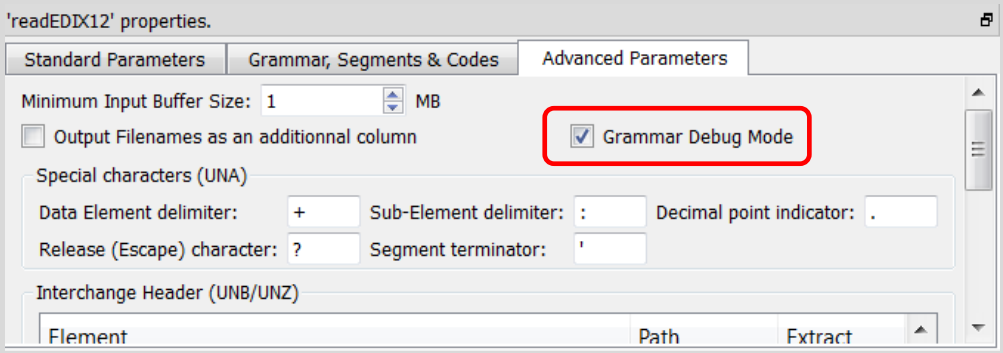
…won't be able to parse this file:

```
AAA+1'
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

CCC+1’
BBB+1’
…because Anatella expects the segment “BBB” after the segment “AAA” (and not the segment “CCC”).

In such situation, you will get a parsing error. To solve the error, you need to correct the grammar. (The correct grammar for the above example is:
<Seg id=’AAA’> <Seg id=’CCC’> <Seg id=’BBB’>).To easily find what corrections are required inside the grammar, you can activate the “Grammar Debug Mode”:



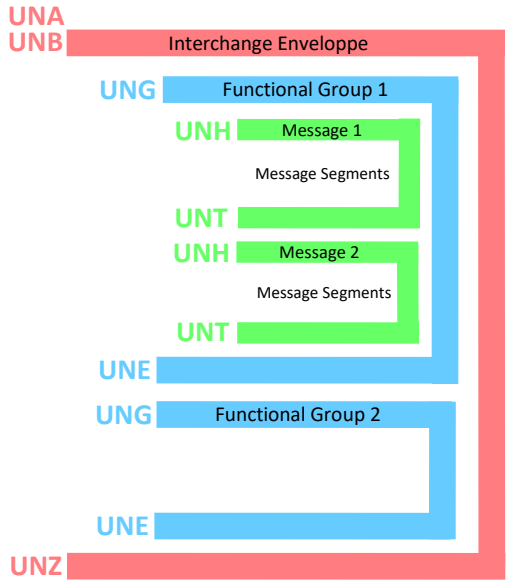
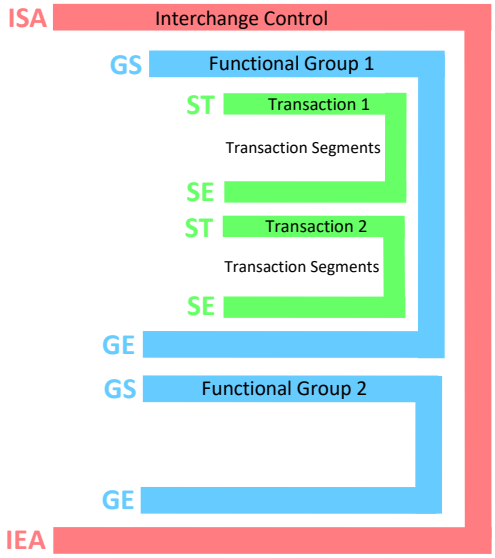
When the “*Grammar Debug Mode*” is activated, you’ll see inside the Anatella Log window some messages that gives you a precise explanation of the parser state (it explains you at which position inside the hierarchical structure of the document the parser is). You can see when the EDI/X12 parser “goes down” or “goes up” one hierarchical level, when it starts a new iteration, etc.

For example:

```
EDI/X12: parsing segment UNB - extract 5 element(s).
EDI/X12: parsing segment UNH -nothing to extract.
EDI/X12: parsing segment BGM -nothing to extract.
EDI/X12: parsing segment DTM -nothing to extract.
EDI/X12: skipping segment DTM
EDI/X12: skipping segment DTM
EDI/X12: Entering Loop 'G0' starting with 'RFF'
EDI/X12: parsing segment RFF -nothing to extract.
EDI/X12: Leaving Loop 'G0'
EDI/X12: Entering Loop 'G1' starting with 'NAD'
EDI/X12: parsing segment NAD -nothing to extract.
EDI/X12: Next iteration (1) of Loop 'G1' starting with 'NAD'
EDI/X12: skipping segment NAD
EDI/X12: Entering Loop 'G2' starting with 'RFF'
EDI/X12: skipping segment RFF
EDI/X12: Leaving Loop 'G2'
EDI/X12: Next iteration (2) of Loop 'G1' starting with 'NAD'
EDI/X12: skipping segment NAD
EDI/X12: Next iteration (3) of Loop 'G1' starting with 'NAD'
EDI/X12: skipping segment NAD
EDI/X12: Entering Loop 'G2' starting with 'RFF'
EDI/X12: skipping segment RFF
EDI/X12: Leaving Loop 'G2'
EDI/X12: Leaving Loop 'G1'
EDI/X12: Entering Loop 'G6' starting with 'CUX'
EDI/X12: parsing segment CUX -nothing to extract.
EDI/X12: Leaving Loop 'G6'
…
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Once all the segments are assembled according to a grammar, they form a complete electronic document. When you need to transfer several of these electronic documents, you can include them inside a single file. The procedure is named “enveloping”. The envelope data is composed of different additional segments that precedes and follows all the electronic documents. This is illustrated on the following charts:

<table>
<tr><th>EDI Enveloping</th><th>X12 Enveloping</th></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>Enveloppe Levels:
1. Message (UNH/UNT)
2. Functional Group (UNG/UNE)
3. Interchange (UNA/UNB/UNZ)</td><td>Enveloppe Levels:
1. Transaction Set (ST/SE)
2. Functional Group (GS/GE)
3. Interchange (ISA/IEA)</td></tr>
<tr><td>Example:

UNB+UNOB:4+SENDER:ZZZ+RECEIVER:ZZZ+
20140930:1159+6002’
UNG+ORDERS+SENDER:ZZZ+RECEIVER:ZZZ+
20140930:1159+9+UN+D:038’
UNH+SSDD1+ORDERS:D:03B:UN:EAN008’
BGM+220+4768+9’
DTM+137:20140930:102’
NAD+BY+541234::9++XYZ Company+123
Church
Street+Brussels+BX+1050+BE’
LIN+1+1+ID-12AB+VN’
QTY+1:100:EA’
PRI+AAA:2765’
UNS+S’
CNT+2:100’
UNT+10+SSDD1’
UNE+1+9’
UNZ+1+6002’</td><td>Example:

ISA*00* *00* *ZZ*SENDER *ZZ*RECEIVER
*03092014*2052*U*00401*00114*O*P^
GS*PO*SENDER*RECEIVER*20140930*252
*702*X*004010^
ST*850*54001^
BEG*00*SA*4768*65*20140930^
N1*SO*XYZ Company^
N3*123 Church Street^
N4*Brussels*BE*1050^
PO1*1*100*EA*27.65**VN*ID-12AB^
CTT*1*100^
SE*8*54001^
GE*1*702^
IEA*1*00114^</td></tr>
</table>

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Functional Groups, often referred to as the "inner envelope," are made up of one or more documents, all of the same type, which can be batched together into one file. Functional Groups are optional in EDI and (often) mandatory in X12.

If present, the UNA segment inside EDI files sets the values of the special characters (i.e. It sets the value of the "Segment Terminator", the "Data Element Separator", etc. for the current file).

5.2.12.2. Extacting Data from EDI/X12 files with Anatella

Warning: Anatella does not support EDI files that:

• … contains encrypted segments (i.e. EDI files that contains USA, USC and USH segments).

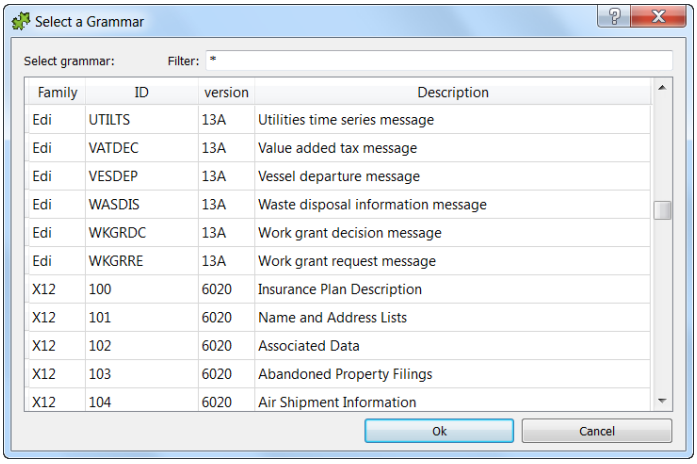
• … contains segments with binary data (i.e. EDI files that contains BIN or BDS segments): If you want to add binary data to EDI files, use Base64 encoding.

These segments are very uncommon and only appears in very old grammars.

Warning: Anatella does not support X12 files that have "Repetition Separators" with a repetition factor greater than 1. "Repetition Separator" are currently defined inside the X12 standard but they are never used with a repetition factor greater than 1, so we are perfectly safe here.

The first step is to specify the filenames of your EDI or X12 files: See section 5.1.1 to have more information on this subject (i.e. You can read the filenames from the input pin. You can also use relative path, wildcards, and Javascript to specify your filename(s)).

The second step is to select the Grammar: Go to the second panel (named "*Grammar, Segments & Codes*") and click the button "*Select Grammar from Templates*". You'll see the following window:



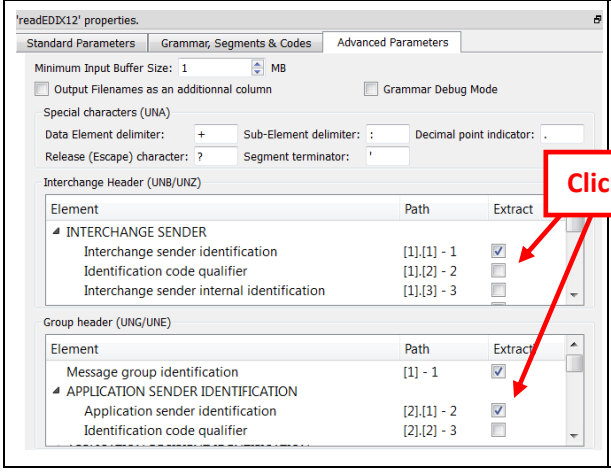
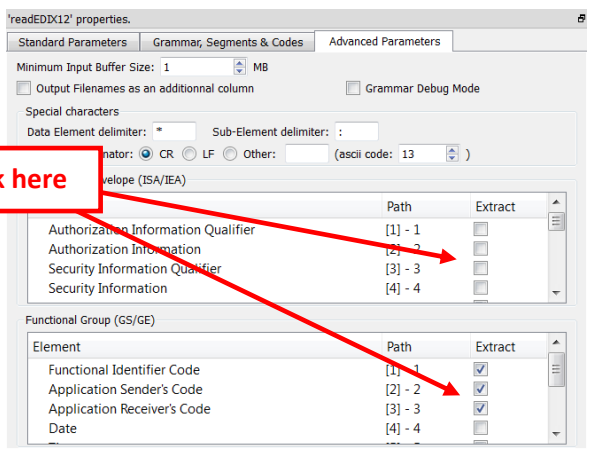
Inside Anatella, there are currently nearly 200 "EDI grammars" and more than 300 "X12 grammars" to choose from. So, you are nearly assured to find the exact grammar that you need. If you don't find your grammar, don't worry: You can still easily add manually later a new grammar (or edit/tweak an existing grammar so that it matches your exact file's structure).

Select your grammar (click the required row inside the table) and click "Ok".

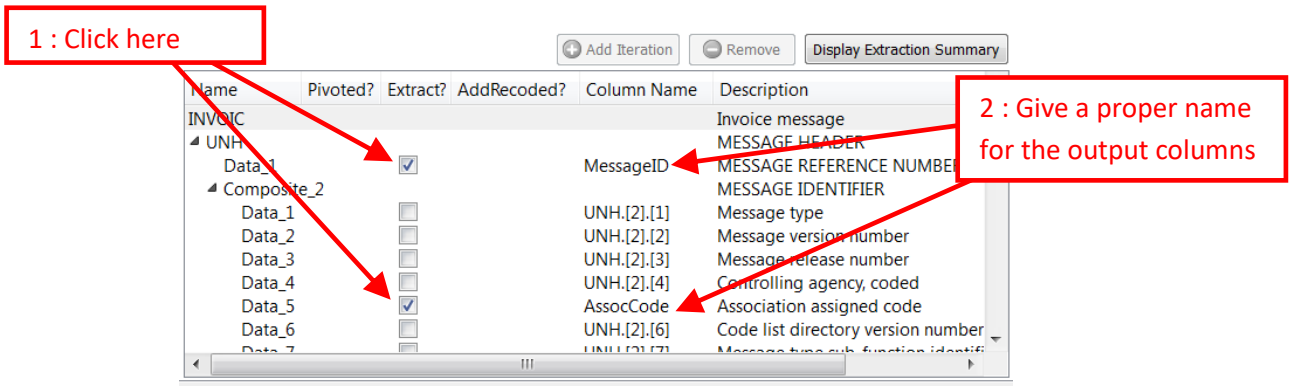
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The third step is to select the fields that you want to extract out of the EDI/X12 files. There are 2 types of fields that you can extract:

- Fields extracted from the "Enveloppe" data (i.e. from the UNB&UNG segments for EDI or from the ISA&GS segments for X12). You can specify these fields in the third panel (that is named "*Advanced Parameters*"):

EDI	X12
	

- Fields Extracted from the Message Segments (These fields depend on the selected grammar). You can specify these fields in the first panel (that is named "Standard Parameters"): For example: Let's assume that you want to extract the fields "Message Reference Number" and "Association Assigned Code" out of each of your messages: We'll have:



We get the following output table:

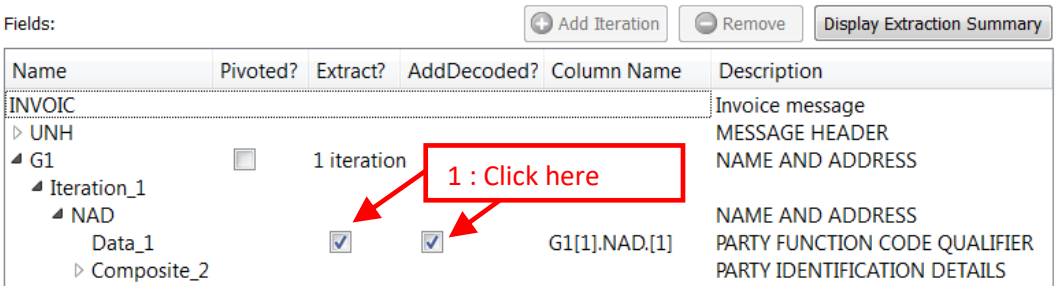
	MessageID	AssocCode
1	1	EAN008
2	2	EAN008
3	3	EAN008
4	4	EAN008

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's now look at a more complex EDI/X12 extraction example:

EDI Grammar	EDI file to parse
<Message envelope='Edi' id="INVOIC" desc="Invoice"> <Seg id="UNH" l="1"/> <Loop id="G1" u="99" desc="NAD - Name and address"> <Seg id="NAD" l="1"/> <Loop id="G2" u="9999" desc="RFF - Reference"> <Seg id="RFF" l="1"/> <Seg id="DTM" u="5"/> </Loop> </Loop> <Seg id="CUX" l="1"/> </Message>	UNB+UNOC:3+ss:14+dd:14+11:0345+1016' UNH+1+INVOIC:D:96A:UN:EAN008' NAD+BY+1111' } Iteration 1 of the Loop "G1" NAD+SU+2222' } Iteration 2 of the Loop "G1" RFF+VA:2222.2' NAD+DP+3333' } Iteration 3 of the Loop "G1" NAD+IV+4444' } Iteration 4 of the Loop "G1" RFF+VA:4444.4' RFF+VA:5555.5' CUX+2:EUR:4' UNT+9+4' UNZ+1+1016'

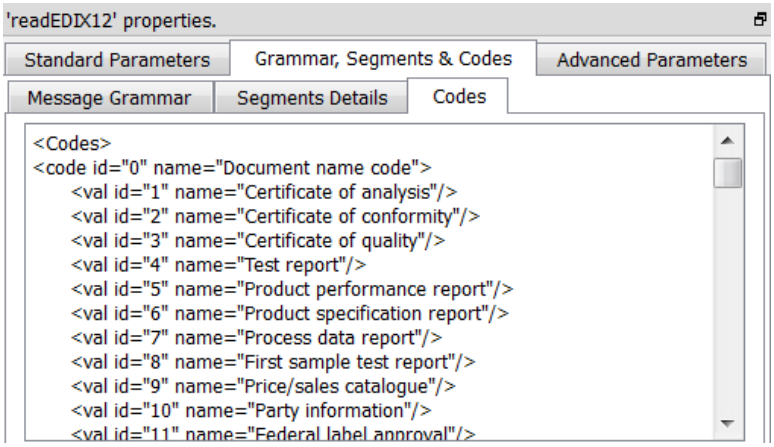
We want to extract the string "BY": i.e. We want to extract the first "Data-Element" of the NAD Segment inside the first iteration of the Loop "G1" (that starts with the NAD Segment): We'll have:



We get the following output table:

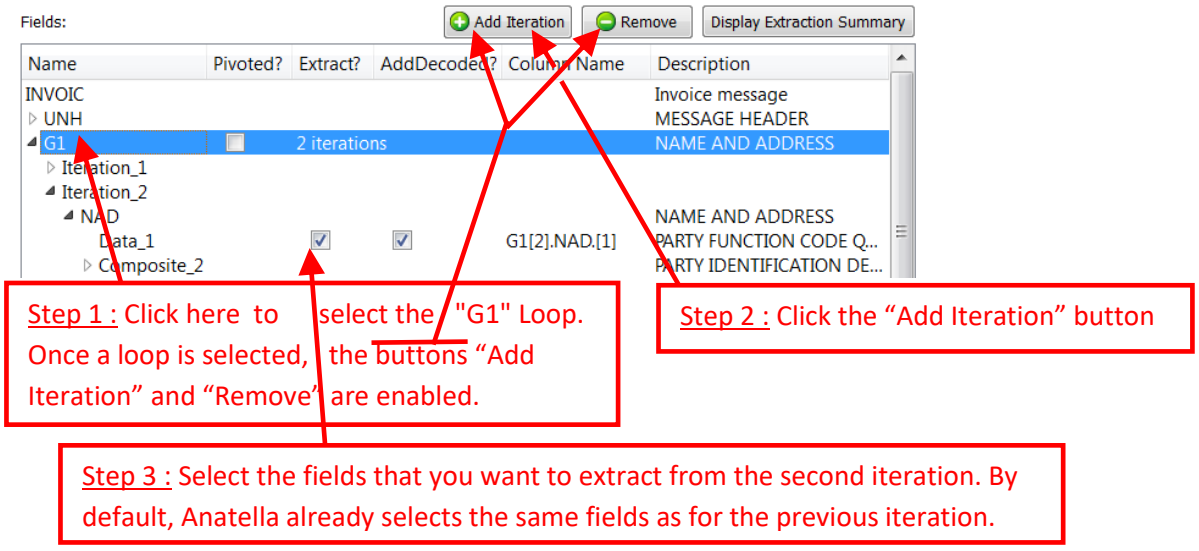
	G1[1].NAD.[1]	G1[1].NAD.[1]_decoded
1	BY	Buyer

Note that we asked to decode the string "BY" (to get "Buyer"). The "decoding tables" are editable: They are given inside the subpanel "Codes" inside the panel "Grammar, Segment & Codes":



Now, we want to also extract "SU": i.e. We want to extract the first "Data-Element" of the NAD Segment inside the **second iteration** of the Loop "G1" (that starts with the NAD Segment): We'll have:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

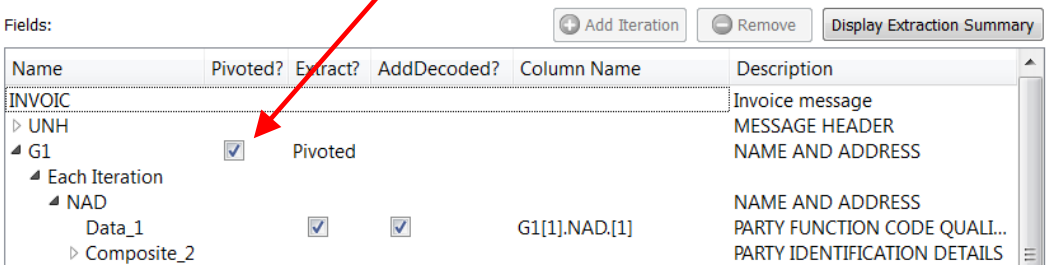


We get the following output table:

	G1[1].NAD.[1]	G1[1].NAD.[1]_decoded	G1[2].NAD.[1]	G1[2].NAD.[1]_decoded
1	BY	Buyer	SU	Supplier

Now, let's assume that we want to extract the first "Data-Element" of all the NAD Segments inside **all the iterations** of the "G1" Loop. There are two different approaches to this request:

1. We can click many times on the button "Add Iterations" (to add more iterations), until we have enough iterations to be (more or less) sure to get the data from all the iterations of the "G1" Loop. This is not a very good&efficient solution (and, also, it creates very wide output tables, and wide tables can be annoying from time to time).
2. We can click the "Pivoted" checkbox:

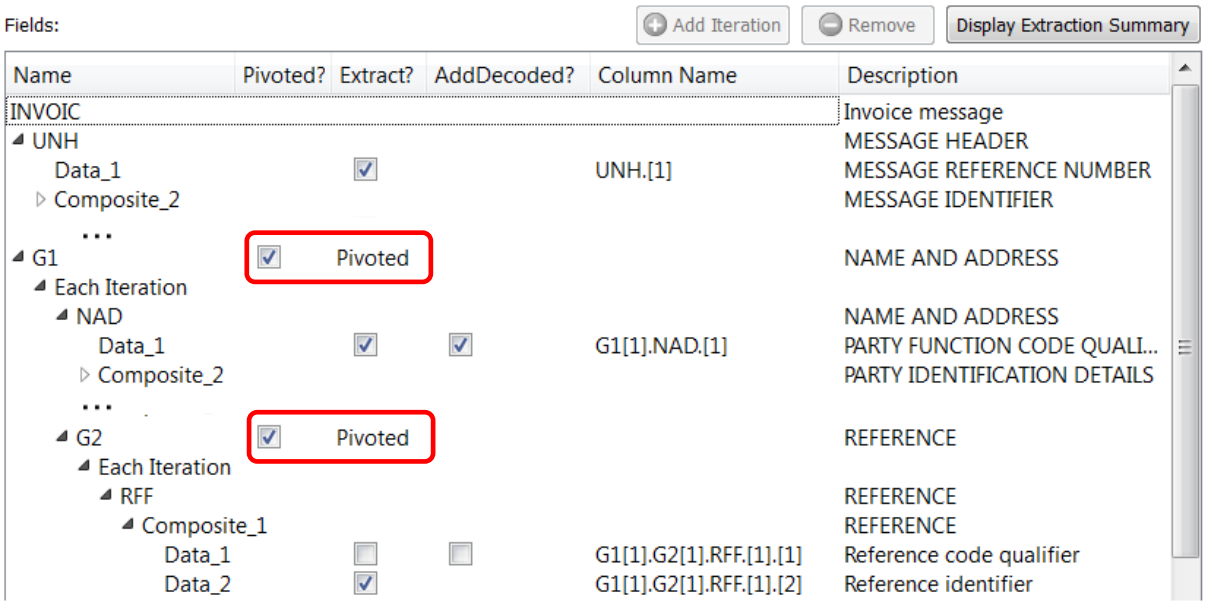


When the "*Pivoted?*" option is enabled, Anatella creates a new row inside the output table for each different iteration of the "G1" Loop. We get the following output table:

	G1[1].NAD.[1]	G1[1].NAD.[1]_decoded
1	BY	Buyer
2	SU	Supplier
3	DP	Delivery party
4	IV	Invoicee

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can check several times the pivoting option (at different level of the Hierarchy). For example:

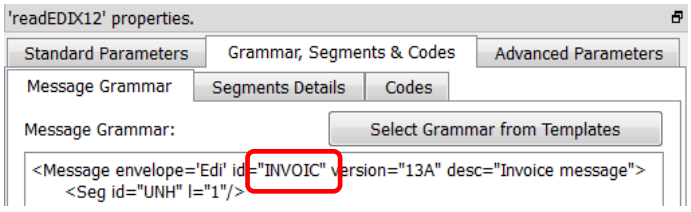


Anatella now creates a new row inside the output table for each different iteration of the **"G1" or "G2"** Loop. We get the following output table:

	UNH.[1]	G1[1].NAD.[1]	G1[1].NAD.[1]_decoded	G1[1].G2[1].RFF.[1].[2]
1	1	BY	Buyer	
2	1	SU	Supplier	2222.2
3	1	DP	Delivery party	
4	1	IV	Invoicee	4444.4
5	1	IV	Invoicee	5555.5

Defining several pivoting points is equivalent to defining several "Levels" inside the XML parser. See section 5.2.10.2 about an example of XML extraction with multiple levels.

Anatella is extracting data only from the messages whose name matches the name of the selected grammar. For example: If we have selected the grammar "INVOIC":



… then Anatella will only process the "INVOIC" messages (simply skipping the others).

This means that, to updates all the required tables inside your enterprise-level datawarehouse using the data originating from EDI/X12 files, you'll typically read several time the same EDI/X12 file(s) (using different grammars and different field's selection).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.12.3. Character Encoding in EDI/X12 files

The default character encoding used to read X12 files is "CP1252" (The "CP1252" character set is exactly like the standard "ISO-8859-1" but it contains, in addition, the Euro Symbol, that is quite useful). If the X12 file has a BOM (Byte-Order-Mark), then Anatella will use the character encoding specified inside the BOM (i.e. it will use UTF-8, UTF-16 or UTF32). If the X12 file does not contain any BOM, Anatella is still able to detect UTF-16 or UTF-32 files properly and open them accordingly.

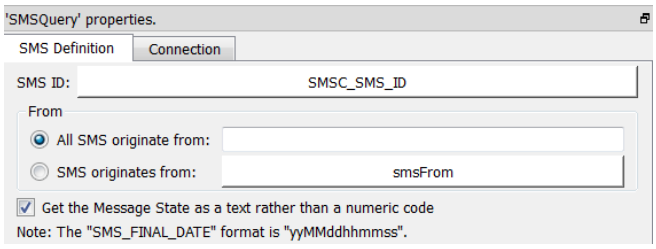
For the EDI files, the same rules as for the X12 files applies (see the previous paragraph). If there is no BOM and if the file is not UTF-16, nor UTF-32, then Anatella looks at the first "Data Element" of the "UNB" Segment:

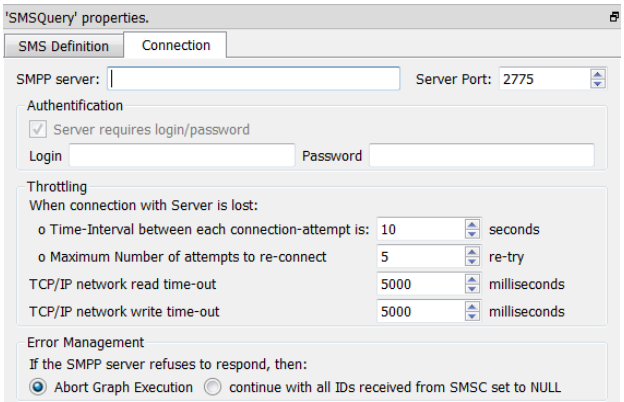
The First "Data Element" of the "UNB" Segment is:	Character Encoding used by Anatella to read the EDI file
UNOA, UNOB, UNOC	CP1252 (Latin1 - Western Europe and Americas)
UNOD	ISO-8859-2 (Latin2 - Slavic and Central European languages)
UNOE	ISO-8859-5 (Latin - Cyrillic)
UNOF	ISO-8859-7 (Latin - Greek)
UNOG	ISO-8859-3 (Latin3 - Esperanto, Galician, Maltese, and Turkish)
UNOH	ISO-8859-4 (Latin4 - Scandinavia/Baltic)
UNOI	ISO-8859-6 (Latin - Arabic)
UNOJ	ISO-8859-8 (Latin - Hebrew)
UNOK	ISO-8859-9 (Latin5 - Same as Latin1 except for Turkish)

5.2.13. Query SMS Delivery Status

Icon:

Property window:





Short description:
Query the SMS delivery Status.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Query the SMS delivery Status. The status of a SMS can be:

Status	Message	Description
NULL	NULL	The SMPP connection to the SMSC server failed.
1	ENROUTE	The message is in enroute state.
2	DELIVERED	Message is delivered to destination
3	EXPIRED	Message validity period has expired.
4	DELETED	Message has been deleted.
5	UNDELIVERABLE	Message is undeliverable
6	ACCEPTED	Message is in accepted state (i.e. it has been manually read on behalf of the subscriber by customer service)
7	UNKNOWN	Message is in invalid state
8	REJECTED	Message is in a rejected state
9	QUERY REQUEST FAILED	The SMSC server could not find your message (ID,Origin) amongst the known messages (IDs,Origins)

The SMSC server returns for the status of the SMS's that are specified inside the input table. The SMSC identifies precisely each SMS based on 2 criterions:

- The SMS ID (that is returned by the

SendSMSsmpp Action: See section 5.26.7)
- The origin of the SMS.

Thus, inside the

QuerySMS Action, you need to specify the SAME origin as the origin that was used inside the

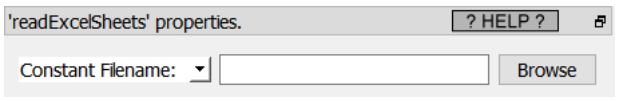
SendSMSsmpp Action (when sending the email). If the SMSC server does not find any SMS that matches the given ID and Origin, it returns Status=9 (QUERY REQUEST FAILED).

For each query to the SMSC server, you obtain 3 fields:

- **"SMS_STATE":** contains the status (ENROUTE, DELIVERED, EXPIRED, DELETED, UNDELIVERABLE, ACCEPTED, UNKNOWN, REJECTED, QUERY REQUEST FAILED)
- **"SMS_FINAL_DATE":** Date and time when the queried message reached a final state. For messages which have not yet reached a final state this field will contain a NULL.The date-time format is "YYMMDDhhmmss".
- **"SMS_ERROR_CODE":** Where appropriate this holds a network error code defining the reason for failure of message delivery. The range of values this field may have, depends entirely on the underlying telecommunications network.

5.2.14. Read Excel Sheets Names

Icon:

Property window: 

Short description:
Read Excel Sheet names

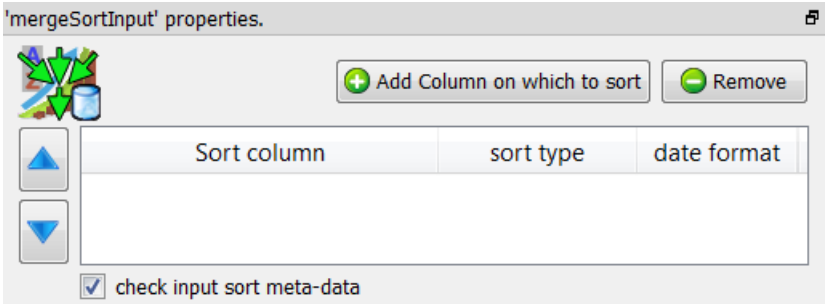
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Self-explanatory.

5.2.15. Merge Sort Input

Icon:

Property window:



Short description:
Merge several locally sorted .gel_anatella files into one globally sorted table.

Long Description:

The input pin of the the MergeSortInput Action is connected to a table that contains (many) .gel_anatella filenames. Typically, this input table will be computed using the fileListFromObsDate Action (see section 5.22.5).

Anatella reads all the corresponding "Gel files" simultaneously to create a globally sorted table as output (This is more or less equivalent to the Append Action but the Append Action does not generate a sorted output table and takes as input tables rather than filenames).

There are some limitations: The different "Gel Files" that are merged together must:

- …all have exactly the same meta-data.
- …all be sorted in the same way as the globally sorted table that the MergeSortInput Action creates as ouput (The sort in the output table is equal to the sort in the intput tables).

The MergeSortInput is:

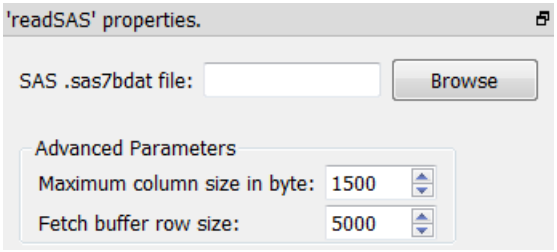
- …a faster alternative to the simple Sort Action: see sections 5.5.2.2. and 5.5.5.4. for more information about this subject.
- …about as fast as a simple Append Action: This means that the "sorting" is essentially performed "for free". This is great!
- …able to produce a sorted table of un-limited size.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The “Merge Sort” algorithm that is used to create the output table is described in section 5.5.2.1.

5.2.16. (old) SAS file reader

Icon:



Property window:

Short description:
Reads a table from a .sas7bdat file

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the .sas7bdat file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

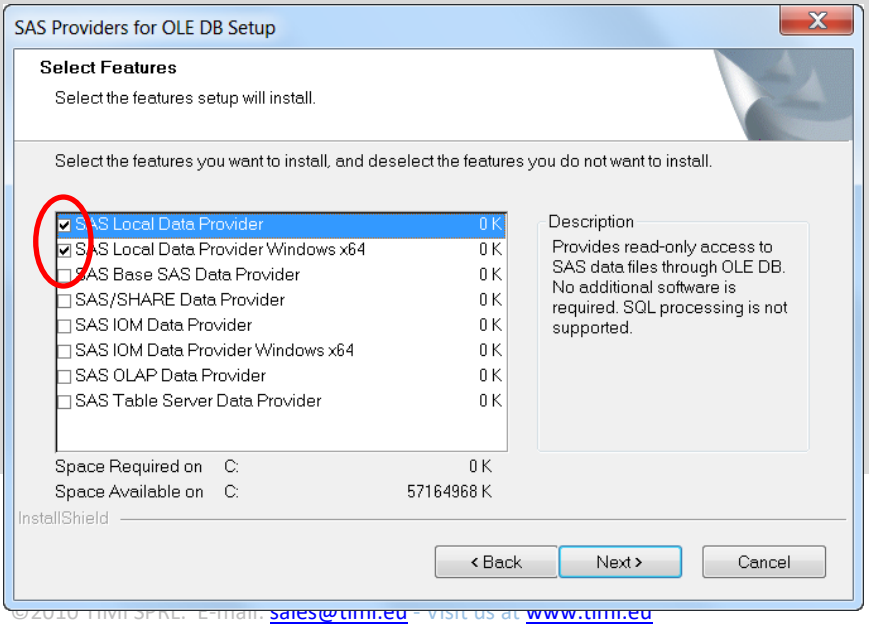
You should rather use the new readStat Action to read .sas7bdat files.

Pre-requisite

To be able to import SAS files inside Anatella using the readSAS Action, you need first to install the “*SAS OleDB provider*” inside your system. You need administrative rights to install the “*SAS OleDB provider*”. You don’t need to install the complete SAS software in your system to be able to use the “*SAS OleDB provider*” because it’s a complete stand-alone, self-contained driver. Installation is thus very easy: you only need a few mouse-clicks.

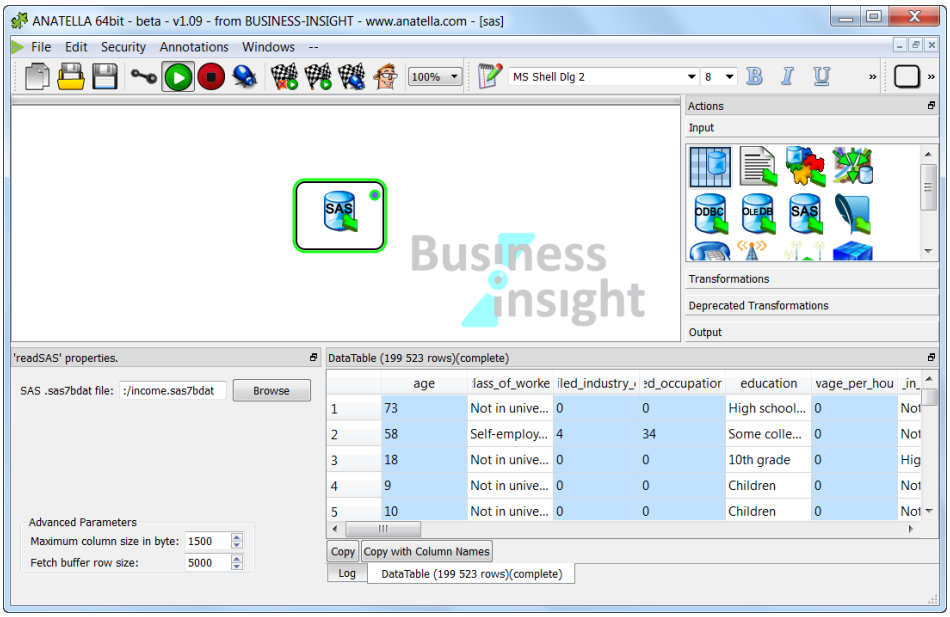
The “*SAS OleDB provider*” is a small & free driver available from the SAS website. For your convenience, a local copy is also available on the Anatella website, here:
http://download.timi.eu/ThirdParty/sasoledb_64bit_and_32bit.zip

To install the “*SAS OleDB provider*”, unzip the archive, double-click the “setup.exe” file and select these 2 options:



©2010 TIMI SPRL. E-mail: sales@timi.eu - visit us at www.timi.eu

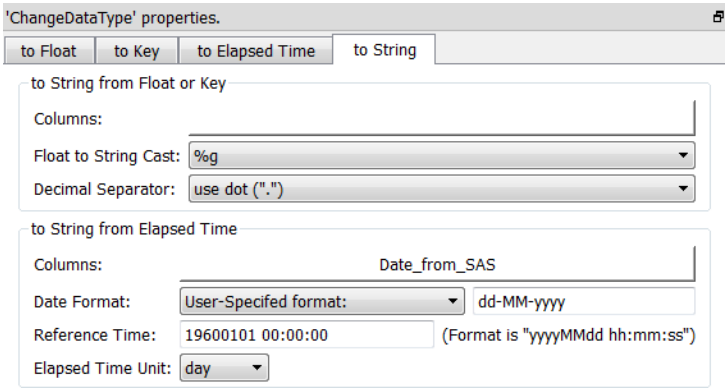
Here is an example:



There are two different ways to store dates inside SAS: Dates can either be stored as Strings or Floating point numbers (SAS is very similar to Anatella with respect to dates). The dates that are stored inside SAS as floating point numbers are imported inside Anatella as floating point numbers (i.e. with a blue background color inside the data Preview window). To convert these dates to "normal" Anatella dates, you can use the "*to String from Elapsed Time*" option of the ChangeDataType Action. Use these parameters:

- Reference Time: 19600101 00:00:00
- Elapsed Time Unit: day.

Here is a screenshot:

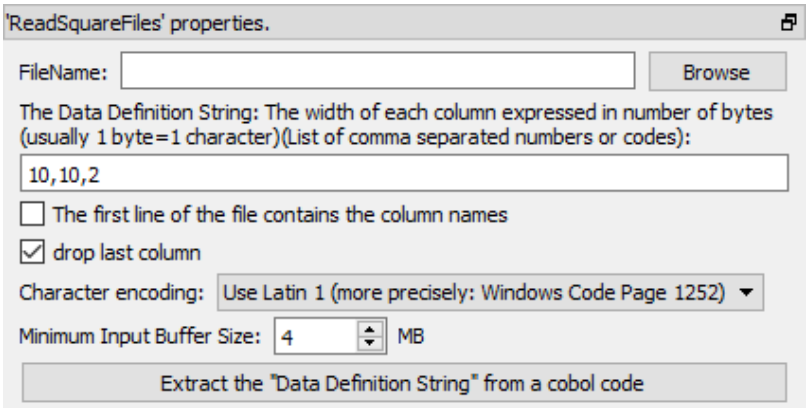


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.17. Square File Reader

Icon:

Property window:



Short description:
Reads a table from a file where each column has a constant-width (i.e. a "square" file).

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the "Square" file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

5.2.17.1. Importing simple text files

This Action allows you to read text files such as this one:

```
01234567890123456789
NAME      AGE
Frank     38
Sabrina   36
David     33
```

The above text file contains 2 columns: NAME and AGE. Each column is 10 chars width. There are 2 "invisible" characters used to mark the end of each line (i.e. the "carriage return" character (\n) and the ""line feed" character (\r)). Thus, the "*data definition string*" parameter is "10,10,2" (10 chars for each column and 2 chars for the end-of-line marker). To remove from the output table the third column (that only contains the end-of-line marker "\r\n"), check the "*drop last column*" check box.

You can drag&drop a .sqr file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadSquare Action inside the Anatella graph.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.17.2. Importing COBOL-generated MainFrame files

The "*data definition string*" parameter required to import data from MainFrame files may quickly become quite complex. This is why we advise you to use the little tool that automatically generates the "*data definition string*" parameter from the COBOL code that generated the file. Here is how this tool works:

Dangerous options

Example of Source COBOL code	Content	Source Metadata-type	Source Endianness	Anatella Code to use inside the "data definition string" parameter		
				Requested Output Metadata-Type:		
				String	Float	Key
PIC X(20) or PIC A(20)	String with 20 characters max	String (20 bytes)	-	20	-	-
PIC 9(7)	Unsigned integer with 7 digits max	String (7 bytes)	-	7	-	-
PIC S9(7)	Signed integer with 7 digits max	String (7 bytes)	-	Z7	-	-
PIC (5) COMP-3	Signed integer with 5 digits max	Binary Coded Decimal BCD (3 bytes ; 2 digits per byte)	-	B5	-	-
PIC S9(4) USAGE COMP	Signed short integer with 4 digits max	Binary (2 bytes)	little endian	-	S	H
			big endian	-	SB	HB
PIC S9(9) USAGE COMP	Signed integer with 9 digits max	Binary (4 bytes)	little endian	-	I	K
			big endian	-	IB	KB
PIC S9(10) USAGE COMP	Signed Long integer with 10 digits or more	Binary (8 bytes)	little endian	-	L	-
			big endian	-	LB	-
PIC 9(4) USAGE COMP	Unsigned short integer with 4 digits max	Binary (2 bytes)	little endian	-	-	H
			big endian	-	-	HB
PIC 9(9) USAGE COMP	Unsigned integer with 9 digits max	Binary (4 bytes)	little endian	-	U	K
			big endian	-	UB	KB
PIC 9(10) USAGE COMP	Unsigned Long integer with 10 digits or more	Binary (8 bytes)	little endian	-	L	-
			big endian	-	LB	-
PIC COMP-1	Single Precision (4 bytes) Floating Point number	Binary (4 bytes)	little endian	-	F	-
			big endian	-	FB	-
PIC COMP-2	Double Precision (8 bytes) Floating Point number	Binary (8 bytes)	little endian	-	D	-
			big endian	-	DB	-

Most of the MainFrames files have the following properties:

- Character Encoding: Use EBCDIC (cp037_IBMUSCanada)
- Endianness: Big endian

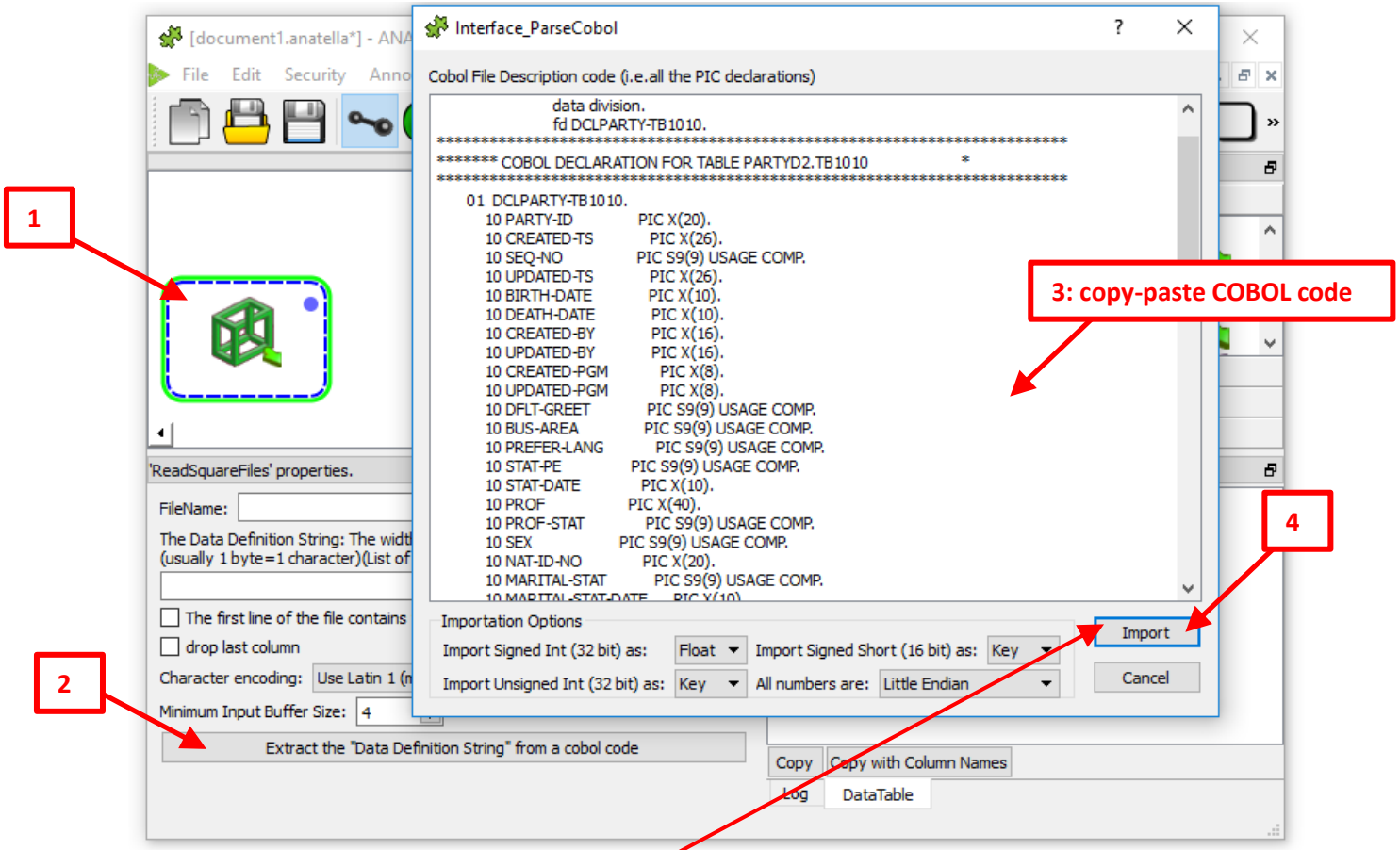
For example, let's assume that we have an input file that contains a table with 2 columns: The first column is a "*Double Precision Floating Point Number*" and the second column is a "*Signed integer with 9 digits max*" (and both columns are "little endian" because they originates from an Intel-based-machine). The "*data definition string*" parameter required to import this file is: "D,I". If we know that the second column actually contains only positive numbers, we could also have used "D,K" (but this is dangerous because any negative number inside the second column will give unexpected results when using "D,K": i.e. This is flagged as a "**dangerous option**" in the above summary table).

The procedure to automagically create the "*data definition string*" parameter from the COBOL code is the following:

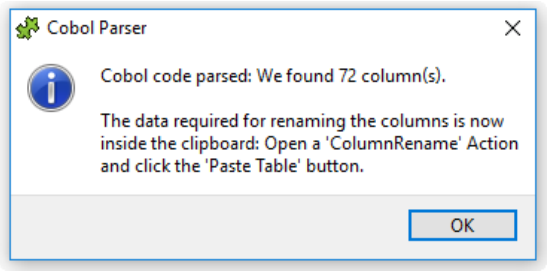
1. Click the button named "**Extract the "Data Definition String" from a cobol code**": The COBOL importation tool appears.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Copy/paste your COBOL code (i.e. the "PIC" declarations) inside the central text field inside the COBOL importation tool:



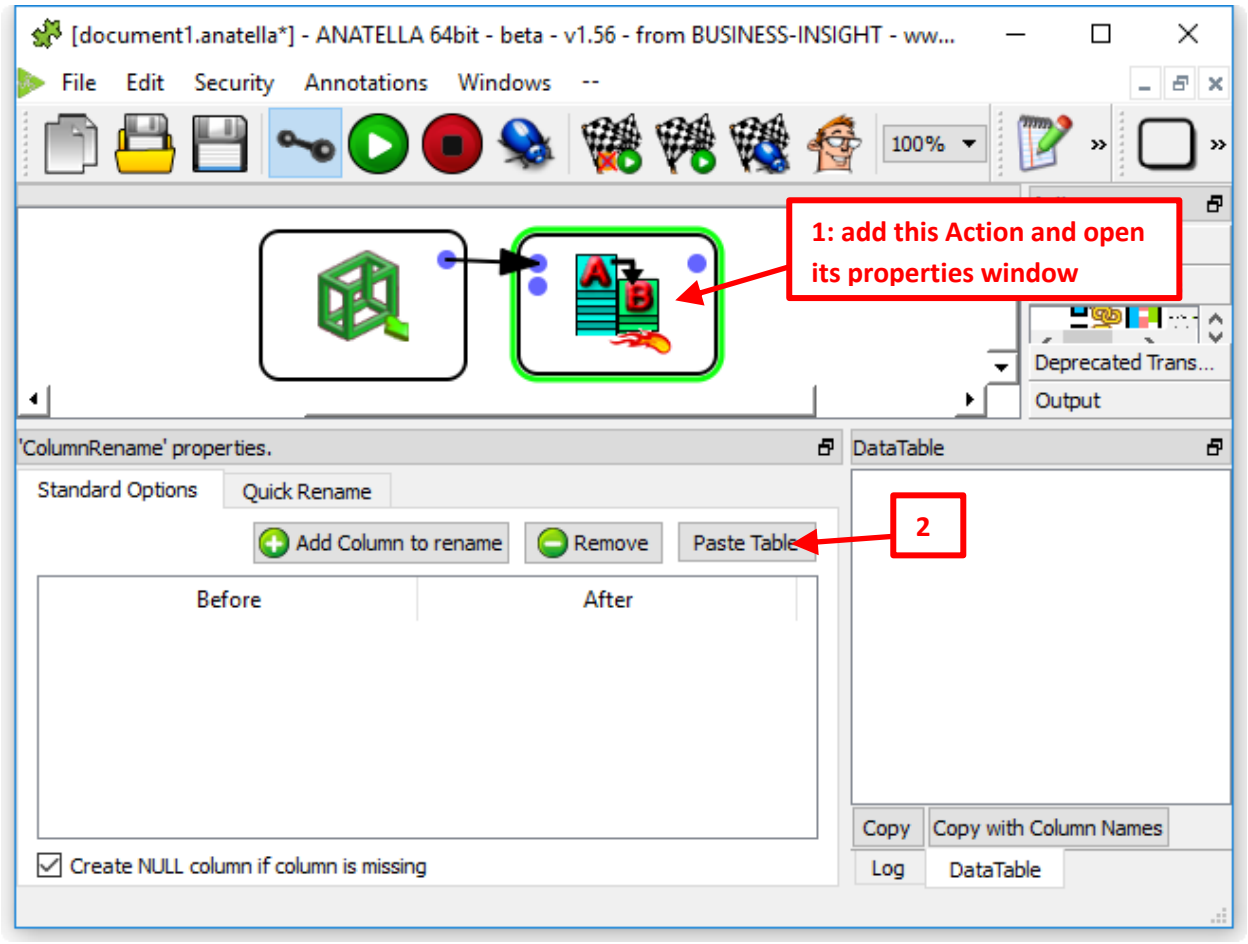
3. Click on the "import" button :
The following window appears :



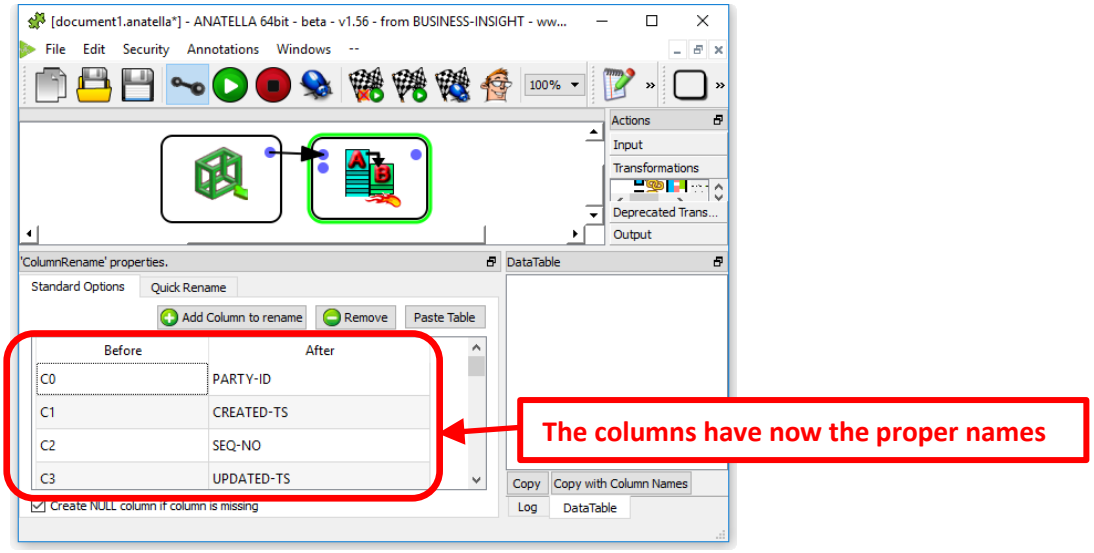
If you stop the importation procedure at this point, you won't get the proper column names (you'll get instead C1, C2, C3,… as column names)

4. Add a "ColumnRename" action, open the properties of this new action and click on the "Paste Table" button:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



5. You should now have :

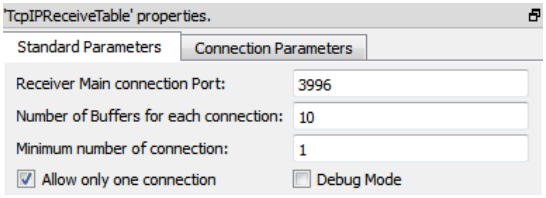


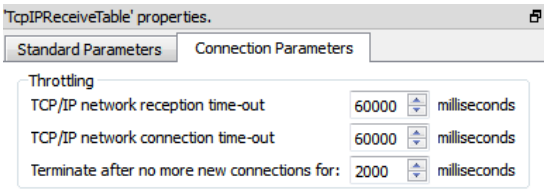
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.18. TCP-IP Receive table

Icon:

Property window:





Short description:
Receive from a TCP/IP connection a table and its meta-data.

Long Description:
This action receive a table on a specific TCP/IP port (the default port value is 3996).

The TcpIPReceiveTable Action and the TcpIPSendTable Action are used together to form the "Map" part inside a "Map-Reduce" algorithm (i.e. it allows Anatella to make distributed computations on several PC's).

When the option "*Allow only one connection*" is:

- **Disabled:** Several TcpIPSendTable Actions can connect simultaneously to the same TCP/IP port: The rows sent by the different TcpIPSendTable will be combined into one unique output table.
  The "sort flags" of the sent tables are dropped (since all the rows from the different incoming tables are mixed in a random order).
  The TcpIPReceiveTable Action detects the end of the table (i.e. the end of all transmissions from all the different TcpIPSendTable Actions) using two mechanisms:
  - The option "*Minimum Number of connection*" forces Anatella to wait for a specific number of emettors (i.e. a minimum number of TcpIPSendTable Action) before closing all connections and ending the table.
  - The option "*Terminate after no more connections for*" forces Anatella to wait for a specific duration (the counter starts after the last "live" connection is closed) before ending the table.
- **Enabled:** The "sort flags" of the (unique) sent table is preserved.

The data transferred using the TcpIPSendTable Action and the TcpIPReceiveTable Action is compressed to reduce the traffic on your computer network (it's the same row-based compression than inside .gel_anatella files). The compression ratio (25%) typically obtained inside the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

TcpIPSendTable and the TcpIPReceiveTable action is thus lower than the compression ratio typically obtained using WinRAR (8%). This means that, when you can choose between:

- Transfer a table using the the TcpIPSendTable and the TcpIPReceiveTable actions.
- Transfer a table stored inside a WinRAR file and using a MS-Windows "shared network drive".

…the later choice might be better because it reduces more the traffic on your computer network (because the volume of data transferred over the network is lower because of the better compression).

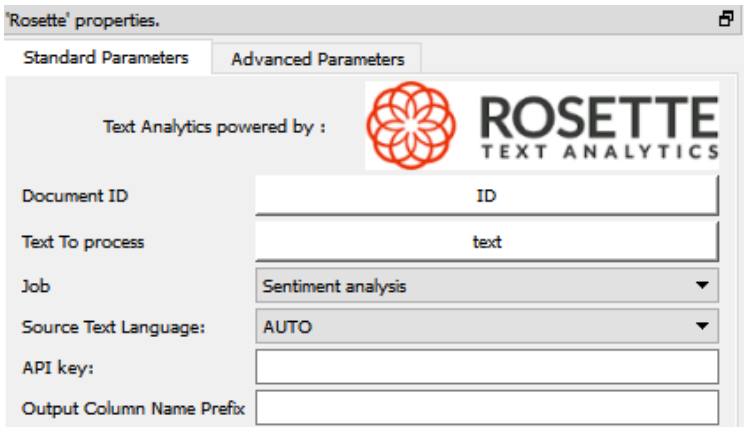
For faster processing, all data receptions are using an asynchronious I/O algorithm (See the section 5.2.6.2. about asynchronous I/O algorithms). This means that Anatella can receive (and also CRC-check and also decompress) several data-block (each data-block containing many rows) while, simultaneously, processing the required data transformations on the rows that are already available. The maximum quantity of data-block kept in RAM is defined inside the parameter "*Number of buffers for each connection*". A large value allows to cope with sudden&brief "speed drops" of your computer network (but it also implies a higher RAM consumption).

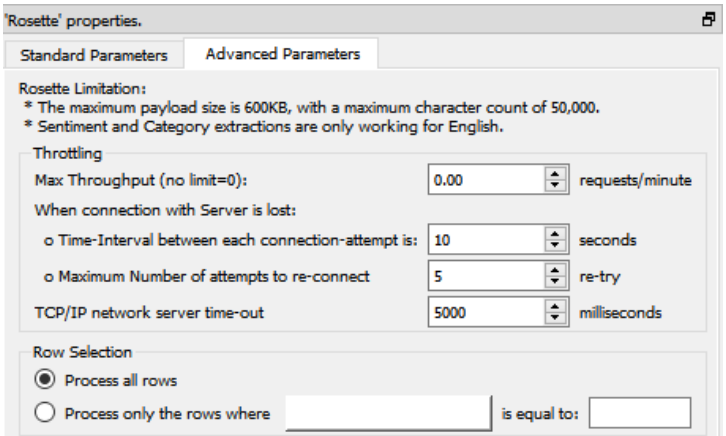
If you need to tranfert different tables using the TcpIPReceiveTable Action, than each TcpIPReceiveTable Action must have a different port number. You can change the port number using the parameter "*Receiver Main connection Port*".

5.2.19. Rosette

Icon:

Property window:





Short description:
Allow to connect to the Rosette API for advanced Text Mining capabilities.

Long Description:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The Rosette API performs different text mining tasks on text corpus:

- Sentiment analysis
- Part-Of-Speech analysis (POS)
- Entity Extraction
- Categorization

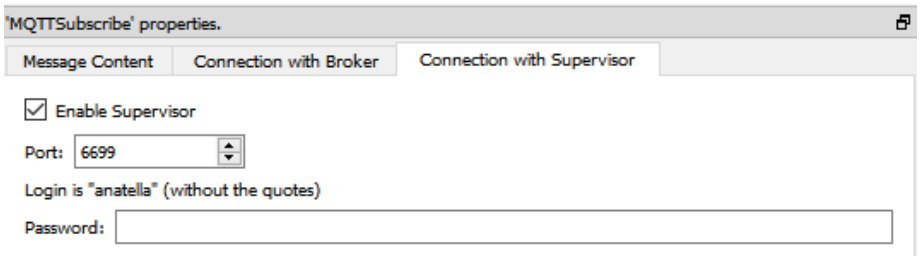
The Rosette API supports many languages:

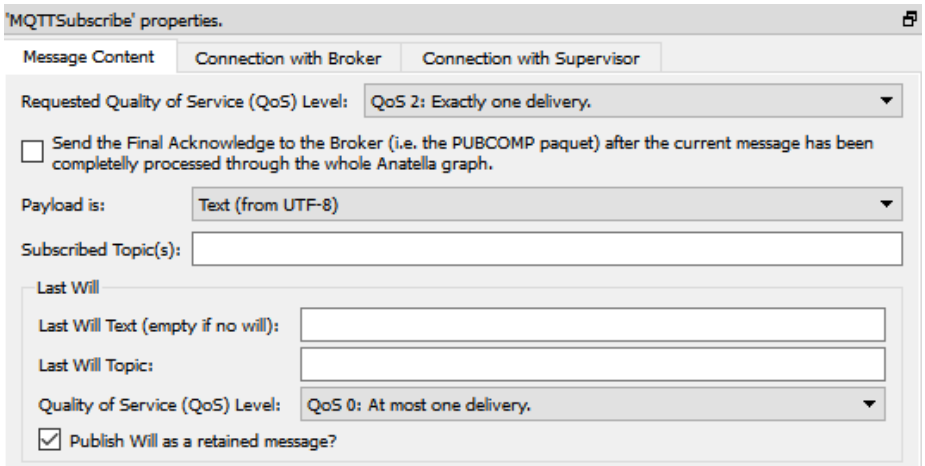
- Arabic (ara)
- Chinese: Simplified & Traditional (zho)
- Dutch (nld)
- English (eng)
- French (fra)
- German (deu)
- Hebrew (heb)
- Indonesian (ind)
- Italian (ita)
- Japanese (jpn)
- Korean (kor)
- Pashto (pus)
- Persian: Dari & Farsi (fas)
- Portuguese (por)
- Russian (rus)
- Spanish (spa)
- Urdu (urd)

5.2.20. MQTT Subscribe (IoT connector)

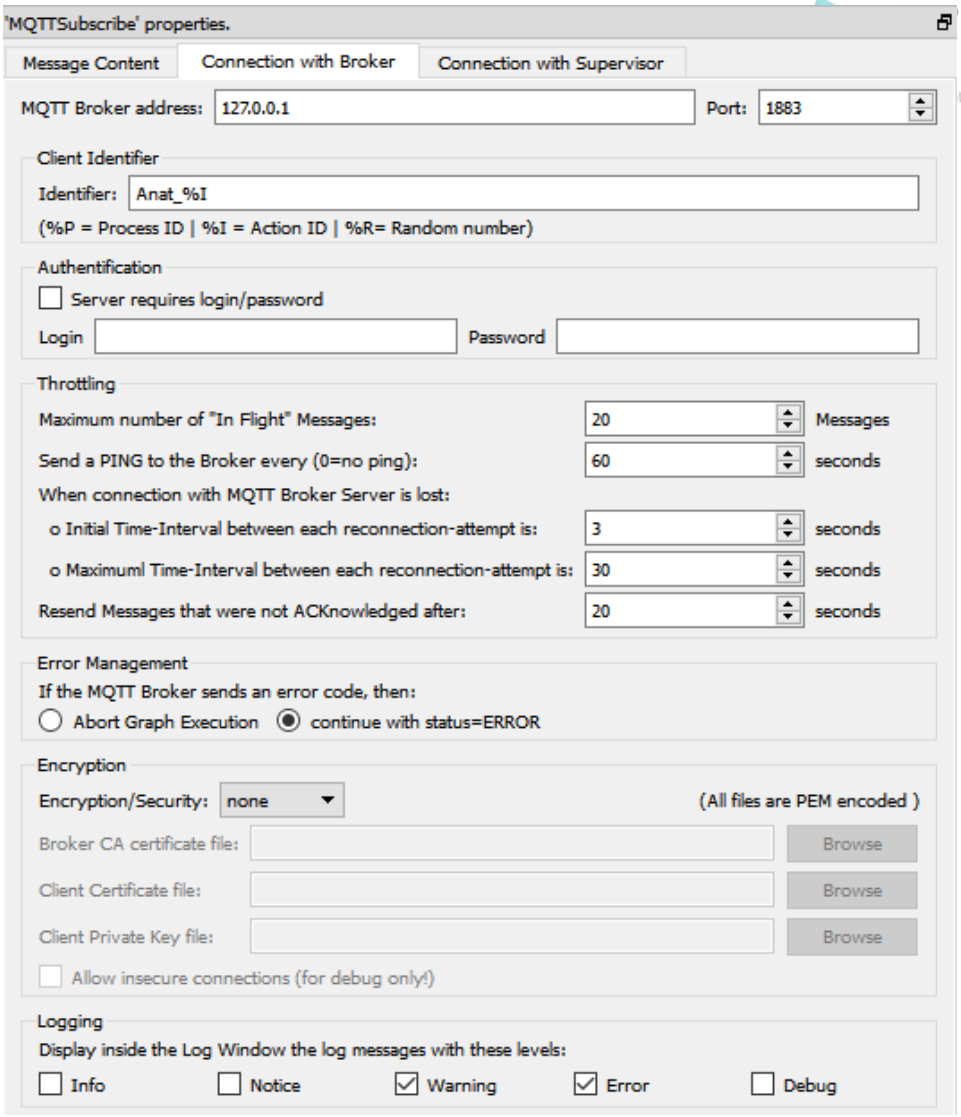
Icon:

Property window:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Short description:
Allow to connect to an “IoT” broker using the MQTT protocol to receive messages.

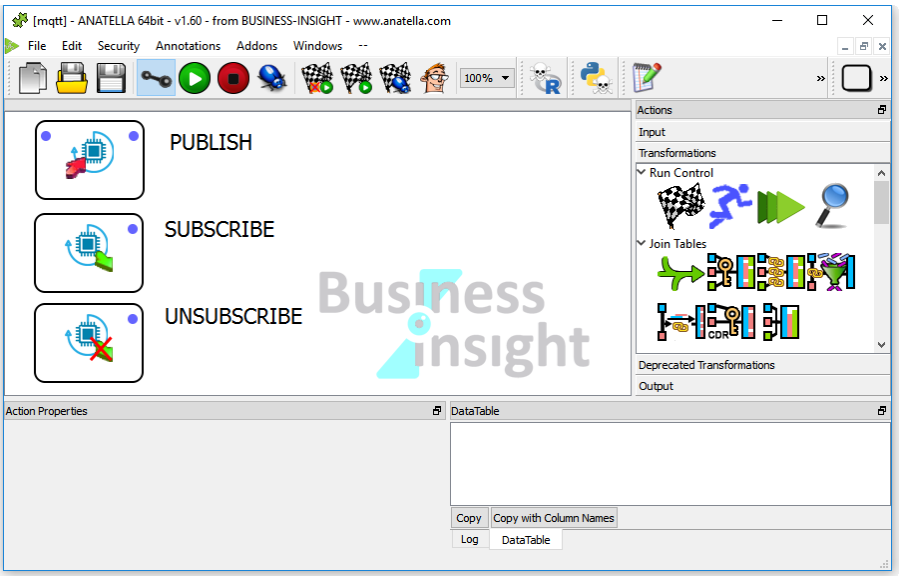
Long Description:
Anatella now natively supports the MQTT protocol. This protocol is the dominant protocol in the IoT field (Internet Of Things). The support of MQTT means that Anatella is now capable of natively receiving, processing (and possibly returning back) MQTT messages in real time. These MQTT messages can be of all types and of all kinds: For example, they can be:

- readings of temperature sensors, readings of flow sensors, etc. That’s interesting to do real-time analytics in a factory to detect leakage or near failure of equipment’s.
- the log of the “VOO” Boxes (VOO is a Belgian telecom operator) that contains the latest "clicks" done on the remote control of the Box. Indeed, each of these log messages (from the “VOO” Boxes) are quite small (a few KB) but there are many of them! (i.e. each box sends a message every 15 minutes: we have thus a few hundred thousand messages each quarter hour!!): i.e. it's the ideal context for implementing a "queuing system" (i.e. a system that collects and processes messages in real time) based on MQTT.
- Location sensors based on a GPS inside an android phone (i.e. a small “app” in a smart phone). For example, for Walt Disney, we could in real-time send a message when a visitor "pass by" an attraction and, at the same time:
    - …the attraction has a small waiting period (no waiting line!)
    - …the attraction is very likely to please the customer.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This implies a large quantity of predictive model (one for each attraction) and this implies the support of a real-time message processing system. This is now easily possible with the latest anatella version. Cool!

In Anatella, the MQTT protocol is supported by these 3 Actions:



More details: The 3 Anatella boxes above are connecting to remote servers (in technical terms: they connect to "brokers") who forward to Anatella in real-time different messages that are immediately analyzed and processed by Anatella (Anatella then carries out different analysises in "streaming" mode, in opposition to a "batch" mode).

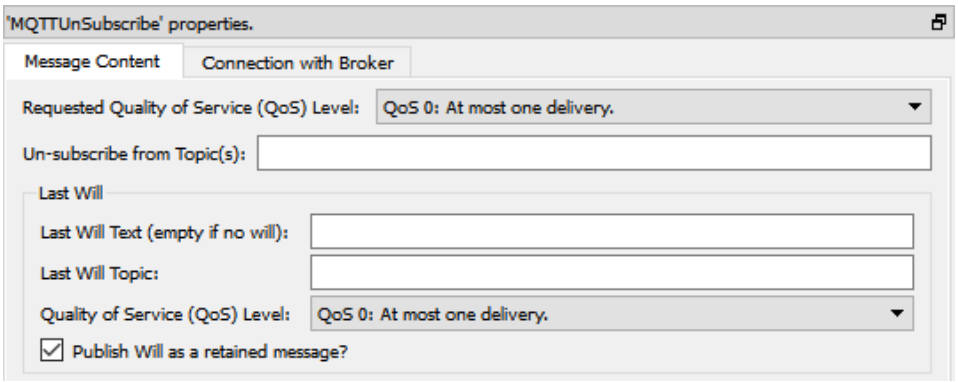
There exists now a plethora of "brokers" (most of them are free) that are compatible with Anatella (since they support the MQTT protocol): For example, these brokers are: Mosquitto, RabbitMQ, Kafka, EMQTT, etc.

For more comfort, the small broker "mosquitto" is now included directly inside the timi distribution (This transforms TIMi as a 100% complete IoT solution directly "out-of-the-box").

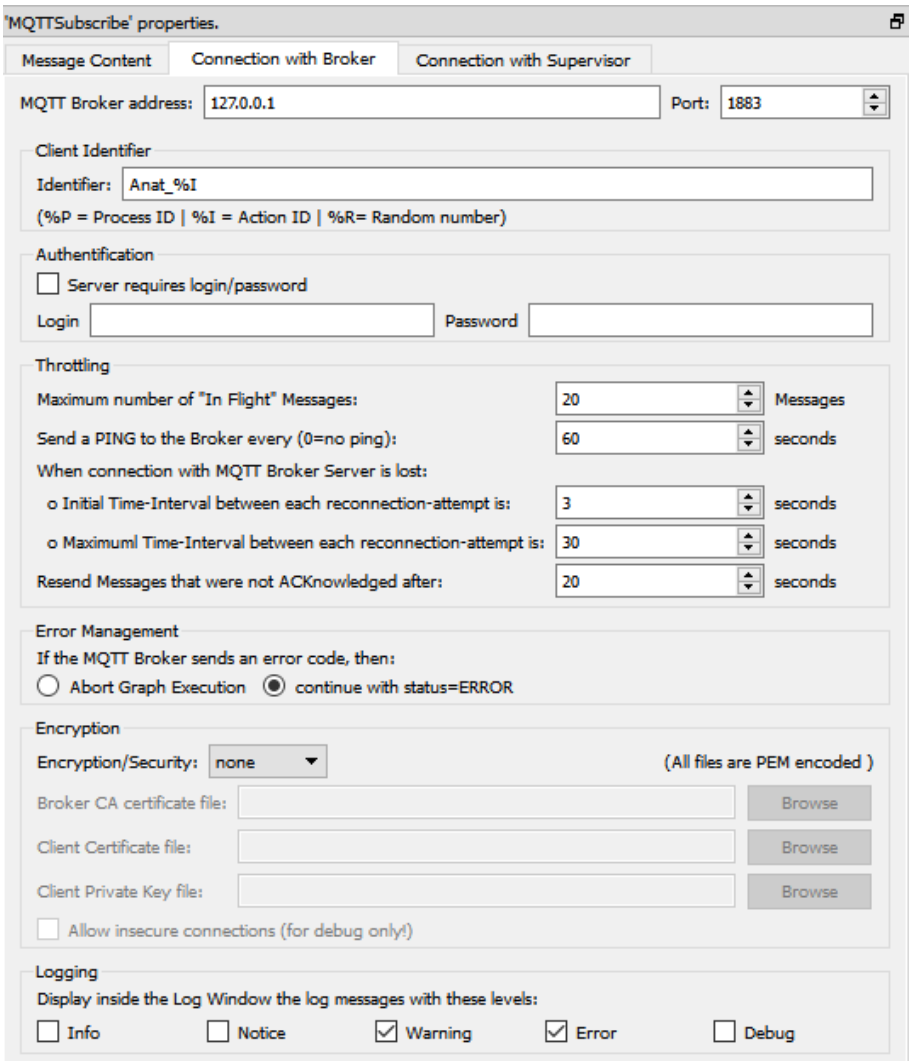
5.2.21. MQTT Unsubscribe (IoT connector)

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

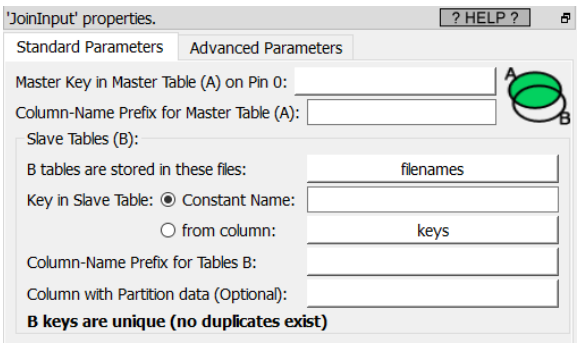


Short description:
Allow to connect to an "IoT" broker using the MQTT protocol to actively unsubscribe from a channel.

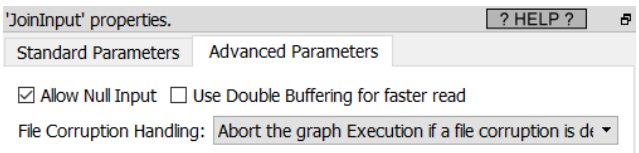
Long Description:
Some brokers (e.g. Kafka) have a "memory" of all the messages that passes through them. These brokers are waiting for a subscriber to (re-)start (i.e. they are waiting for the Anatella process to re-start) and, once the Anatella process is started, they forward all the saved messages to it: They forward all the messages that are inside the subscribed channel(s) and that were not yet processed by the subscriber. To prevent the broker to save into its memory all the messages, you need to actively tell to the broker that you don't want to receive any message anymore in the future: This is the exact objective of the "MQTT Unsubscribe" Action.

5.2.22. Join Input

Icon:



Property window:



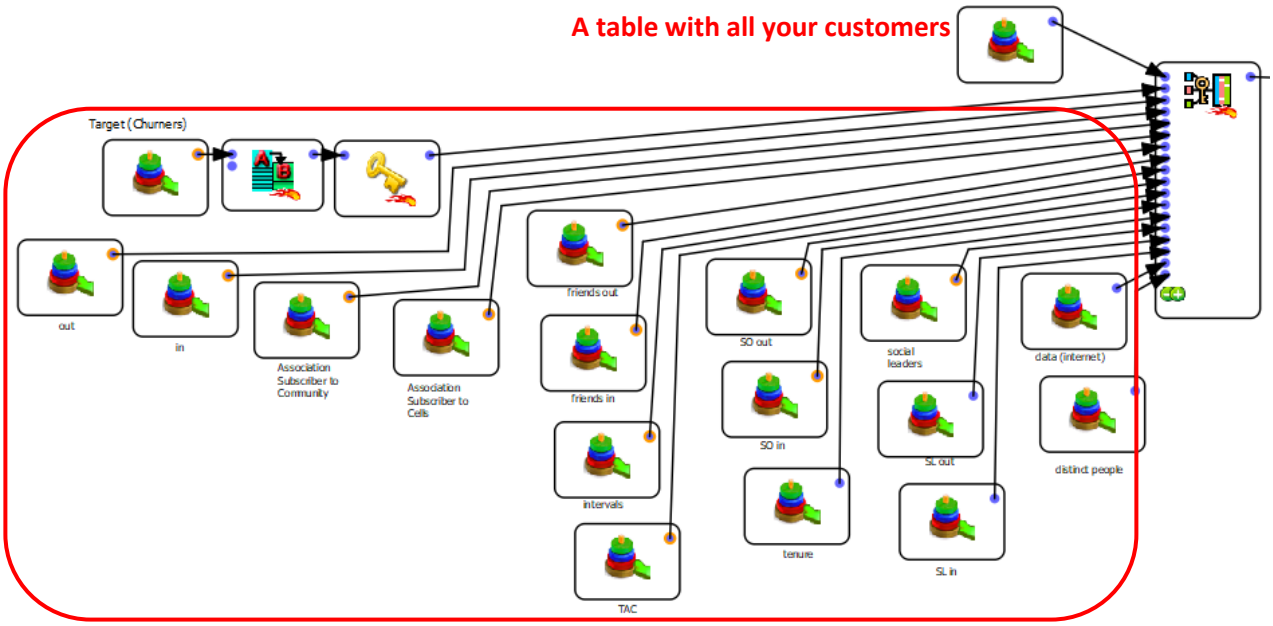
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Left-Join several tables into one gigantic table.

Long description:

Pre-requisite
All the input tables must be sorted on the "Key" columns using the SAME sorting algorithm (all "numeric sort" or all "alpha-numeric sort").

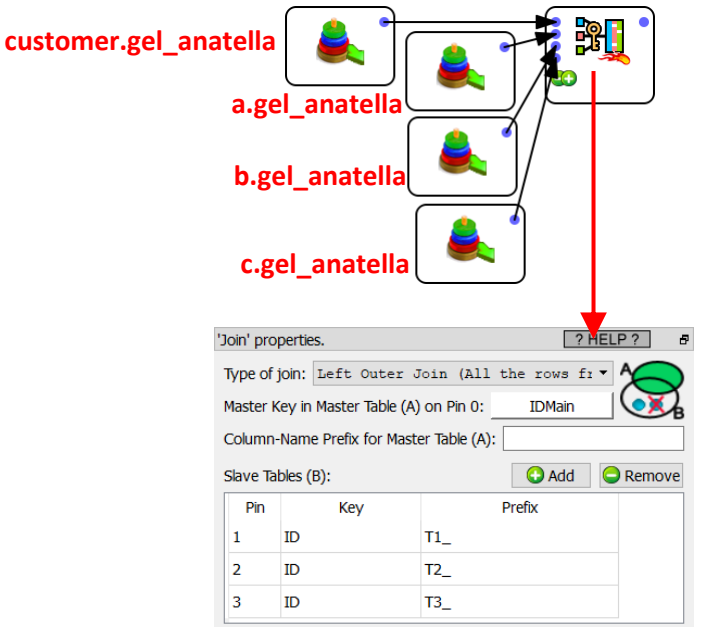
The first step before creating any predictive model is to build a learning dataset that contains as many "good" features as possible (this is named "feature engineering"). Typically, each group of features is computed using one different Anatella graph. To gain time, you'll usually run all the required "feature engineering" graphs in parallel (using the ParallelRun Action from section 5.3.3. or using the loopAnatellaGraphAdv Action from section 5.20.6 or using the "Loop Jenkins" action from section 5.21.1). Each of these "feature engineering" graph delivers, as output, one small .gel_anatella file that contains some more features. The final step of the construction of your learning dataset is to "assemble" all these small .gel_anatella files into one large unique .gel_anatella file (that is your learning dataset). This "final assembly" might look like this:

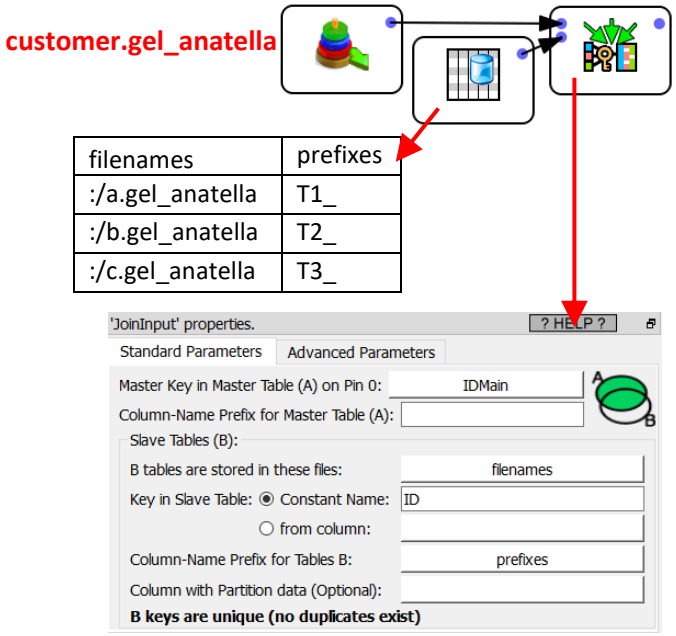


The above (real-life) example illustrates that, for a large number of different features, the construction of this "final assembly" might rapidly become an un-manageable graph with too many arrows and boxes (that looks like a plate of spaghetti! 😉). At that point, you quickly understand why the JoinInput Action is really interesting!

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To illustrate how to the JoinInput Action works, let's look at a first example: The objective of this example is to add to a "customer.gel_anatella" table some additional features that are (initially) saved inside the 3 "slave" tables that are named "a.gel_anatella", "b.gel_anatella" and "c.gel_anatella". To make this assembly, we can use any of these two Anatella graphs:





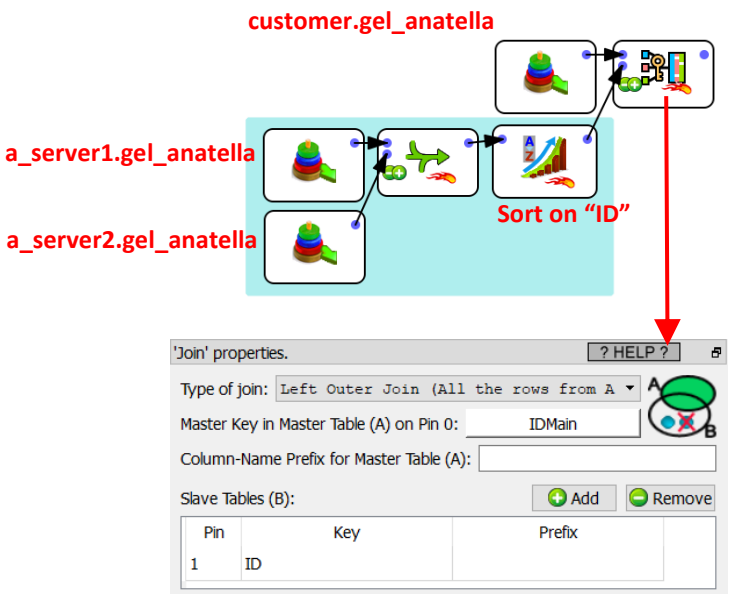
At first sight, these 2 graphs do look pretty similar in complexity. The main difference occurs when you increase the number of "slave" tables (that contains all the computed features) inside the Join (the above example only contains 3 "slave" tables: "a.gel_anatella", "b.gel_anatella" and "c.gel_anatella"): With the JoinInput Action, you can easily have thousands of Slave tables, while the (other) approach based on the simple Join Action becomes un-manageable pretty quickly: Already with as little as 100 "Slave" tables, the Anatella graph is becoming a total mess to maintain&support: Just think about the number of boxes! (it will work but it will definitively be less manageable than the JoinInput Action).

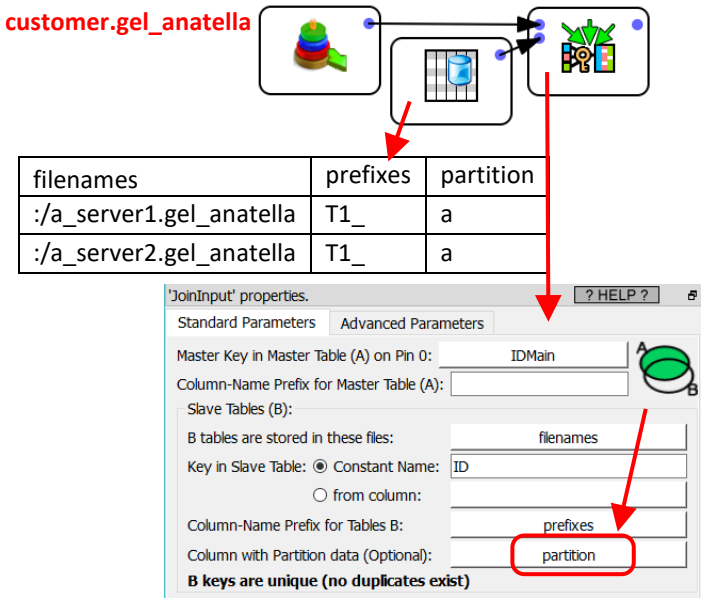
The maximum number of "slave" tables inside the JoinInput Action is around 20000. This means that Anatella can effectively compute a Left join between 20000 tables! (as a comparison, the best databases are limited to 32 tables in a join)

Let's now assume that you need to create a learning/scoring dataset that has a very large number of rows. Each row represents one customer. To divide the computing time by 2, you decided to store the raw data collected on your customers on two different servers (the data from the customers with an odd primary key is stored on the first server and the data from the customers with an even primary key is stored on the second server). Then, when you are computing your features, you use both servers simultaneously (e.g. using the "Loop Jenkins" action from section 5.21.1). At the end of the feature computation, the first server contains all the features for half your customers and the second

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

server contains all the features for the other half of your custromer. Then, you need to do the “final assembly”: To make this assembly, we can use any of these two Anatella graphs:

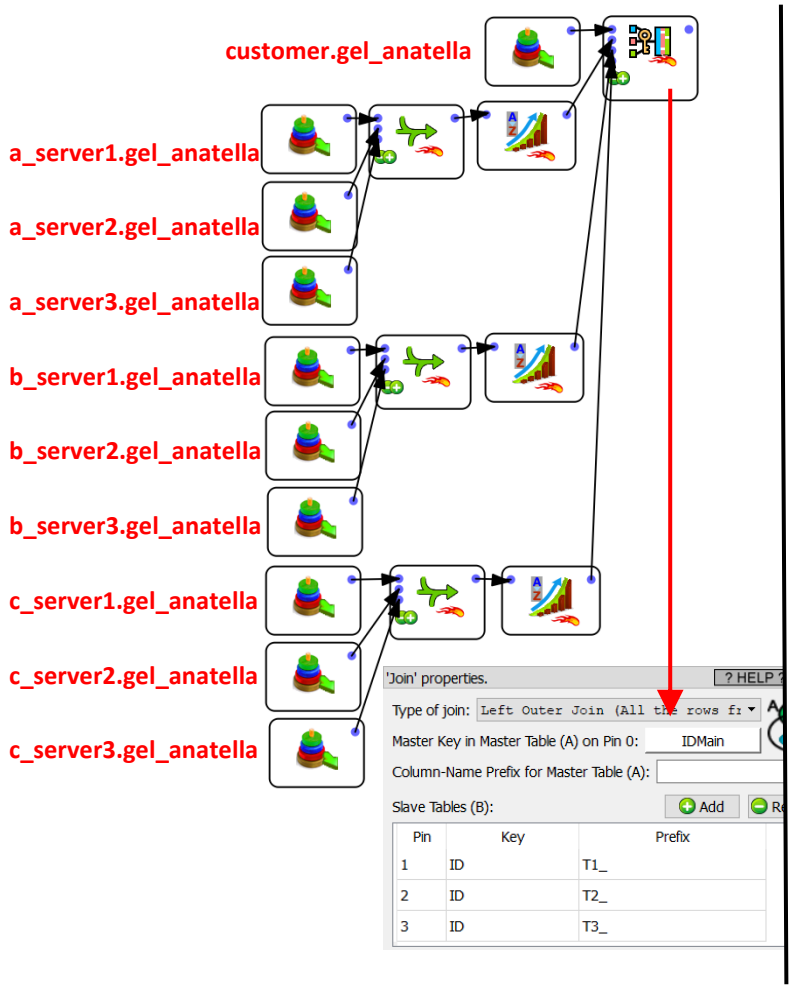


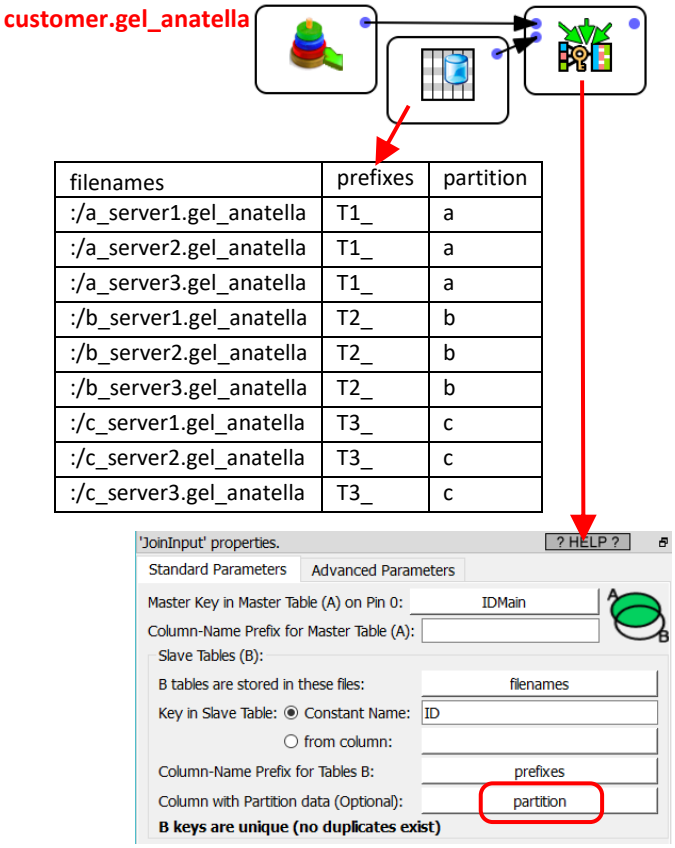


What happens if:

- …instead of using 2 servers, we use 3 servers?
- …instead of computing only the “a” features, we also compute the “b” and “c” feature sets?

To make the “final assembly”, we can use any of these two Anatella graphs:





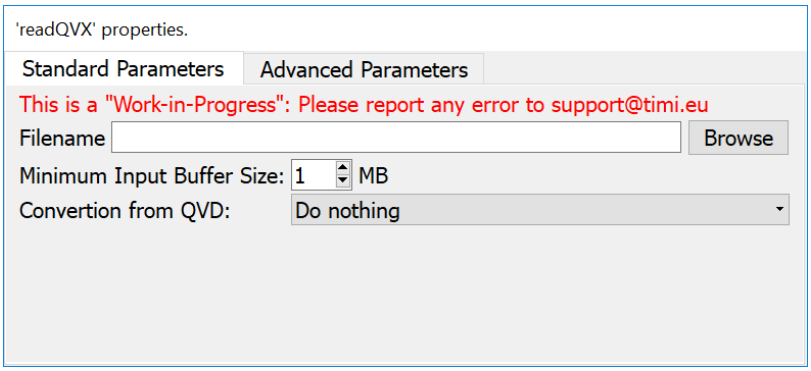
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Inside the above example, all the partitions used for the JoinInput Action are composed of 3 files.
In practice, each partition can have a different number of files: No worries! 😉

5.2.23. Read QVX

Icon:

Property window:



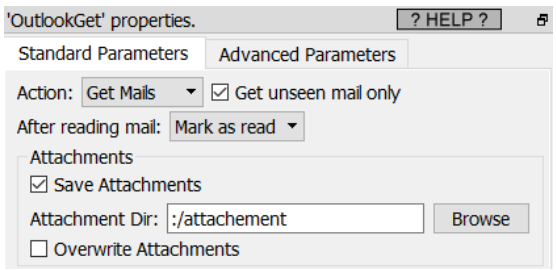
Short description:
Open a QVX File

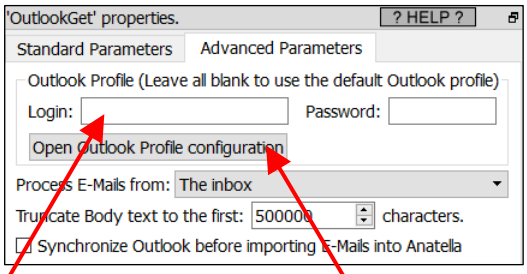
Long Description:
Select a QVX file to open in Anatella.

5.2.24. Outlook Get

Icon:

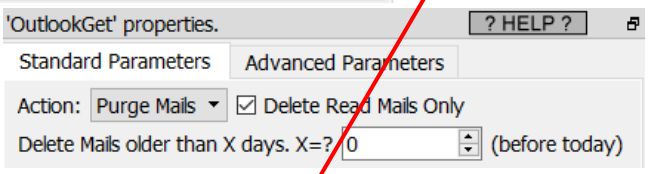
Property window:





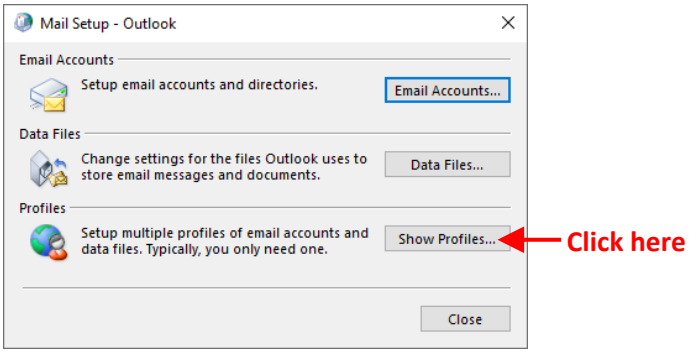
Short description:
Get Emails from Outlook.

Long Description:



5.2.24.1. About Outlook Profiles

All your emails and settings inside Outlook are stored inside an "Outlook Profile". Before sending or receiving any email, you must select the right "Outlook Profile". Anatella will attempt to use the Default "Outlook profile" to send/receive emails but, in some specific cases, this might not be the right "Outlook profile" to use. To select the "Outlook profile" that Anatella is using, you can use the parameter named Outlook "Login":
To get the different admissible values for the Outlook "Login", click the "Open Outlook Profile Configuration" button here:
You should then see the following window:



Click here

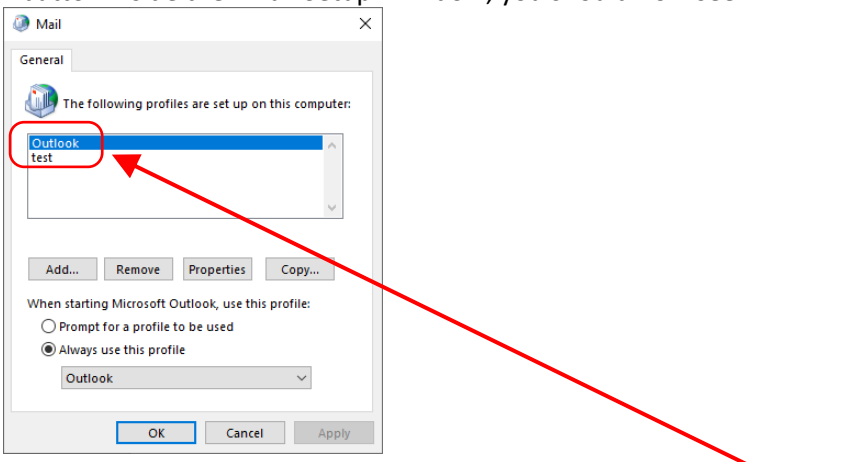
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If you cannot see the above window ("Mail Setup"), this means that, typically, you are running the 64-bit version of Anatella and you have the 32-bit version of Outlook that is installed.

You need the **32**-bit version of Anatella to interact with the **32**-bit version of Outlook.
You need the **64**-bit version of Anatella to interact with the **64**-bit version of Outlook.

You can install on the same PC, at the same time, both the "Anatella 32-bit" and the "Anatella 64-bit". This means that you can still use the fastest "Anatella 64-bit" to do all your data transformations and only use the Anatella 32-bit to communicate with Outlook. You'll find mode informations on how to install simultaneously the "Anatella 32-bit" and the "Anatella 64-bit" on the same PC inside the section 10.11.

Click the "Show Profiles" button inside the "Mail Setup" window, you should now see:



The different admissible values for the parameter Outlook "Login" inside Anatella, here are:
In the example above, it is either "Outlook" or "test".

5.2.24.2. Outlook Automation

It's quite difficult to automate/schedule tasks that are using the outlookGet Action. This is why you should rather use instead the getMailOffice action, that offer the same set of functionalities but can be easily automated with Jenkins.

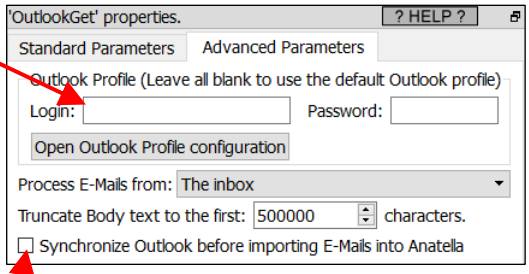
You cannot use Jenkins to run a graph containing the outlookGet Action: You must use instead the "Windows Task Scheduler" (i.e. run "taskschd.msc"). Furthermore, the "security options" of the task must be "*Run only when the user is logged on*" (and you must uncheck the "*hidden*" checkbox option).

5.2.24.3. Outlook Synchronization

Before importing any E-mails from Outlook, Anatella can, optionnaly, ask to Outlook to run a "synchronization" on all the accounts configured inside Outlook. This means that Outlook will connect to all the known IMAP servers, POP3 servers, MS-Exchange servers, etc. and download locally all the new emails that were recently received. Anatella is only able to import the emails that are available locally, so it's important to run a "synchronization" procedure before any importation operation inside Anatella.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To get a correct synchronization for the right Outlook "Profile", you need to give the right Outlook Login here: see the previous section 5.2.24.1. to know how to find the right value for this parameter.

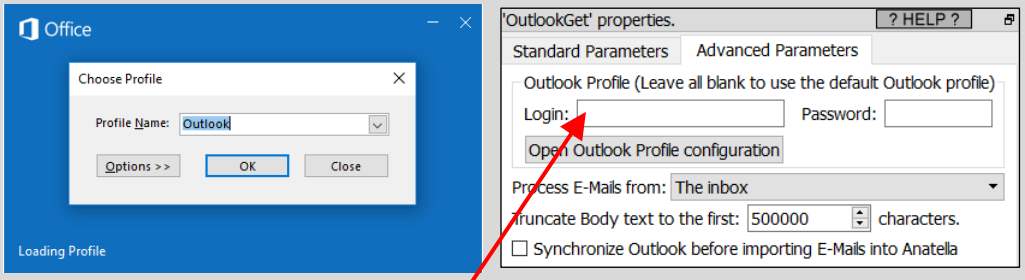


If, for some reason, you already know that your outlook accounts are already properly synchronized, you can disable the "synchronization" initiated by Anatella: To do so, un-check the checkbox here (This is not recommanded but it might save you a little bit of running-time because the "synchronization" procedure can be lengthy):

Each time Anatella runs the Outlook "Synchronization" procedure, you'll see inside the Anatella Log window:

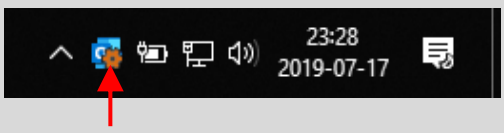
```
Synchronizing Outlook…
Outlook synchronization finished successfully!
```

The Outlook "Synchronization" procedure is performed by a small external executable named "OutlookSync.exe". Sometime, when the "Synchronization" procedure starts (i.e. when the "OutlookSync.exe" executable starts), an annoying (and blocking) Dialog Box appears:



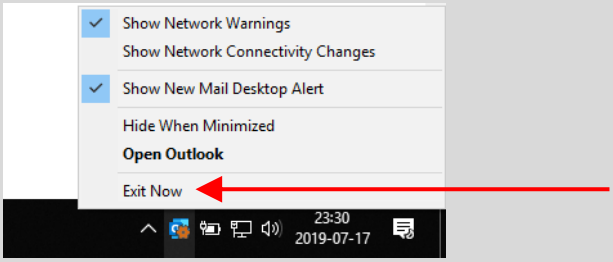
To get rid of this annoying dialog box, select the right Outlook "Login" parameter inside Anatella here: (see the previous section 5.2.24.1. to know how to find the right value for this parameter).

You can see inside the notification area of the MS-Windows taskbar a small icon that signals that the "OutlookSync.exe" executable (that performs the Outlook "Synchronization" procedure) is running:



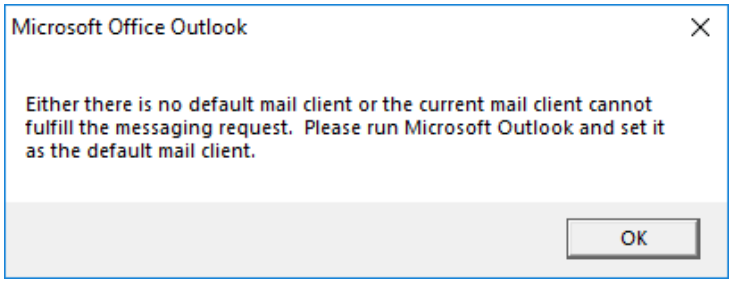
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This same icon might sometime stay visible after the "Synchronization" procedure is finished. If it bothers you, you can safely remove the icon by right-clicking it and then selecting the "*Exit Now*" option:



5.2.24.4. Using the correct 32-bit or 64-bit Anatella version

Most of the time, you will find that the 32-bit version of MS-Outlook is installed on your machine. To communicate with the 32-bit version of MS-Outlook, you *must* use the 32-bit version of Anatella. More precisely, in such a situation, the 64-bit version of Anatella won't work: i.e. You'll get this error message:



If you see the above error message, it means that you need to use the 32-bit version of Anatella to communicate with Outlook (because you have the most common "32-bit version" of Outlook).

In the same order of idea, you must use the 64-bit version of Anatella to communicate with the 64-bit version of MS-Outlook.

You can install on the same PC, at the same time, both the "Anatella 32-bit" and the "Anatella 64-bit". This means that you can still use the fastest "Anatella 64-bit" to do all your data transformations and only use the Anatella 32-bit to communicate with Outlook. You'll find mode informations on how to install simultaneously the "Anatella 32-bit" and the "Anatella 64-bit" on the same PC inside the section 10.11.

5.2.24.5. About the "MailTextBody" output column

As output of the OutlookGet Action, you'll receive a table with the following 13 columns: *MailReceiveTime, MailSubject, MailTextBody, MailHtmlBody, FromName, FromAddress, NbRecip, To, CC, BCC, Reply-To, Attachments, CurrentOutlookProfile, Mailbox*. Most of the time, the column "*MailTextBody*" will be empty: This just means that:

- …the content (body) of the email is in HTML format (and not in the simpler unformatted TEXT format)
- …the content of the email is inside the "*MailHtmlBody*" column.

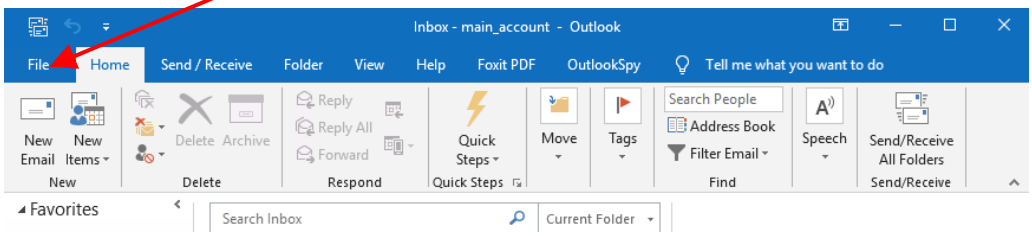
In a similar way, when the content of the email is in unformatted TEXT format, the "*MailHtmlBody*" column will, most of the time, be empty (and the column "*MailTextBody*" is filled-in).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

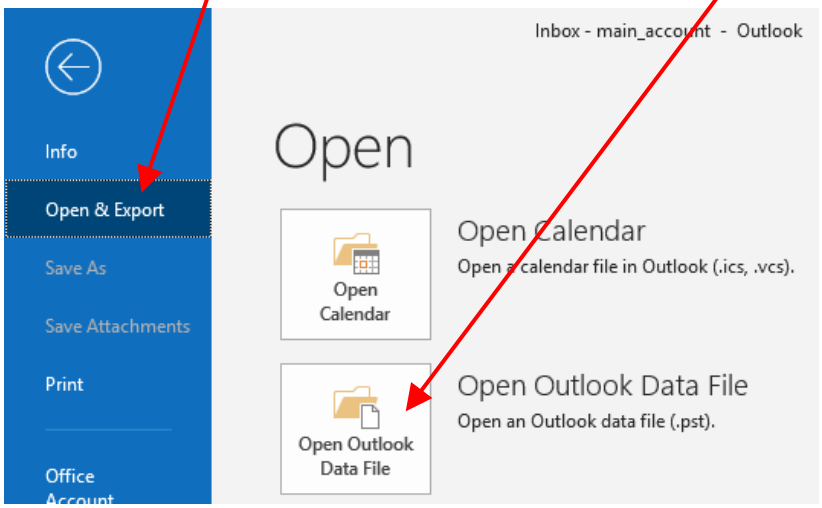
5.2.24.6. Importing large PST files in Anatella

To do so:

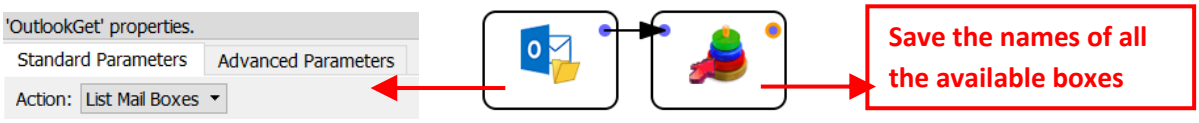
1. Open outlook and click on “File”:



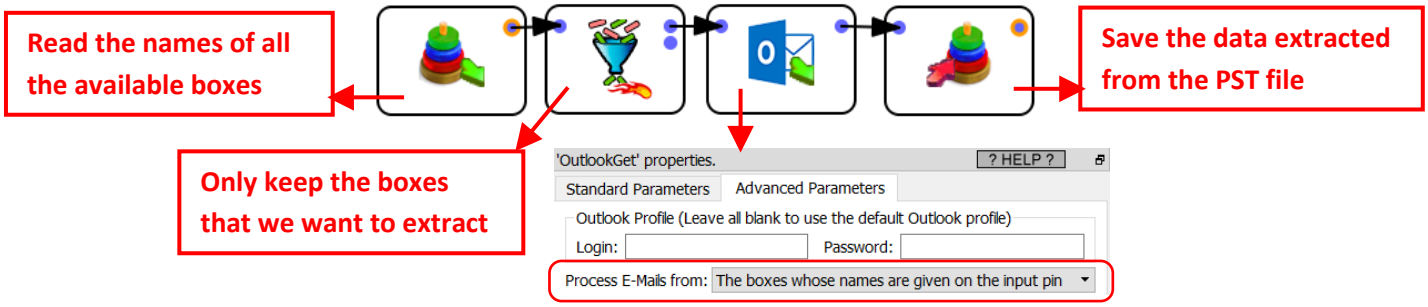
2. Click on “Open & Export”; After, click on “Open Outlook Data File”:



3. Browse your disk and select your pst file.
4. Open Anatella and use the “List Mail Boxes” mode inside the OutlookGet action to retrieve a list of all the available boxes:



5. Extract the content of the PST file in Anatella:

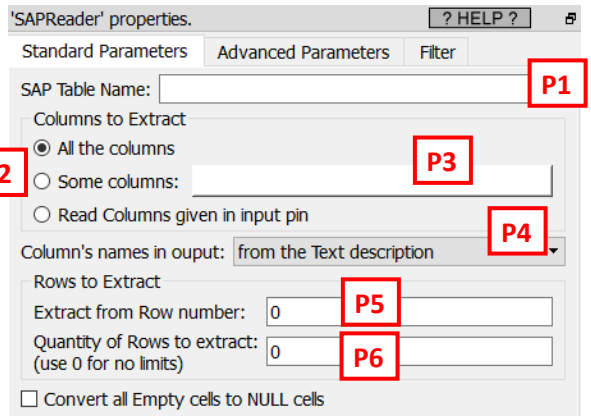


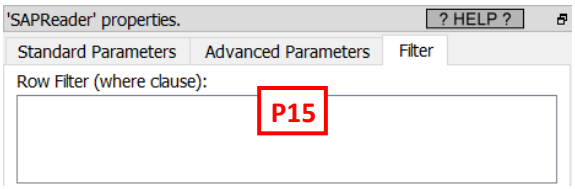
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

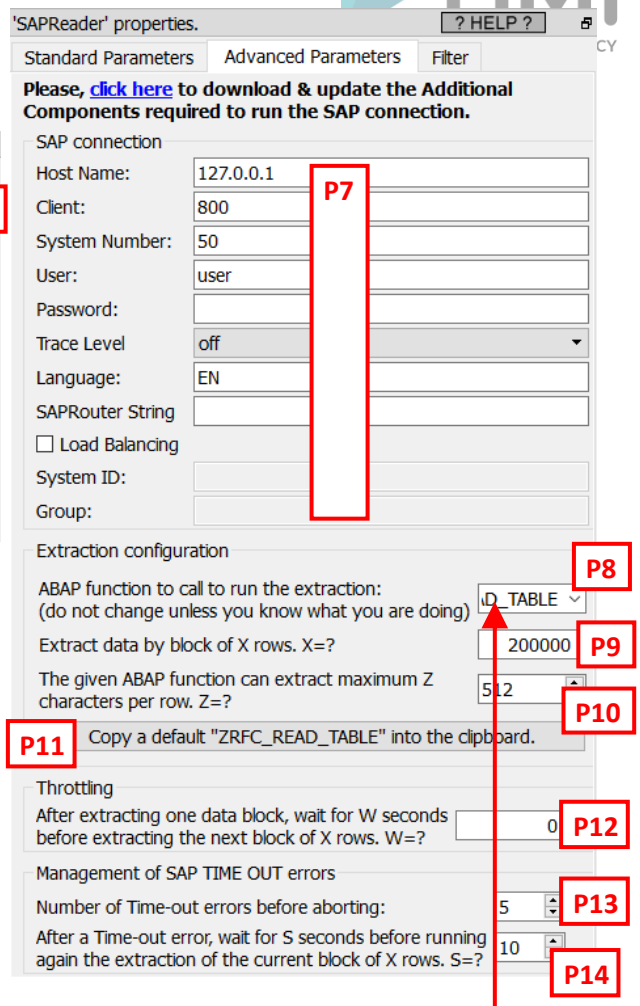
5.2.25. Read SAP tables

Icon:

Property window:







Short description:
Extract tables from SAP

Long Description:
The SAP table extraction is performed using the newest NetWeaver RFC libraries. By default, to run the extraction, Anatella calls the ABAB function named “BBP_RFC_READ_TABLE”, but you can change that using the parameter **P8** here:

Instead of running “BBP_RFC_READ_TABLE”, you can run any alternative ABAP function with the same signature. All the ABAP functions with the following signature are compatible with Anatella:

```
FUNCTION bbp_rfc_read_table.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(QUERY_TABLE) LIKE  DD02L-TABNAME
*"     VALUE(DELIMITER) LIKE  SONV-FLAG DEFAULT SPACE
*"     VALUE(NO_DATA) LIKE  SONV-FLAG DEFAULT SPACE
*"     VALUE(ROWSKIPS) LIKE  SOID-ACCNT DEFAULT 0
*"     VALUE(ROWCOUNT) LIKE  SOID-ACCNT DEFAULT 0
*"  TABLES
*"      OPTIONS STRUCTURE  RFC_DB_OPT
*"      FIELDS STRUCTURE  RFC_DB_FLD
*"      DATA STRUCTURE  TAB512
*"  EXCEPTIONS
*"      TABLE_NOT_AVAILABLE
*"      TABLE_WITHOUT_DATA
*"      FIELD_NOT_VALID
*"      NOT_AUTHORIZED
*"      DATA_BUFFER_EXCEEDED
*"----------------------------------
```

The possible values for the type of the OPTIONS table are:
(This is a DDIC structure with only one TEXT field of type CHAR.
A larger value allows you to write more complex&longer filters).

RFC_DB_OPT	CHAR 70	(The default type of RFC_READ_TABLE)
VALUEREC	CHAR 255	

The possible values for the type of the DATA table are:
(A higher length allows to reduce the number of simultaneous connections required to extract all the selected columns)

TAB512	CHAR 512	(This is the default type of RFC_READ_TABLE and BBP_RFC_READ_TABLE)
SDBLIN1024	CHAR 1024	
TBL2048	CHAR 2048	
SDOKURL	CHAR 4096	(Best: It allows the extraction of all the columns in BSEG using only 1 connection with SAP)
L004TAB	CHAR 8192	

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

These are all the standard ABAP functions that are compatible with Anatella:

```
BBP_RFC_READ_TABLE
RFC_READ_TABLE
/SAPDS/RFC_READ_TABLE  (which is still different from "RFC_READ_TABLE")
Z_AW_RFC_READ_TABLE
ZRFC_READ_TABLE
```

Many SAP system only contains the ABAP function "RFC_READ_TABLE" and no other function.

If you get an error inside the Anatella-log-window that says:
SAP ERROR: cannot describe function …
or
SAP ERROR: authorization refused to access function…
This means that you need to use another ABAP function to run the extraction.

If you don't have any ABAP function that is compatible with Anatella inside your SAP system, don't worry: You can still get the code of a good ABAP function (that is named "`ZRFC_READ_TABLE`") that is compatible with Anatella: Just click the button named "`Copy a default "ZRFC_READ_TABLE" into the clipboard`" (This is the parameter **P11**). However, in such (very **un**common situation), you'll need to contact an SAP administrator (and thus you'll need to have administrative rights to SAP) to add the required ABAP function inside your SAP system. Luckily, 99% of the time, the "`RFC_READ_TABLE`" ABAP function is available (and then you don't need any administrative rights to run all the extractions: i.e. a simple & common SAP login is just enough).

Which ABAP function should I use? What's the order of preference?
The order is (from best to worst):

/SAPDS/RFC_READ_TABLE
BBP_RFC_READ_TABLE
Z_AW_RFC_READ_TABLE
ZRFC_READ_TABLE
RFC_READ_TABLE

The "RFC_READ_TABLE" ABAP function has the advantage to be available in (nearly) all SAP systems.

The disadvantage of the "RFC_READ_TABLE" ABAP function is that, sometime, when working with some columns of type FLOAT, this function might cause an ABAP exception named "ASSIGN_BASE_WRONG_ALIGNMENT". When this happens, the extraction fails (In this situation, just remove the FLOAT column that caused the error from the list of selected columns and re-run the extraction).

The extraction methogology that is used by Anatella guarantees a total safety: No data is ever written inside SAP. The extraction algorithm used in Anatella is working in pure "read-only" mode: It is thus 100% safe. There are no dangers of loosing any data from your SAP system.

All these ABAP functions proceed in the same way to do the extraction:
More precisely, they all execute the following steps:

1. Run the following SQL command:
   SELECT * FROM <table_given_in_parameter_**P1**> WHERE <filter_given_in_parameter_**P15**>

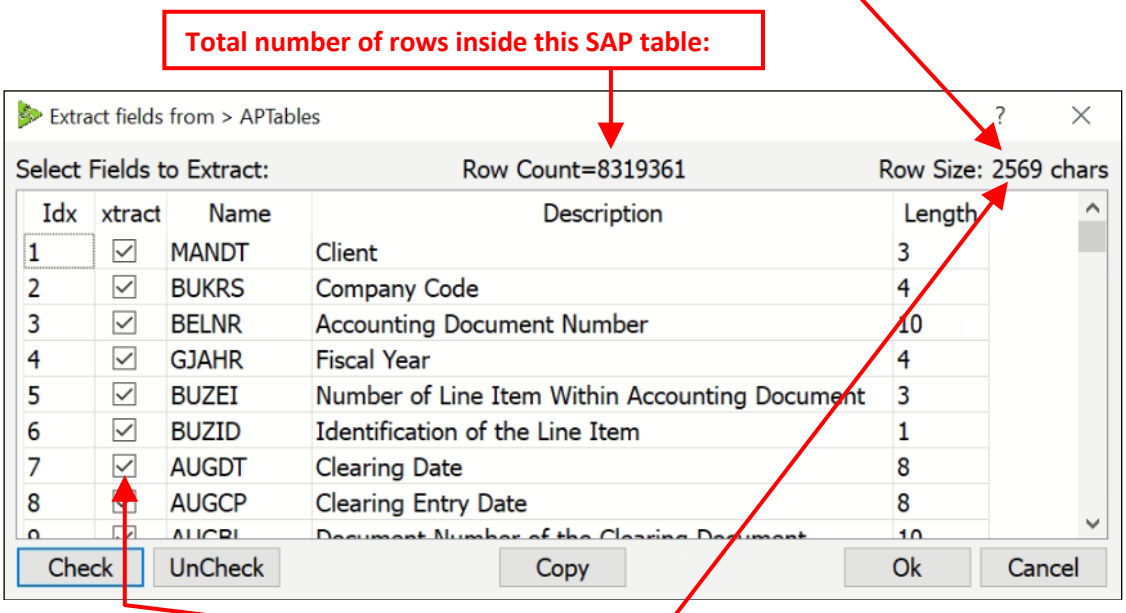
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Run a loop to skip the first X rows of the output table of the "SQL SELECT" command that was performed during the previous step 1. The quantity X of "skipped" row is given in the parameter **P5.**
3. Create "in-memory" (inside the memory of the remote SAP server) a large table "T" that contains at most R rows and one single column.
   The maximum quantity ("R") of rows inside the output table "T" is defined using both the parameters **P6** and **P9**.
   The single column of the output table "T" is a very wide column: it contains the concatenation inside one unique string "S" of all the columns that were selected using the parameters **P2** and **P3**.
   Depending on the ABAP function used, there are different limits to the length of this large string "S". For example, when you use the standard "RFC_READ_TABLE" or "BBP_RFC_READ_TABLE" ABAP functions, the maximum length of the string "S" is 512 characters.
4. Send back to Anatella the output table "T" using the computer network. At that point, the whole output table "T" is fully stored inside both the RAM memory of the SAP server and inside the RAM memory of the Anatella server.

There are many different ways for the above ABAP function to fail. Luckily, the Anatella engine that is used to extract data from SAP has many built-in "work arounds" to still be able to run a successful extraction. You're in for a treat because the SAP-data-extraction-engine included inside Anatella is one of the most advanced engine currently available!

The most common cause of failure is due to the limitation on the length of the string "S" (that is used inside the single column of the output table). By default, this string "S" is limited to 512 characters and this is way too short to be able to extract all the columns of nearly all the "wide" tables (such as the "BSAK" table from SAP).

One first solution (to this "512 characters-limit" problem) is to avoid to extract all the columns from the SAP table and only extract a subset of the columns (i.e. to stay below the 512-character limit). To do so, you can just select, using the parameter **P2** and **P3**, the minimum number of columns that are required to do your analysis. To help you, when you click on **P3**, Anatella displays here in real-time the number of characters required to do the extraction (this is very handy!):



This means that you can click here: to easily select with your mouse the columns to extract and directly see if you are still below the 512-character limit here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When you click on **P3**, Anatella connects to the SAP sever to retrieve information about the columns in the currently selected table. If the connection to SAP fails, Anatella will display the standard "Column chooser" window (to still enable you to edit your column's selection). So, if you see the "Column chooser" window when you click on **P3**, it means that you need to check your log window for more details on why the SAP connection failed.

When you click on **P3**, Anatella also displays the total row count of your currently selected SAP table. To get this row count, Anatella calls the SAP function named "EM_GET_NUMBER_OF_ENTRIES". If you don't have this function inside your SAP system, or if you don't have sufficient priviliges to execute this function, then Anatella will not be able to display the "Row Count" (No worries: everything else still works without any trouble: e.g. the table extraction and the column selection still works ok). If you don't see any row count, check the Anatella log window to get more details on why you couldn't execute "EM_GET_NUMBER_OF_ENTRIES".

For analytical tasks, the above solution is not very convenient because, when you are working on an analytical project, you don't know in advance what are the columns that you'll need later on (most of the time). So, you want to be able to extract all the columns easily so that you have everything available, just in case you need it later. This is when the parameter **P10** comes into play (by default, it's "512"). The parameter **P10** informs Anatella that it cannot extract more than **P10** characters using one connection to SAP. So, a possible workaround (to the "512 characters-limit" problem) is to open several simultaneous connections to SAP! Each connection extracts a different subset of the user-selected columns (to always stay below the character limit defined in the parameter **P10**) and, finally, Anatella re-combine everything into one single "wide" table. In other words, Anatella merges into one table all the different small tables obtained from the different small extractions (that are each running using a different SAP connection). Don't worry if this explanation looks complex: In reality, everything is totally transparent and non-visible to the final user. When Anatella opens several simultenous connections to SAP, you'll just see inside the Anatella log window the following warning message:

```
Running... (31/10/19 17:30:07)

WARNING in SAP Extraction: The current row-length is greater than the
maximum admissible length (1806>512 characters): We will thus use 4
connections to perform the extraction (to avoid this situation, you should
extract less columns).

Success! (finished at 31/10/19 18:25:35 after 55.48 minutes - Peak Memory
Consumption~ 921 MB)
```

The obvious advantage of opening several simultaneous connections to SAP is that you are not limited anymore to 512 characters per row. The downsides are the following:

- The load on the SAP server is slightly higher.
- Since we are using several different connections to extract the data from SAP, we cannot enforce any ACID properties anymore. To know more about the ACID properties of databases, please refer to this page: https://en.wikipedia.org/wiki/ACID

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

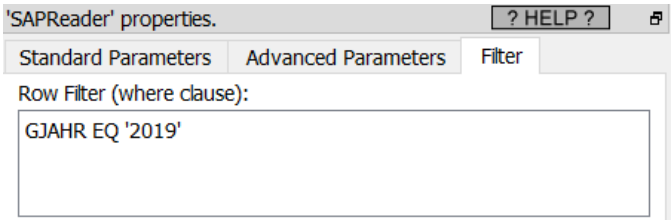
To summarize, it can happen a very **un**common situation where we have some inconsistencies between the different columns that were extracted using the different connections.
To avoid this inconsistent situation, the Anatella engine runs all the required extractions simultaneously, at the exact same time, so that it's almost impossible to obtain inconsistant data. Just to be sure, it's also better to run your extraction during the night, when the load on the SAP server is at a minimum.
You shouldn't care too much about this last point because, the most common database that is used "behind-the-scene" in SAP is DB2 and DB2 does not support any ACID properties anyway.

Another common cause of failure of the extraction is an excessive RAM consumption.
An "out-of-RAM" memory error can happen…

- …during step 1: During this step, the SAP system runs a heavy "SELECT * FROM table" SQL command without placing any limits on the number of rows or on the number of columns returned by the "SQL SELECT" command.

  This means that, even if you used the Anatella parameter **P6** to extract only 1 row out of the SAP system, the SAP server will still run this heavy "SQL SELECT" command that can, potentially, materialize inside the RAM memory of the SAP server a (very heavy) table with millions of records. This is just horrible.

  To avoid this situation, you can use the parameter **P15** to reduce the size of the table returned by the "SQL SELECT" command. For example, if you want to extract from the table "BSAK" all the rows where the "Invoicing Year" (i.e. the column "GJAHR" in SAP) is 2019, you'll write:



  Please note that the SQL language used inside SAP is not a "standard" SQL: For example:
  - Simple operators used to compare field values are EQ, NE, GT, LT, GE and LE (equivalent to =, <>, >, <, >=, <=).
  - You need to place numbers in-between single quotes.

  This means that this "standard" SQL command:

```
SELECT * from BSAK WHERE GJAHR=2019
```

  …will not work inside SAP and you need to write instead:

```
SELECT * from BSAK WHERE GJAHR EQ '2019'
```

  (see also the screenshot here above to know how to write this row filter inside Anatella).

  If you see the following message inside the Anatella log window:

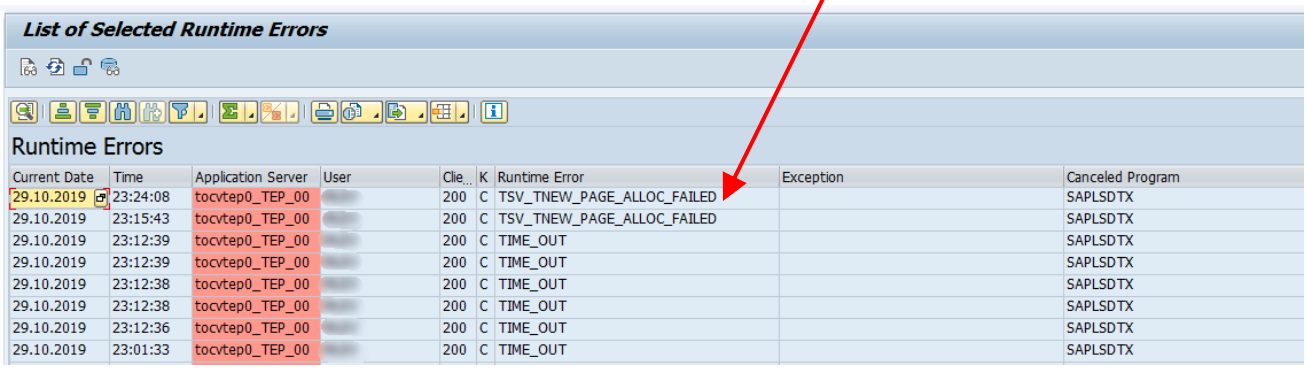
> SAP error calling function 'BBP_RFC_READ_TABLE'
> (SAPSQL_PARSE_ERROR: An error has occurred while parsing a dynamic entry.)

  …this means that the SAP system does not understand your user-defined "Where clause" SQL filter that is defined inside the parameter **P15** (i.e. There is a syntax error inside your "Where clause").
- …during the Steps 3 and 4: During these two steps, the SAP system creates in-memory the output table "T" (inside the RAM memory of the SAP server and, thereafter, inside the RAM

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

memory of the Anatella server). This output table "T" might be very large and consume too much RAM memory.

In such situation, you'll see inside the LOG of the SAP server:



Current Date	Time	Application Server	User	Clie...	K	Runtime Error	Exception	Canceled Program
29.10.2019	23:24:08	tocvtep0_TEP_00		200	C	TSV_TNEW_PAGE_ALLOC_FAILED		SAPLSDTX
29.10.2019	23:15:43	tocvtep0_TEP_00		200	C	TSV_TNEW_PAGE_ALLOC_FAILED		SAPLSDTX
29.10.2019	23:12:39	tocvtep0_TEP_00		200	C	TIME_OUT		SAPLSDTX
29.10.2019	23:12:39	tocvtep0_TEP_00		200	C	TIME_OUT		SAPLSDTX
29.10.2019	23:12:38	tocvtep0_TEP_00		200	C	TIME_OUT		SAPLSDTX
29.10.2019	23:12:38	tocvtep0_TEP_00		200	C	TIME_OUT		SAPLSDTX
29.10.2019	23:12:36	tocvtep0_TEP_00		200	C	TIME_OUT		SAPLSDTX
29.10.2019	23:01:33	tocvtep0_TEP_00		200	C	TIME_OUT		SAPLSDTX

To solve this error, you need to reduce the RAM memory consumption required to run the extraction: e.g. one solution is to reduce the number of rows inside the output table "T" that is used during the steps 3 and 4 of the ABAP function. To reduce the number of rows inside the output table "T", you can decrease the value of the Anatella parameter **P9**.

Let's now assume that:

- The Anatella parameter **P9** is 1000.
  In other words, the maximum size of the output table "T" is 1000 rows.
- There are, in total, 3500 rows inside table to extract from SAP.

In such a situation, Anatella will run the ABAP function 4 times (i.e. Anatella will run 4 extractions):

- Run 1: Anatella extract the rows 0 to 1000
- Run 2: Anatella extract the rows 1000 to 2000
- Run 3: Anatella extract the rows 2000 to 3000
- Run 4: Anatella extract the rows 3000 to 3500

Anatella will automatically merge the 4 tables obtained from these 4 extractions into a single large table. Everything is totally transparent and non visible to the final user.

The obvious advantage of using a small value for the parameter **P9** is that you will use less RAM memory on the server and, thus, your extractions are more likely to run successfully. The downsides are:

- A longer extraction-time because: If you use a smaller value for the parameter **P9**, you will be forced to make many more calls to the ABAP function to extract the full table. And, each of these "calls" introduces a significant overhead (i.e. the overhead is really big: it's the time required to run the steps 1 and 2 from the ABAP function). This overhead is heavily slowing down the extraction procedure.

  To reduce the overhead, you can use the parameter **P15** to add a "Where clause" filter because this effectively reduces the running-time of the steps 1 and 2 from the ABAP function.

- Since we are using several different runs to extract the data from SAP, we cannot enforce any ACID properties anymore

Finally, the SAP system sometimes returns a "TIME OUT error" for no apparent reason. No worries: When this happens Anatella just re-runs the extraction of the last block of rows (the size of this block of row is defined using the parameter **P9**) and the extraction continues without any problem. The

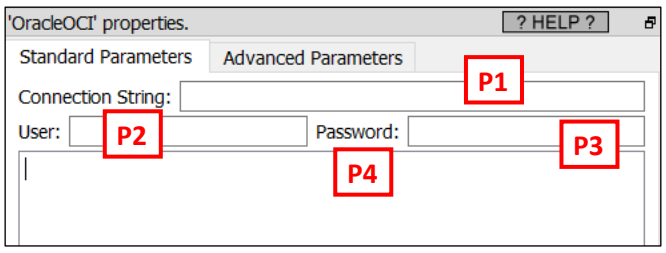
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

parameters **P13** and **P14** control how Anatella reacts to the TIME OUT errors. If you see many TIME OUT errors, it might be because you are extracting data from SAP at a "too high pace" for your (old) SAP system to handle: i.e. You need to slow down a little: i.e. You should increase the parameter **P12** to a higher value (e.g. 40).

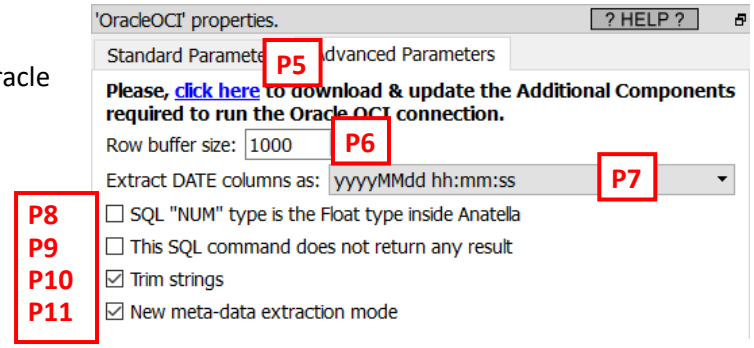
5.2.26. Oracle OCI Connector

Icon:

Property window:



Short description:
Extract data from Oracle



Long Description:
This action runs some SQL command to extract data from Oracle (i.e. it also runs any free-text SQL command inside Oracle).

The ODBC drivers from Oracle have strong limitations and we strongly suggest you to use the OracleOCI Action and the upsertOCI Action in lieu and place of the equivalent ODBC actions.

Here are some of the reasons why you might be interested in using this OracleOCI Action rather than the standard ReadODBC action (from the sections 5.1.6, 5.1.6.4 and 5.2.2) to extract data from Oracle:

1. If you want to insert data at high-speed inside Oracle you must use the upsertOCI Action. But, as soon as you start using the upsertOCI Action, all ODBC connections to Oracle fail. There are no other choices anymore than using the OracleOCI Action to access Oracle (i.e. you cannot mix inside the same graph OCI-based actions and ODBC-based actions).

2. If you want a better control on the connection "timeouts": For very large extractions (i.e. an extraction of several TB), you'll quickly notice that the ReadODBC action generates a large quantity of "timeout" errors. In opposition, the Oracle OCI Action is a lot more stable and reliable. This action actually uses the same exact same libraires (i.e. the OCI libraries) that are also used inside the "SQLPlus" tool to connect to Oracle (this tool is

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

recognized to be the most stable and most efficient way to interact with Oracle).
3. The easiest way to connect Anatella to Oracle under Linux (inside Wine) is through this Oracle OCI Action (i.e. the setup of the ODBC connection for Oracle inside Wine is *much* more difficult).

You can use the ">" character at the start of the SQL field (parameter **P4**) to create dynamically the SQL command to execute (using Javascript and some Global Parameters). You'll find more details on "Graph Global Parameters" in the section 5.1.5.

Before using this action, you first need to download the additional Oracle OCI components (it's a 196MB download): i.e. Click the blue URL (parameter **P5**) to automatically download and install the Oracle OCI components.

The parameter **P1** (named "Connection String") can be a reference to the file "TNSNAMES.ORA", or it can be directly the contents of the file "TNSNAMES.ORA". For example, a "Connection String" such as this one is perfectly valid:

```
(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.1.1) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = XE)))
```

The parameter **P6** represents the number of rows that are extracted "in one block" out of the database. A higher value is better because it extracts larger "blocks of rows" and thus minimizes round trips of network data transfers (and thus **the data extraction runs faster**). Do not put a value too high neither, otherwise the Oracle drivers may crash (and it also consumes a little more RAM).

If you get an error while executing the OracleOCI Action that looks like that:

```
OCI FAILED: Cannot execute the SQL code(1): ORA-24333: zero iteration count
```

…this means that your SQL command is special: i.e. It's not a "SELECT" (that returns some rows from the database) but something else (that does not return anything). To fix the error, just check the checkbox **P9**.

If you get an error while executing the OracleOCI Action that looks like that:

```
FAILED: OCIEnvCreate()
OCI FAILED: Cannot execute the SQL code(1): □□
```

…this means that you used the readODBC action or the upsertODBC action to access Oracle. As soon as you use an ODBC-based connection to access data inside Oracle, all OCI-based connections stop working (and the other way around is also true). You cannot mix the two types of Oracle connections (ODBC and OCI) inside the same graph.

The solution is: To regain access to Oracle, close the Anatella windows (click the × button in the top-right corner) and re-open Anatella. Once Anatella is re-open you can re-start using OCI again.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

*Regarding the parameter **P8**:*

Almost always, the columns of the "NUM" types can be extracted and stored inside a "blue" column of the type "FLOAT" inside Anatella (see the section 5.1.2. about data types inside Anatella). Using a "blue/FLOAT" column inside Anatella is more efficient in terms of storage space on the SSD and in terms of running-time.

BUT, there exists a situation where a "NUM" column contains some numbers with a very high count of digits: i.e. there are some number with more than 14 digits inside the "NUM" column. Such "wide" columns cannot be store insided the type "FLOAT" inside Anatella (that is limited to 15 digits maximum). So, to be on the safe-side, the parameter **P8** is left, by default, unchecked, so that the "NUM" columns are extracted as simple texts that is unlimited in terms of number of digits (rather than extracted as the type "FLOAT" inside Anatella). For efficiency reasons, it might be worth considering checking the parameter **P8** (but, then, you should be sure to have numbers with less than 14 digits inside the "NUM" columns).

*Regarding the parameter **P7** and the way Oracle handles Dates*

A time-related column inside a table can contain:

- Some **Time** (hours+minutes+seconds)
- Some **Date** (year+month+dayOfMonth)
- Some **Time Stamp** (year+month+dayOfMonth + hours+minutes+seconds)

Inside Oracle SQL, when you create a new table, you can define a time-related column …

- ..as the SQL type "DATE" **or**
- ..as the SQL type "TIMESTAMP".

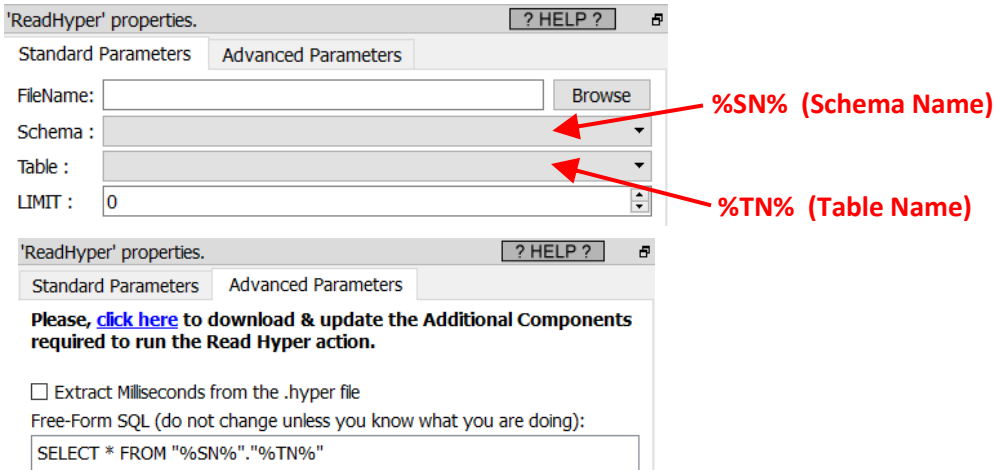
It seems logical to assume that the SQL "DATE"-type columns are containing some actual **Date** (i.e. they contain: year+month+dayOfMonth data), but Oracle does NOT actually enforce anything: i.e. inside a SQL "DATE"-type column, you can find any kind of content: you'll find "**Time**", "**Date**" and "**Time Stamp**" informations. So, to be sure to extract all the information contained inside a "DATE"-type column, the parameter **P7** is set, by default, to "yyyyMMdd hh:mm:ss".

BUT, if you were consistant when declaring your column's SQL types when you created your tables inside Oracle, and you made sure to declare a column with the SQL "DATE" type when it actually really contains some actual **Dates**, then you can set the parameter **P7** to "yyyyMMdd" to have a better, more compact data extraction.

5.2.27. Tableau .hyper File Reader

Icon:

Property window:



Short description:
Reads a table from a .Hyper file from Tableau.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
See section 5.1.1 to have more information on how to specify the filename of the .hyper file (i.e. you can use relative path and Javascript to specify your filename).

Since Anatella v2.46, you can execute ***any*** SQL command on the database contained inside the .hyper file. By default, Anatella extracts the whole table (that is selected inside the first panel named "Standard Parameters") using the following "default" SQL command:

SELECT * from "%SN"."%TN"

The Hyper SQL engine included inside Anatella supports all common SQL operations: e.g. "delete", "insert", "update". You can also use the "magic" > character to run a dynamic SQL command on the hyper database. For example, you can use:

> "DELETE FROM \"%SN%\".\"%TN%\" WHERE \"id\">"+myCutOffID

..to delete from your table all the rows with an "id" number that is above the value of the Global Parameter named "myCutOffID".

At some point, if you manually edit the Free-form SQL command, you might get the following error message:

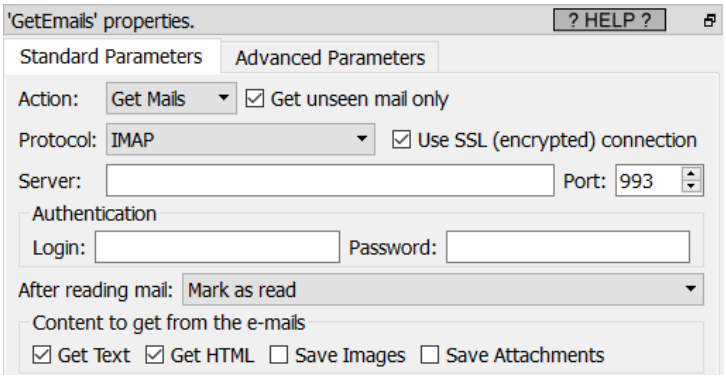
```
Tableau Hyper Error (62):syntax error: got identifier, expected "delete", "insert", "update": line 2, column 1: SyntaxError: Parse error
```

To fix this error just remove any carriage return inside your SQL command.

5.2.28. Get Emails

Icon:

Property window:

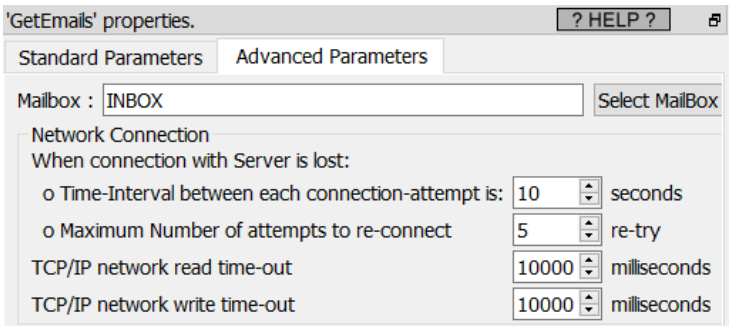


Short description:
Get emails from an IMAP or POP3 mail server.

Long Description:
Self explanatory.

After downloading an email you can:

- Do nothing
- Mark it as "read"
- Delete it

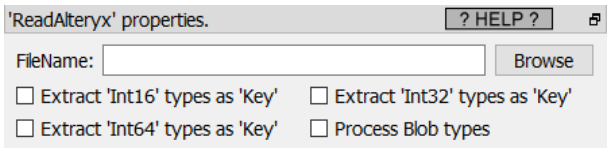


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.2.29. Read .yxdb Alteryx file

Icon:

Property window:

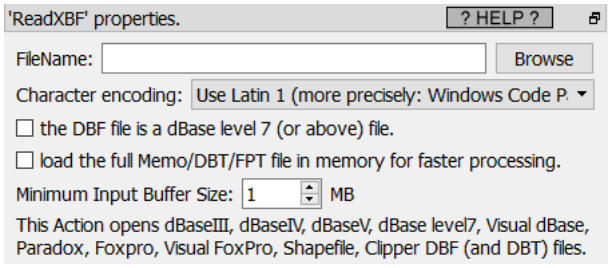


Short description: Read .yxdb Alteryx file

5.2.30. Read XDF file

Icon:

Property window:



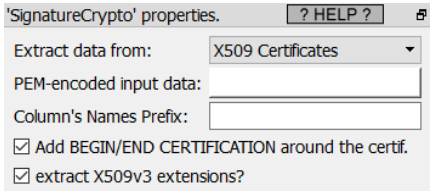
Short description:
Load a .dbf, .dbfs, .xbf, .dbt file.

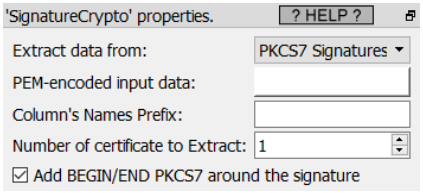
Long Description:
This Action opens dBaseIII, dBaseIV, dBaseV, dBase level7, Visual dBase, Paradox, Foxpro, Visual FoxPro, Shapefile, Clipper DBF (and DBT) files.

5.2.31. Extract Data from Cryptographic Certificates

Icon:

Property window:





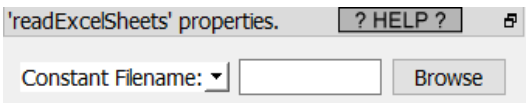
Short description:
Extract Data from Cryptographic Certificates.

Long Description:
Self-Explanatory.

5.2.32. Read Sheets Metadata from Excel files

Icon:

Property window:



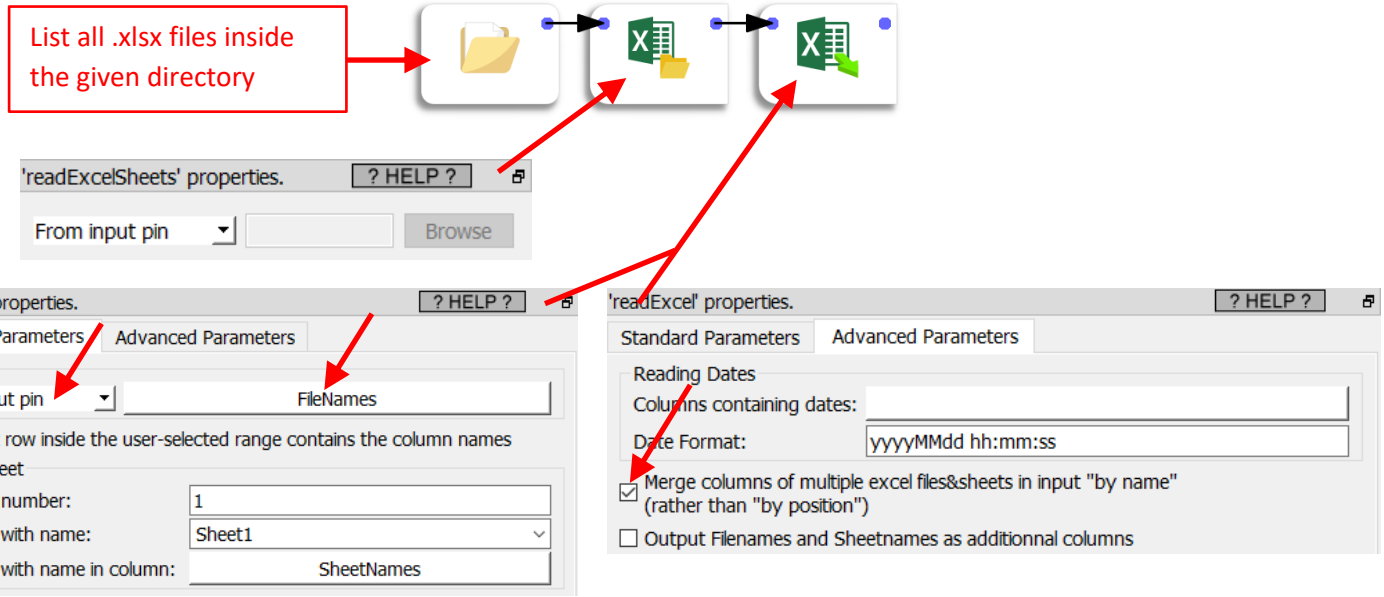
Short description:
Extract the Sheet names from Excel files.

Long Description:
This action can process any Excel file format (old and new: .xls and .xlsx).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

See section 5.1.1 to have more information on how to specify the filename of the .xls/.xlsx file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

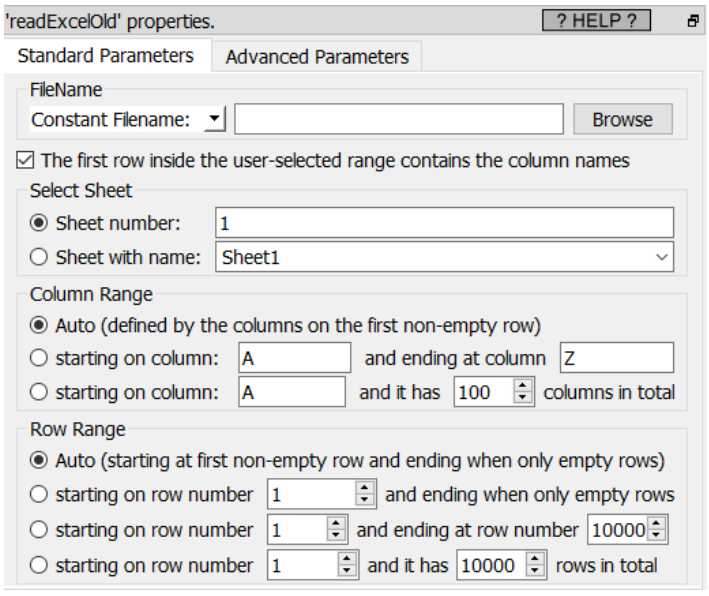
Typically, this action is useful when you want to read all the sheets from one (or many) excel files. For example, the Anatella graph here below outputs into a single big table the content of all the sheets from all the excel files stored in a given directory:



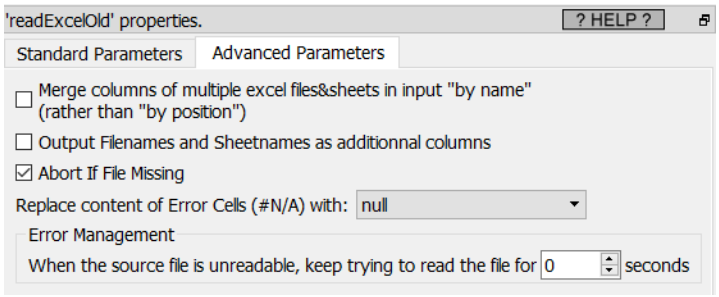
5.2.33. Read the Old Excel File format

Icon:

Property window:



Short description:
Read the older .xls Excel file format



Long Description:

The readExcelOld action reads only the older excel file formats: i.e. the files with the extentions .xls and .xlt (from the version old version of Excel before 2003). If you need to import into Anatella some data stored into the new excel file format (i.e. the files with the extensions .xls**x** or .xlsm), you should use the readExcel action (see section 5.2.9. for more details about this action).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

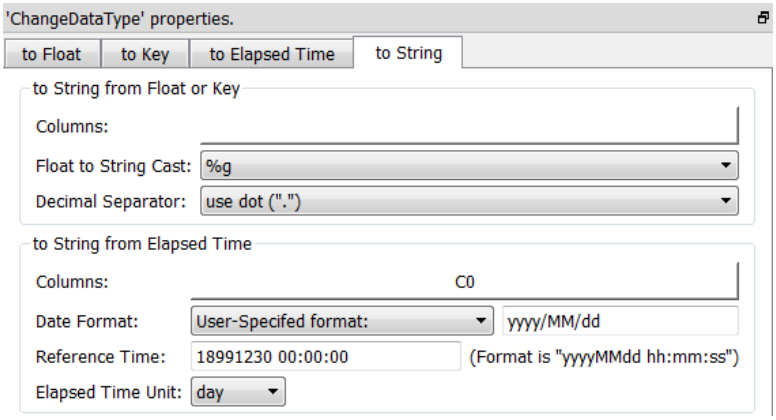
See section 5.1.1 to have more information on how to specify the filename of the .xls file (i.e. You can use relative path, wildcards, and Javascript to specify your filename).

You can drag&drop a .xls file from a MS-File-Explorer-Window into an Anatella-Graph-Window: This will directly create the corresponding ReadExcelOld Action inside the Anatella graph.

Anatella automatically recognize the "dates&times" that are stored in MS-Excel .xls files and directly import them in a human-readable-format ("yyyy-MM-dd hh:mm:ss"). If, for a reason or another, this recognition mechanism fails and you receive inside Anatella some dates stored as numbers, you can still convert these dates to "normal" Anatella dates using the "*to String from Elapsed Time*" option of the ChangeDataType Action. Use these parameters:

- Reference Time: 18991230 00:00:00
- Elapsed Time Unit: day.

Here is a screenshot:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.3. TA – Run Control (TA=Transformations Actions)

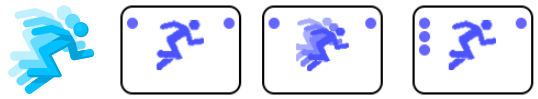
5.3.1. Run to finish Line

Icon:

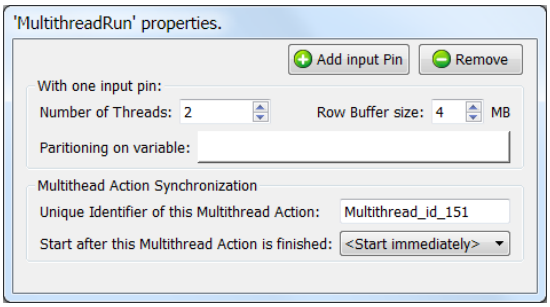
Property window: None

Short & Long description: See section 4.4.1.

5.3.2. Multithread Run

Icon: 

Property window:

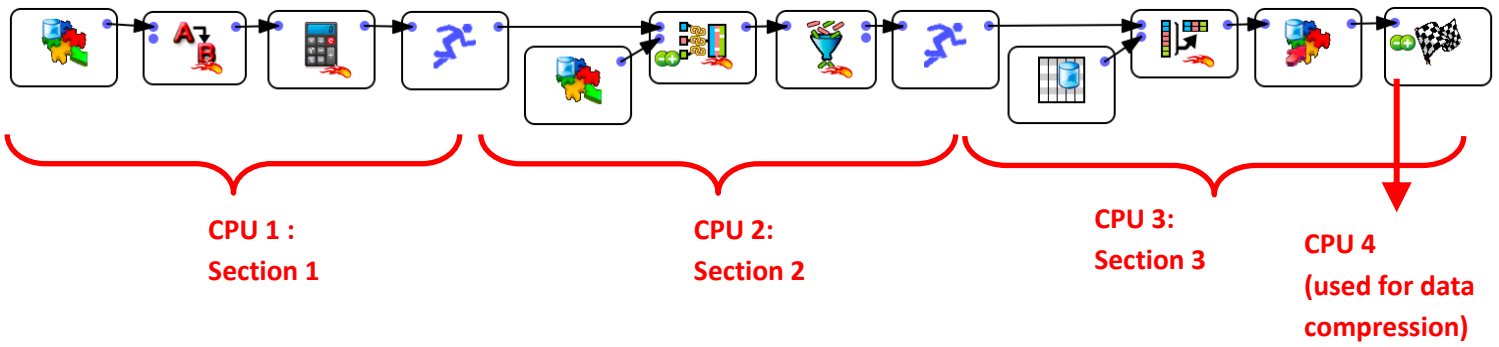


Short description:
Allows to use several CPU's/Threads to run the data transformation graphs.

Long Description:

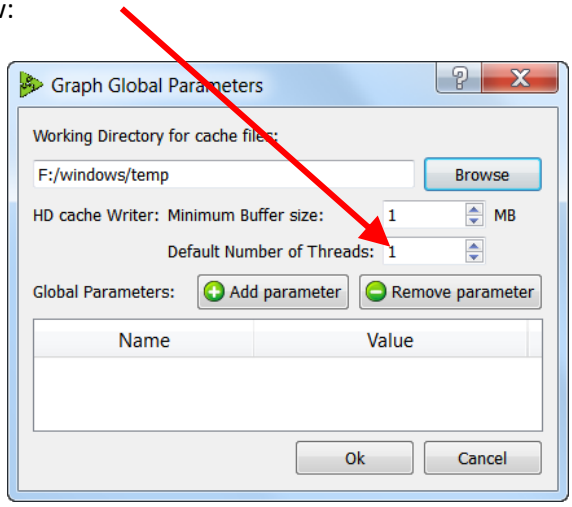
5.3.2.1. Defining Multithread Sections

Using the Multithread Action, you can cut your data-transformation graph in different sections, each section is running on a different CPU. For example, this Anatella-Graph runs on 3(+1) CPU's:



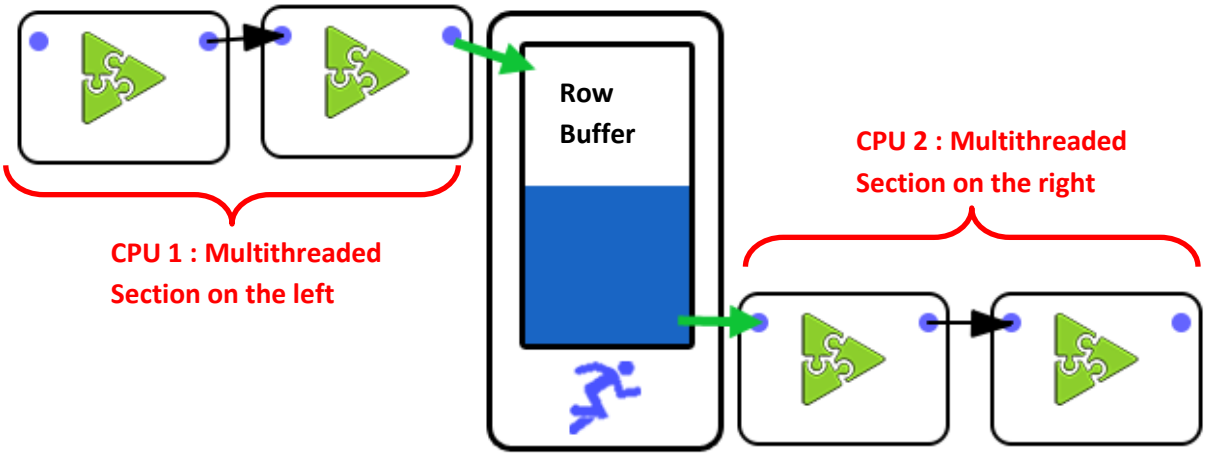
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The GelWriter Action always use one or more CPU "on its own" to compress the data before writing it on the hard drive. You can change the number of CPU's used for compression, in the "Graph Global Parameter" window:



Inside Anatella, the data is processed row-by-row. In the above graph, the Multithread Actions have 2 purposes:

1. They are defining the different "Multithreaded Sections".
2. They are passing-by the rows from one section to another.
   Typically, the different sections are processing the flow of rows at different non-constant speed. To be able to handle the small variations of data flow speed in each section, each Multithread Actions is equipped internally with a FIFO-row-buffer ("FIFO" is a technical terms meaning "First In, First Out"). Schematically, we can represent this FIFO-row-buffer in the following way:



You can think of the FIFO-row-buffer as a water-tank, except that, rather than filling it with water, you fill-it with rows.

If the section on the **left** of the FIFO-row-buffer is processing the rows at a very high rate (higher than the section on the right), then the FIFO-row-buffer will eventually reach "full capacity". In such situation, Anatella stops the execution of the **left** section for a little amount of time, so that the FIFO-row-buffer "empties out" a little.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If the section on the **right** of the FIFO-row-buffer is processing the rows at a very high rate (higher than the section on the left), then the FIFO-row-buffer will eventually be completely emptied out (no rows anymore). In such situation, Anatella stops the execution of the **right** section for a little amount of time, so that the FIFO-row-buffer "fills in" a little. This phenomenon is named "starvation": The CPU assigned to the **right** section is "starving" (it does not have enough rows coming in to fulfill its big "appetite").

To avoid stopping (temporarily) the execution of one of the section of the graph, all sections should process the rows at approximately the same speed. If all the sections have more or less the same row-flow-speed the CPU assigned to each section never stops (i.e. there is no "starvation") and we can reach maximum efficiency: The data-transformation-graph is running at maximum speed.

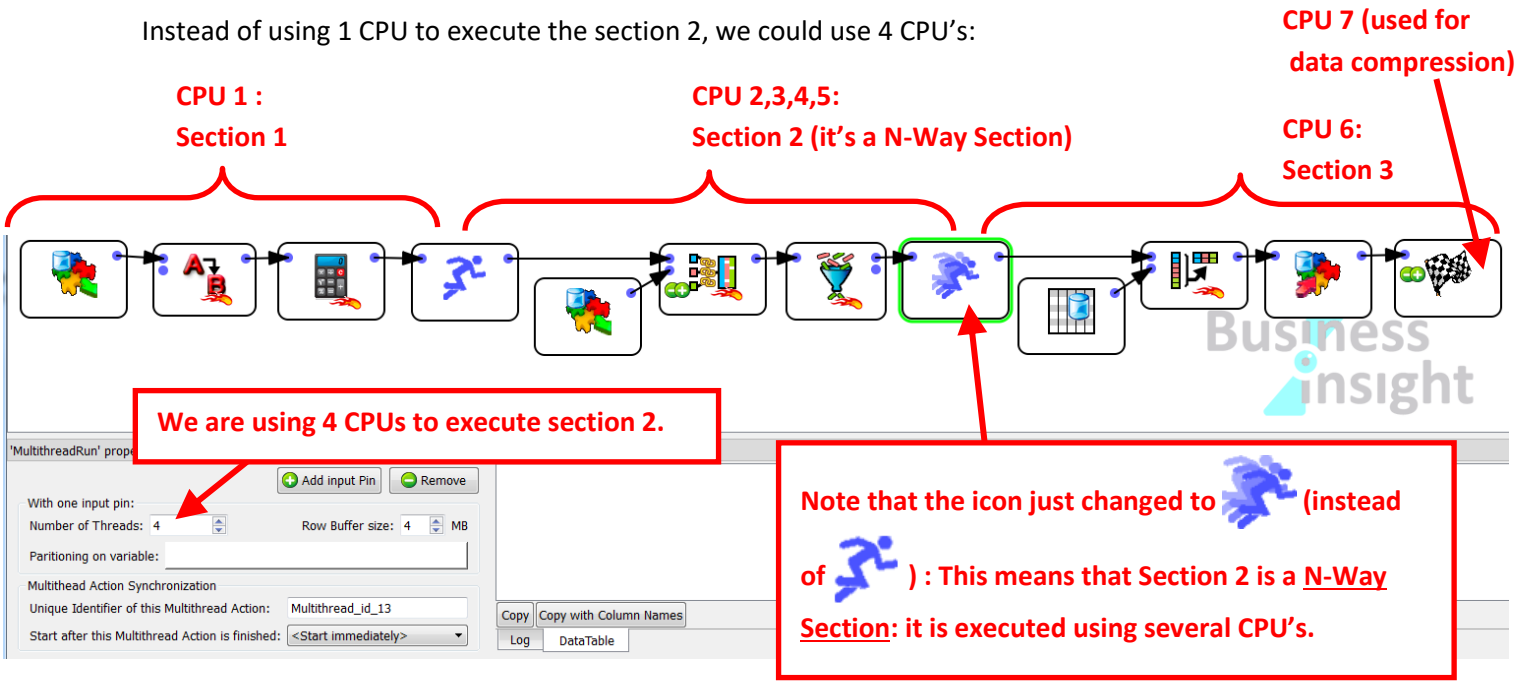
In general, the speed of the data-transformation graph is determined by the speed of the slowest multithreaded section (this is the "bottleneck" of the data transformation graph).

To avoid starvation (and thus reach maximum processing speed), you must correctly "balance" the workload of each segment.

5.3.2.2. Balancing Workload in different Multithread Sections

The multiJoin Action is usually quite slow (especially when there are many large "slave" tables). Thus, in the example above, the section 2 is the slowest. Section 2 thus drives the processing speed of the whole Anatella data-transformation graph.

Instead of using 1 CPU to execute the section 2, we could use 4 CPU's:



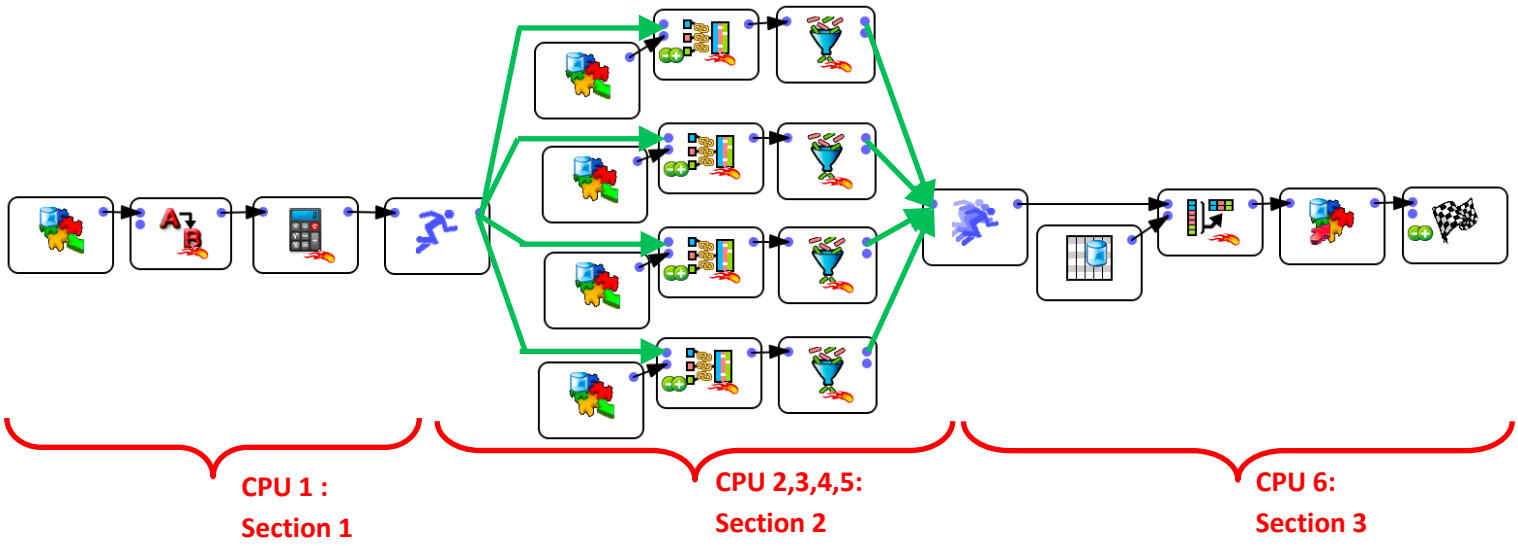
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

A section that is executed using several CPU's is named a "N-Way" section. To create a "N-Way" section, simply set the parameter "*Number of threads*" of the Multithread Action to the value 2 or higher.

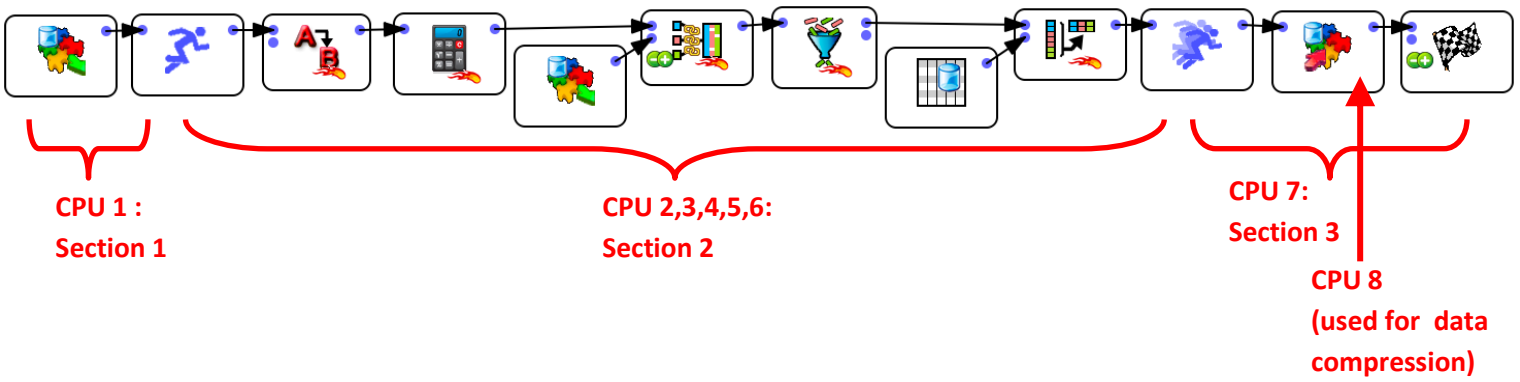
You cannot obtain any data preview of the output pins of the Actions included inside a "N-Way" section. To notify you of this limitation, these pins turn **RED** during graph execution. If, for some reason, there exists an HD Cache for a pin that turns **RED**, it will be deleted.

All HD caches are disabled (and deleted) inside "N-Way" sections. Thus, we suggest you to avoid using "N-Way" sections during the initial Anatella graph development phase. Once your graph has been tested and is working properly, you can start adding "N-Way" sections to increase processing speed.

Internally, when you ask to execute a N-Way section, Anatella automatically duplicates all the Actions inside the N-Way section and then dispatch a constant "flow of rows" to each of the duplicates. Each duplicate is running "in parallel", on a different CPU. If we go back to the above example, you could think of the "conceptual" graph that is *really* executed as the following one:



Now, with this new setting (**i**.e. with 4 CPU's for section 2), the slowest segment (i.e. the one that drives the global speed of the Anatella graph) could be section 1 or 3. It can be difficult to adjust properly the number of CPU's assigned to each section to obtain the highest processing speed. Let's avoid this difficulty by rewriting the same data transformation in a slightly different way:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The above data transformation graph is optimal in terms of execution speed because:

- Section 1 and section 3 are very fast because they each contain only one Action that consumes very little CPU. Thus, if there is a bottleneck, it should be in section 2 (and not in section 1 or 3).
- Section 2 is highly multithreaded (it uses 6 CPU's). If we could afford more CPU's, we would use them in section 2 because if, there is a bottleneck, it should be here.

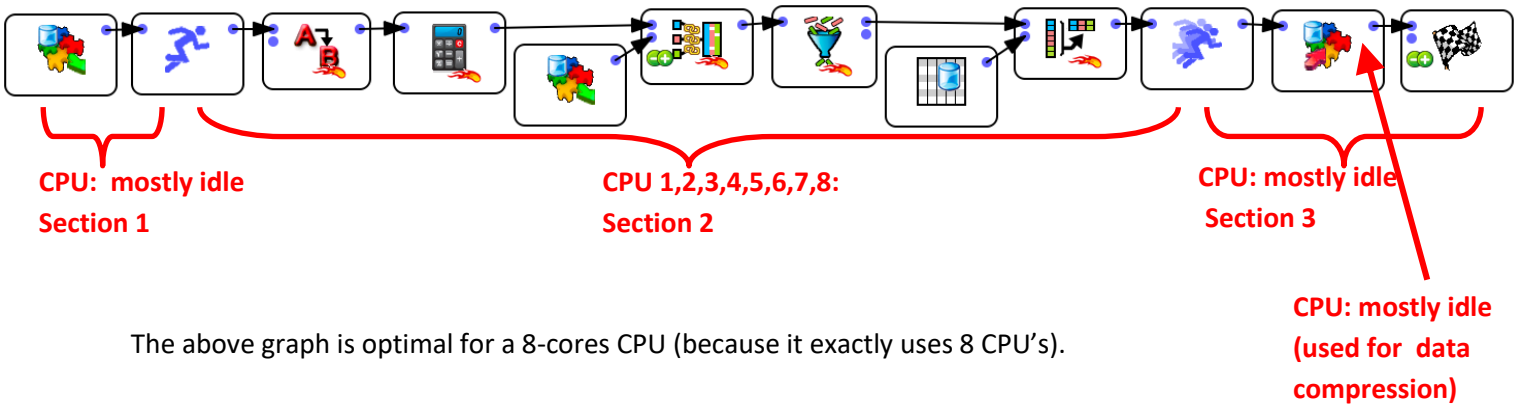
The above graph is *nearly* optimal for a 8-cores CPU (because it exactly uses 8 CPU's).
However, it can still be improved a little:

- The only Action included in Section 1 is reading a .gel_anatella file.
  Reading a .gel_anatella file is a 2 step-process:
  - Read a bunch of binary data (typically 1MB) from the Hard Drive and copy into main RAM memory. This takes 90% of the time.
  - De-compress the binary data into usable data. This takes 10% of the time.

  This means that the CPU assigned to Section 1 will be idle most of the time (90% of the time), simply waiting from the Hard Drive to complete the "Read" (i.e. the "step 1".). When a CPU is idle, we could use it to execute another section of the graph.
- The only Action included in Section 3 is writing a .gel_anatella file.
  Writing a .gel_anatella file is a 2 step-process:
  - Fill-in an internal memory buffer in RAM with a whole bunch of data (typically 4MB). (5% of the time)
  - Tell the compression-thread to compress the data (25% of the time) and flush the data on the hard-drive (i.e. copy the data from main RAM memory to hard drive)(70% of the time).

  This means that the CPU assigned to Section 3 will be idle most of the time (70% of the time), simply waiting from the Hard Drive to complete the "Write" (i.e. the "step 2".). When a CPU is idle, we could use it to execute another section of the graph.

Knowing that the CPU's assigned to section 1 & 3 will be idle most of the time, we can re-assign these 2 CPU's to section 2: We finally obtain:



The above graph is optimal for a 8-cores CPU (because it exactly uses 8 CPU's).

5.3.2.3. Input Actions and Multithread Sections

The ReadCSV Action is "injecting" new rows (that are read from the Hard Drive) into the graph.
Each time Anatella calls the ReadCSV Action, it does the following:

1. Look at the internal Row-Buffer of the Action:
   a. <u>If the Row-buffer is empty:</u>
      i. Read a bunch of data (typically 1 MB) from the Hard Drive and copy into the Row-Buffer (And, optionally, de-compress the binary data into usable data).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

ii. Select the first row in the buffer.

b. If the Row-Buffer is not empty:
Select the next row in the buffer

2. "Inject" the selected row into the Anatella graph and remove it from the Row-Buffer.

The ReadCSV Actions is using a synchronous (i.e. blocking) I/O algorithm (See the section 5.2.6.2. about asynchronous I/O algorithms). This means that, while Anatella is occupied reading some data from the Hard Drive (when it copies 1MB of data from Hard Drive to main RAM memory), it "freezes" the whole Multithreaded Section that contains the ReadCSV Actions. To avoid freezing the whole data-transformation-graph, it's a good idea to isolate the ReadCSV Action in a separate Section (thus using a Multithread Action).

The same remark applies to all the other "**Input Action**" that are based on a simple synchronous (i.e. blocking) I/O algorithm: the SASReader Action, the ODBCReader Action, etc. For example, you'll often have:

You should not use the above combination to increase the running speed of the Action that have asynchronous (i.e. non-blocking) I/O algorithms (See the section 5.2.6.2. about asynchronous I/O algorithms): these Actions include: the GelReader Action, the ColumnarGelFileReader Action, the readStat Action and the TcpIPReceiveTable Action.

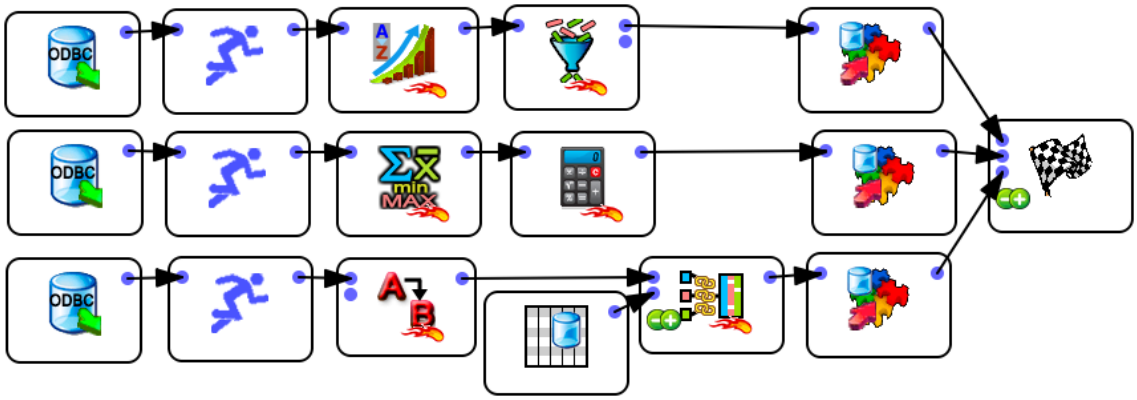
Conceptually, the logic is the same with both solution: Use an additional thread to allow the rest of the data-transformation-graph to run while the data are extracted from the Hard Drive. The difference comes from FIFO buffer located inside the multithread Action. This FIFO buffer implies a (slow) deep-copy of all the rows that are going through the multithread Action. When using an asynchronous I/O algorithm, you don't have to perform this deep copy at all and this is thus more efficient.

In this way, when the ReadCSV Action "freezes" (because it's waiting for the Hard Drive), it only blocks its own Multithreaded-Section but the rest of the transformation graph (i.e. the other Sections) can still continue to run (without any interruption), using the rows that are inside the FIFO-row-buffer

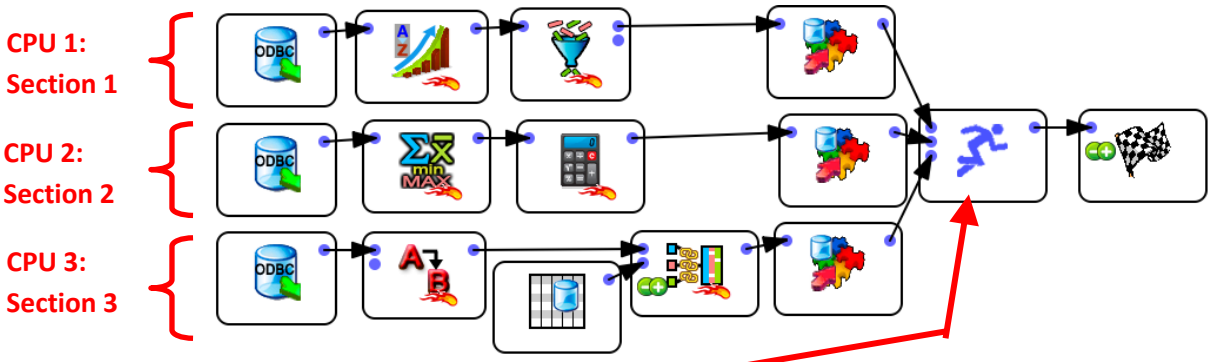
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

of the Multithread Action, just next to it. Of course, if it freezes for too long (i.e. if the Hard Drive or the database is very slow), then the the FIFO-row-buffer of the Multithread Action empties out and, once again, the whole data-processing stops (this is sometime referred, in technical terms, as a "Pipeline Stall"). This happens very often with the ODBCReader Action because the databases systems are usually very slow compared to Anatella Graphs: More precisely: Databases have usually some difficulties to "deliver" the rows at the high-speed required by Anatella for optimal execution speed. In such common situation, one way to reach high-processing speed is to run "in parallel" different SQL extractions, in different Multithreaded Sections.

For example, this is not very efficient (despite the fact that it includes several Multithread Actions):



The above graph will run the 3 database extractions one after the other and the processing speed will most likely be quite slow because of "Pipeline Stalls". The following graph (that runs the 3 database extractions "in parallel") is a better solution:



You can think of the Multithread Action with several input pins as the "multithreaded" equivalent of the standard GlobalRunFlag. The GlobalRunFlag executes the graph **sequentially**: First, the GlobalRunFlag runs all the Actions connected to the pin 0.

Once it's finished, the GlobalRunFlag runs all the Actions connected to the pin 1,

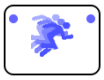
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Once it's finished, the GlobalRunFlag runs all the Actions connected to the pin 2,

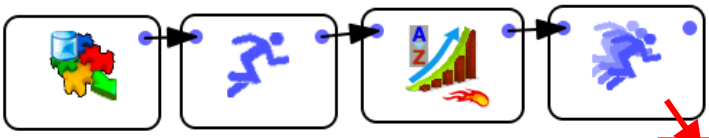
Once it's finished, the GlobalRunFlag runs all the Actions connected to the pin 3, etc.

In opposition, the Multithread Action executes the graph in **parallel**. As soon as you run the Graph (e.g. as soon as you pressed F5), all the Action connected to the Multithread Action start running at the same time. You can still have some control on the order in which the Actions are executed using the "Synchronization" option of the Multithread Action.

5.3.2.4. Limitations of N-Way Multithread Sections

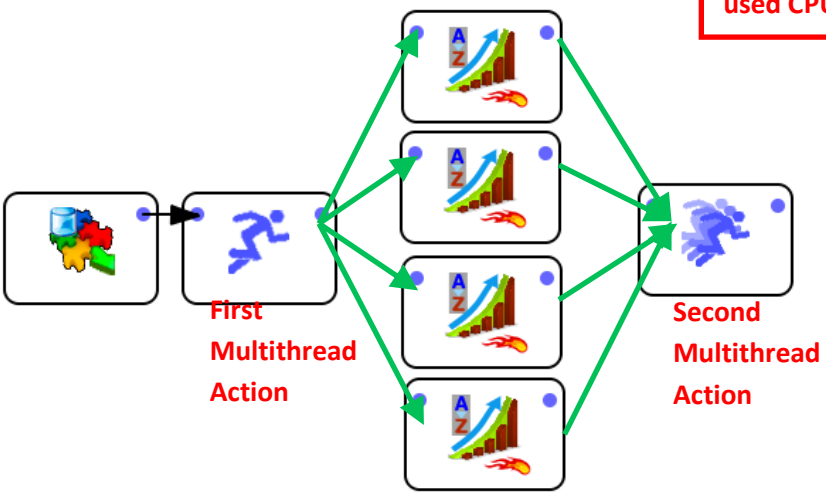
Let's call "N-Way Sections", the Multithreaded Sections that are executed using **several** CPU's: Such sections are represented using the icon (instead of the "standard" icon). A "N-Way Section" ends with the  Action and always starts with a Action.

Not all Actions can be included inside N-Way sections. For example, the Action or the Simple Join Action cannot be included inside a "N-Way Section". Indeed, if we had the following (non-working) graph:



Number of used CPU=4

… This would be equivalent to the following "conceptual" graph:



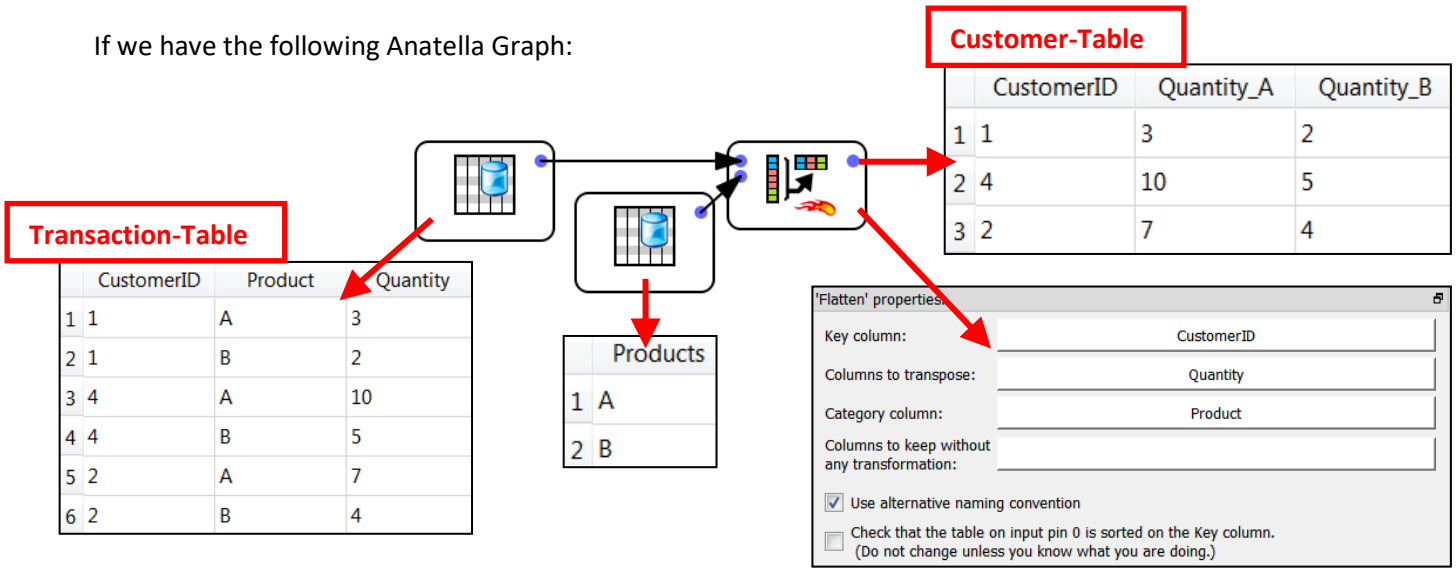
The above illustration explains that each Sort Action process (i.e. sort) one quarter (¼) of the total number of rows. At this point, we only have a "partial" sort (and not a "complete" sort) because

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

each sort was done on one quarter (¼) of the whole table (and not the whole table). Furthermore, the rows that are "going out" of the Sort Actions are put together, in a random order, inside the FIFO-Row-Buffer of the "second Multithread Action". Thus, even the (partial) sorting of the row is lost (because of the random order of the rows inside the final FIFO-Row-Buffer on the right).

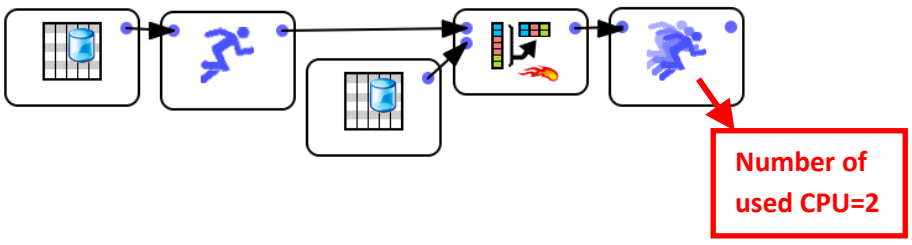
5.3.2.5. Overcoming the Limitations of N-Way Multithread Sections

If we have the following Anatella Graph:



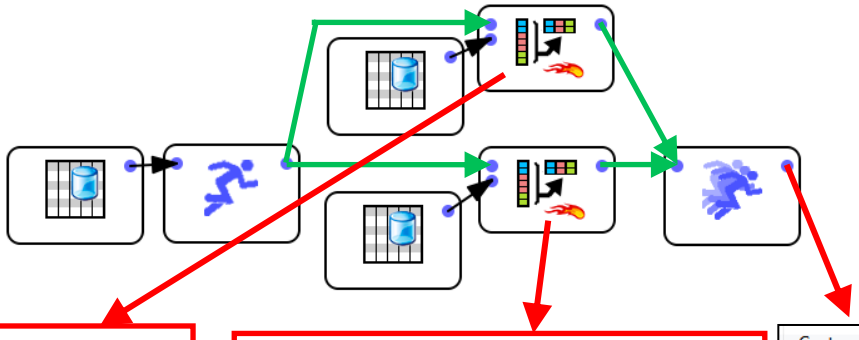
We have a transaction-table, where each row represents one transaction: "*How many items of product A or B did the customer X buy?*". Using the "Flatten" Action, we transform this transaction-table into a Customer-table (where each row represents a customer).

The "Flatten" Action reads the "CustomerID" column in the table on the input pin 0 and, **each time the "CustomerID" column changes**, it ouputs one row containing all the data about this "CustomerID". If we execute the "Flatten" Action inside a N-Way Section, we'll have:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The above graph is equivalent to the following "conceptual" graph:



This action will receive in input ½ of the table. More precisely, it will receive:

CustomerID	Product	Quantity
1	A	3
4	A	10
2	A	7

…and it generates as output:

CustomerID	Quantity_A	Quantity_B
1	3	
4	10	
2	7	

This action will receive in input ½ of the table. More precisely, it will receive:

CustomerID	Product	Quantity
1	B	2
4	B	5
2	B	4

…and it generates as output:

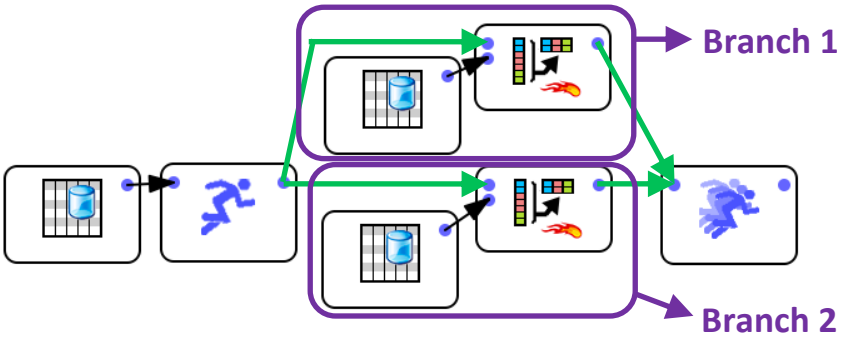
CustomerID	Quantity_A	Quantity_B
1		2
4		5
2		4

CustomerID	Quantity_A	Quantity_B
1	3	
1		2
2	7	
2		4
4	10	
4		5

Erroneous output

With the default settings, The "Flatten" Action is not working properly inside a N-Way section. We will now see how to correct this situation.

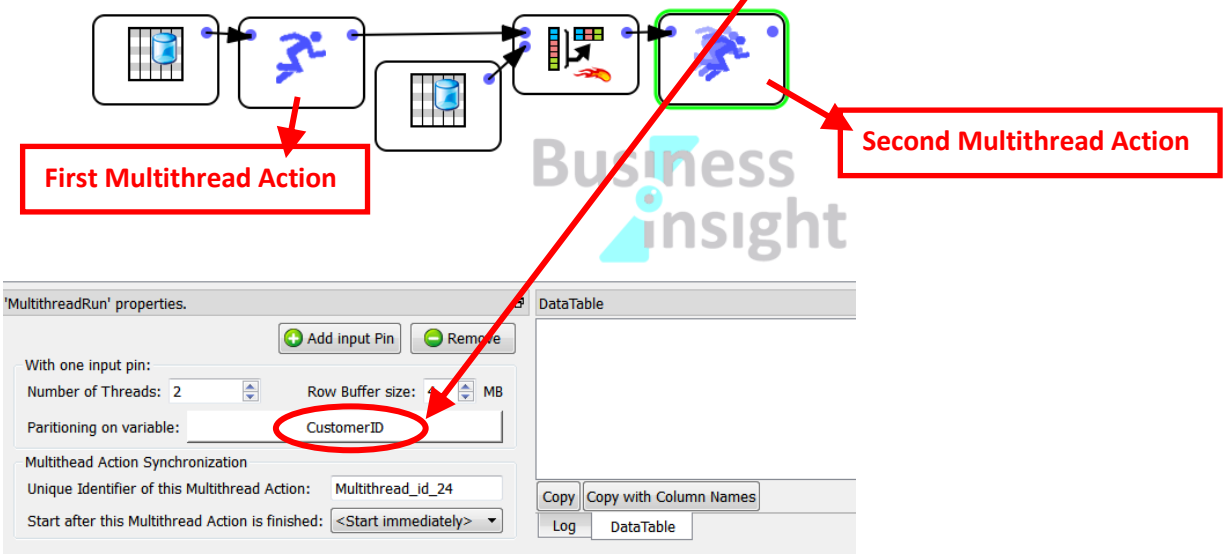
Let's now define "*Branch 1*" and "*Branch 2*"in the following way:



In the above example, the "Flatten" Action inside "*Branch 1*" receives only a fraction of the rows related to a specific customer (e.g. For the "CostumerID=1", it receives only the row about the "A" product and nothing about the "B' product). It thus produced a wrong result (e.g. for the "CostumerID=1", it outputs only the quantity of A product and nothing for the B product). In order for

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

the The "Flatten" Action inside "*Branch 1*" to work properly, it should receive all the rows related to a specific customer (e.g. it should receive all the rows related to the "CustomerID=1"). With the default settings, the rows that enter into the first Multithread Action are forwarded more-or-less randomly to either "*Branch 1*" or "*Branch 2*". To guarantee that all the rows related to the same customer are going inside the same branch, we must use the "partitioning" parameter of the Multithread Action: For example, this will work properly:



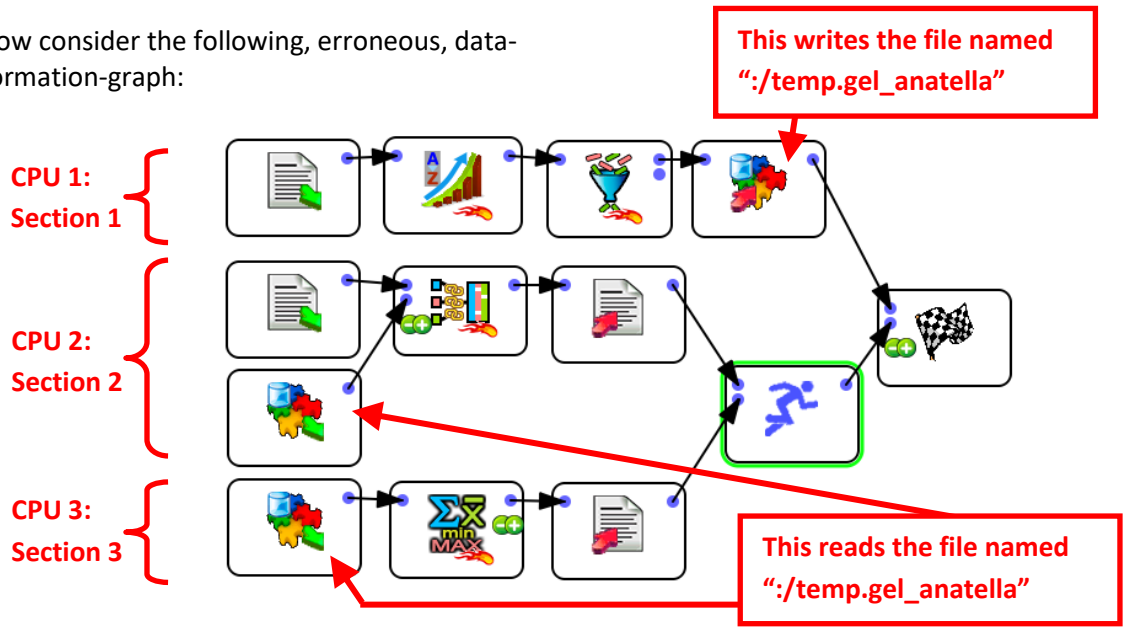
The "partitioning" parameter basically says to the **first** Multithread Action: "*Continue to send the input rows to the same branch, as long as the CostumerID stays the same*" (Beware: please note that this "partitioning" parameter is set in the parameter panel of the **second** Multithread Action). Thanks to the "partitioning" parameter, we can run in parallel an even larger variety of Actions (such as the "Flatten" Action, the Aggregate Action, the Partitioned Sort Action, the kpi_Stock Action, etc.).

To know more about how to overcome the limitations of the N-Way Multithread Section, please refer to section "*5.5.5.2. Using the "out-of-memory mode" of the Aggregate Action inside a N-Way Multithreaded Section*".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

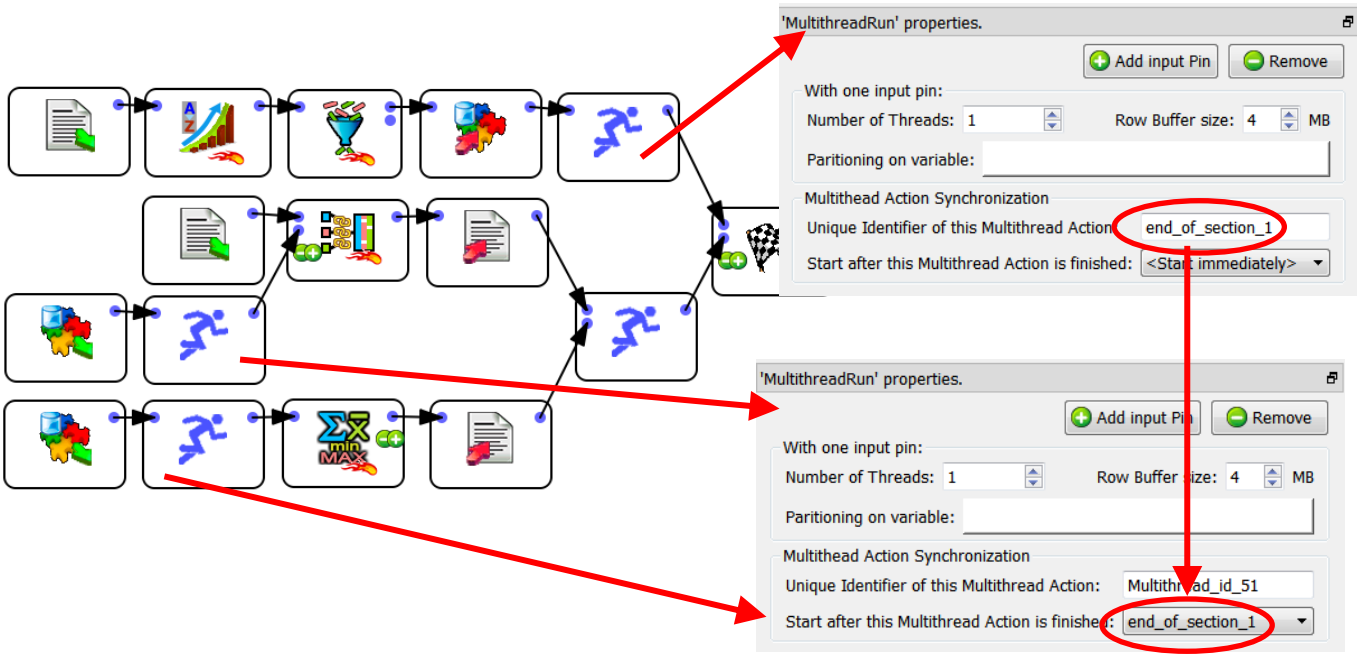
5.3.2.6. Thread Synchronization

Let's now consider the following, erroneous, data-transformation-graph:



The above data-transformation-graph won't run properly. Here is why: When we start the execution of the graph, the Sections 1, 2 & 3 will all start together, at the same time. In particular, the two ReadGel Actions (that are inside Section 2 and section 3) will directly fail because the file ":/temp.gel_anatella" has not been computed yet.

For the above graph to run properly, we should wait for complete creation of the ":/temp.gel_anatella" file (this is done in Section 1) and, then (and only then), we can start reading the file (in Sections 2 & 3). In other word, we need to wait for the end of the execution of Section 1 before reading the ":/temp.gel_anatella" file. This is accomplished in the following way:
This will work properly:

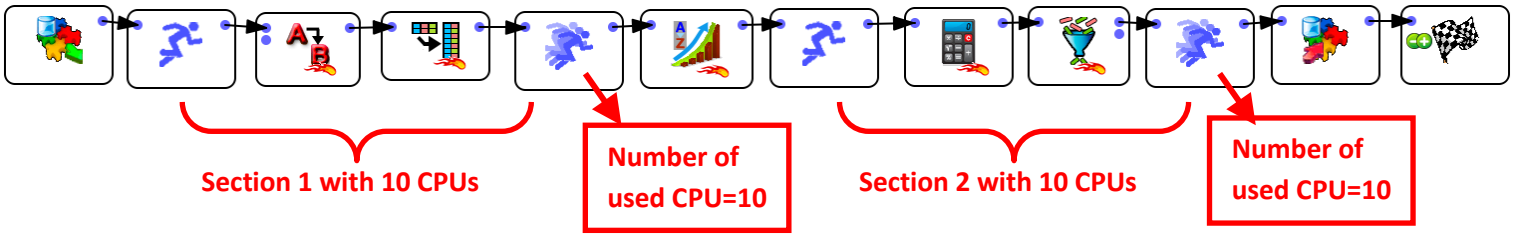


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Another solution (to ensure that the file ":/temp.gel_anatella" is created before any "read" occurs) is the following:

1. Copy the section 1 into a file named "section1.anatella".
2. Copy the sections 2 and 3 into a file named "section2and3.anatella".
3. Use the Parallel Run action to run the two Anatella graphs "section1.anatella" and "section2and3.anatella" **sequentially** (and not in parallel!).

Please note that, although there are three Multithreaded Sections in the above graph, it only uses a maximum of 2 CPU's at the same time (because Section 2 and 3 do not run at the same time as Section 1). This is a common phenomenon. For example:

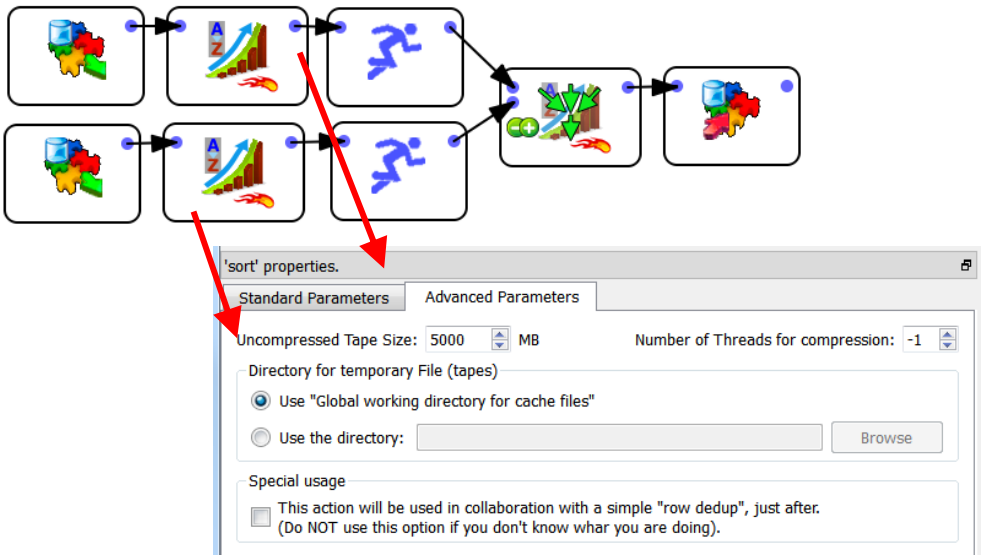


The above graph will NOT use 10+10=20 CPUs. Indeed, Section 2 will run only after the Section 1 is finished (this is because of the

Sort Action that waits until it received all "input" rows before emitting the first "output" row). The above graph will thus use a maximum of 12 CPUs.

From the above graph, you can see that it's not possible to run the

Sort Action inside a N-Way section. Does-it mean that all "sorts" are performed using only one CPU? No! The following graph computes the 2 "sorts" in parallel and outputs a globally **sorted** ".gel_anatella" file that contains all the data from the 2 input ".gel_anatella" files:



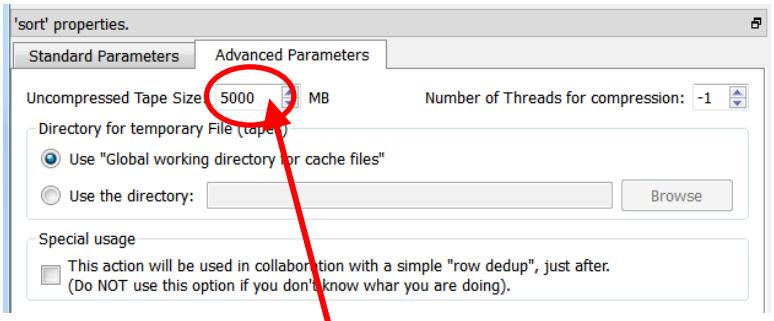
Please note that we used the MergeSort Action in order to produce a sorted output (The MergeSort Action is the equivalent of the Append Action with the slight difference that it

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

produces a sorted output). Using the above technique, it's possible to use many CPU's to perform a sort.

5.3.2.7. Main RAM Memory Consumption

What's the main RAM memory consumption of the above graph?



Each

Sort Action consumes 5 GB RAM: Since these two

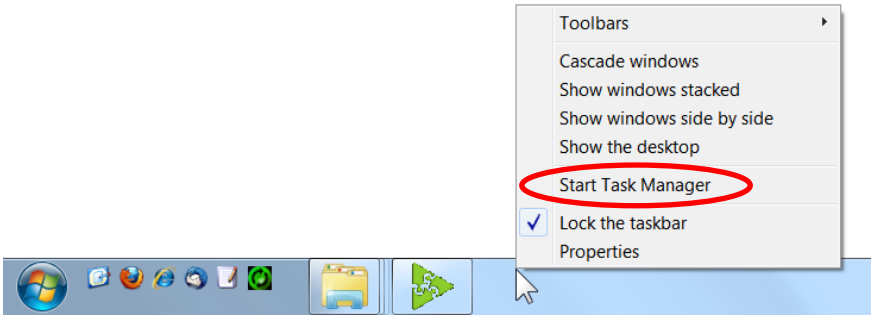
Sort Actions are running in parallel, the total RAM memory consumption of this graph is 5GB+5GB=10GB. The same graph executed on 1 CPU (i.e. without the Multithread Actions) consumes only 5GB RAM.

When you are executing in parallel some Anatella graph, the total RAM memory consumption usually increases substantially (compared to the "1-CPU-execution"). In the above example, the memory consumption for the "parallel" execution rises to 10 GB, compared to only 5GB for the simple, "sequential" execution.

If the total RAM memory consumption exceeds the amount of physical RAM inside your computer, then you are in trouble because MSWindows will be forced to "swap" (also named "paging"). When MSWindows "swaps" all processing speed is divided by 100 or more: The computation becomes so slow that it's better to stop all computation and re-design your data-transformation-graph to use less RAM.

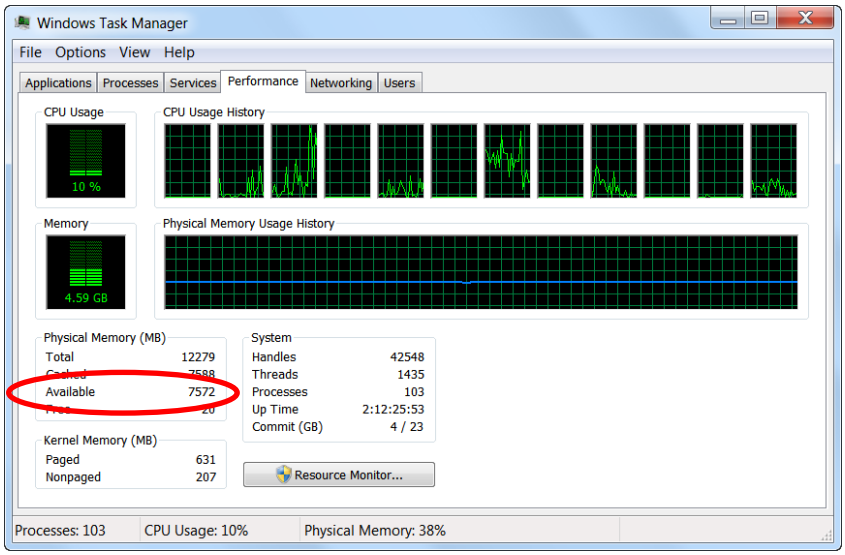
You can check the total physical RAM memory available to you and the memory consumption of your Anatella graphs, by looking at the MSWindows "Task Manager".

To run the MSWindows "Task Manager", right-click the MSWindows Taskbar and select "Start Task Manager":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The amount of total physical RAM memory available to run your graphs is visible here:



In the above example, I still have 7572 MB ≈ 7.5 GB of Available Physical RAM memory to run my Anatella graphs. When you start your data-transformation-graph, Anatella start consumming some RAM memory and the amount of Available Physical RAM memory decreases. When this amount drops to zero, MSWindows will start "swapping" and all computations (from the whole computer) nearly stop. You should avoid that!

The Actions that consume a great quantity of RAM memory are:

- The MultiJoin Action: This Action starts by loading into RAM memory all the Slaves tables (on pin 1 and above). If there are a great quantity of large slave tables, the MultiJoin Action will consume a great quantity of RAM.
- The FilterOnKey Action: This Action works in the same way as the MultiJoin Action: It start by leading into RAM memory the table on the second pin. If this table is large, then it will consume a large quantity of RAM.
- The Sort Action: To sort large table, you need a large "tape size". It means creating a large buffer into RAM memory (that contains one tape data).
- The Aggregate Action (when the option "Use in-RAM algorithm for small output tables" is checked): All the output tables (i.e. all the "aggregates") must fit into RAM memory.
- Most of the Actions for "Graph Mining" are loading the whole graph to analyze into RAM memory. If this graph is large, then it will consume a large quantity of RAM. These Actions include the CommunityDetection Action, the NodeAnalysis Action, the SignificanceLevel Action, the Leadership Action.
- Most of the Actions for "Operational Research" are loading the whole table to analyze into RAM memory. If this table is large, then it will consume a large quantity of RAM. These Actions include the KNN Action and the DatasetReduction Action and the AssignmentSolver Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You should pay extra attention to RAM memory consumption when the above Actions are running in parallel. If you exceed the total amount of Available Physical RAM memory of your computer, all computations will slow down radically.

When using Anatella 32-bit (in opposition to Anatella 64-bit), the total RAM memory consumption can never exceed 2 GB. If you try to use more than 2GB RAM, Anatella 32-bit will stop (and, most of the time, it will stop abruptly with the message "segmentation fault"). This limitation is imposed by the hardware and MSWindows.

On the contrary, Anatella 64-bit does not have any limitation on the total amount of RAM memory that it can use. However, if you consume more RAM than the total amount of Available Physical RAM memory of your computer, all computations will slow down dramatically.

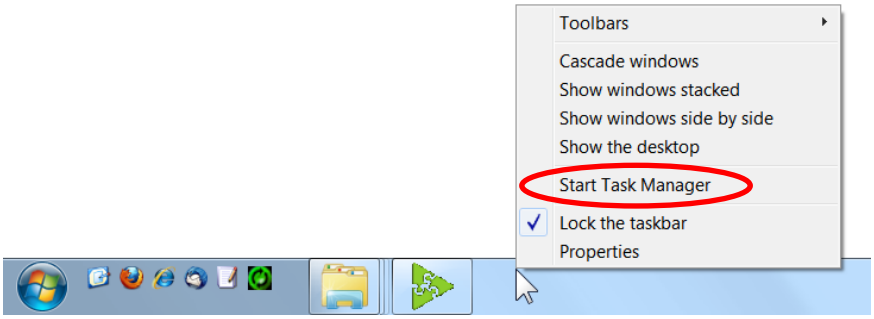
Is it possible, using Anatella 32-bit, to use more than 2GB RAM memory?

Yes: Using the ParallelRun Action, you can run several Anatella-graphs "in parallel". Each graph runs inside its own process (i.e. inside its own "anatella.exe" application) and uses a maximum of 2GB RAM. The sum of the RAM consumption of different processes can be above 2GB.

5.3.2.8. Fine-Tuning your script for Best Multithread Speed

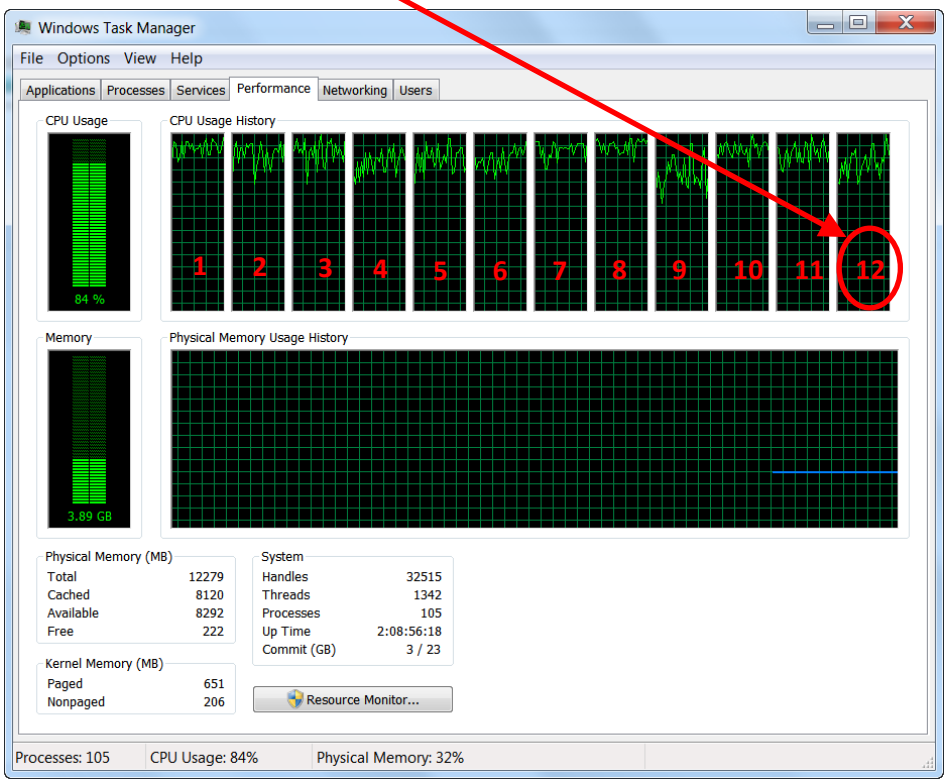
You can check if the workload of your different Sections is correctly "balanced", by looking at the MSWindows "Task Manager".

To run the MSWindows "Task Manager", right-click the MSWindows Taskbar and select "Start Task Manager":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The MSWindows “Task Manager” appears. You can see the number of cores (i.e. this is more or less equivalent to the number of CPU’s) here:



You can see that, on this example, there are 12 cores (i.e. this is more or less equivalent to 12 CPU’s).

You can estimate how much CPU a running Anatella Graph is using: On a 12-core system:

When Anatella fully uses 1 CPU, the “CPU” column displays 8 % (≈1x100/12).
When Anatella fully uses 2 CPU’s, the “CPU” column displays 16 % (≈2x100/12).
When Anatella fully uses 3 CPU’s, the “CPU” column displays 25 % (≈3x100/12).
When Anatella fully uses 4 CPU’s, the “CPU” column displays 33 % (≈4x100/12).
When Anatella fully uses 5 CPU’s, the “CPU” column displays 41 % (≈5x100/12).
Etc.

If your Anatella graph is designed to use 4 CPU’s, then the “CPU” column inside the MSWindows “Task Manager” should display 33% (on a computer that has 12 cores). If that’s not the case (e.g. you get 12%), it means that some Multithread Sections are “starving” and you are not completely using your 4 CPU’s. Maybe you should re-design your Anatella graph to balance in another way the computation workload to better use your 4 CPU’s.

You should also try to avoid using more CPU’s than the than the real physical amount of CPU’s inside your server because, in this situation, MSWindows must “emulate” by software the missing CPU’s and this leads to a very large speed drop: You’ll have “pipeline stall” and “starving” everywhere. For example: you have a 4 CPU server and your Anatella graph is designed to use 7 CPU’s: This is bad: Your data-transformation will run in very inefficient way: Try decreasing the number of CPU used inside your graph.

The “emulation” of many CPU’s involves a procedure named “context switching”: For example, when you emulate two 2 *virtual* CPU’s using one *real* CPU’s, what you are actually doing is:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

a) Use your (only)*real* CPU for a few milliseconds to execute the computation that the **first** *virtual* CPU must do.
b) Save the "state" of the computation assigned to the **first** *virtual* CPU (A "state" typically includes the content of all CPU registers and a reset of the micro-instruction execution pipeline).
c) Restore the "state" of the computation assigned to the **second** *virtual* CPU.
d) Use your (only)*real* CPU for a few milliseconds to execute the computation that the **second** *virtual* CPU must do.
e) Save the "state" of the computation assigned to the **second** *virtual* CPU.
f) Restore the "state" of the computation assigned to the **first** *virtual* CPU.
g) Go back to step a).

The process described in the steps (b) & (c) is one "context switch" (The steps (e) & (f) are also one "context switch"). **"Context Switching" consumes a lot of CPU resources.** It can happen that 60% of the CPU computational power is used to execute "context switching". In such situation, only the remaining 40% of the CPU resources are available to compute the data transformation graph and it will thus run very slowly.

In the ideal situation, the "Task Manager" should display a number *near* 100% of CPU-consumption (e.g. 95% or 99%). If you try to use too much CPU's inside your graph (i.e. you try to go "above" 100% of CPU consumption), the computation speed will (most of the time) be reduced. Sometime (not very often), you'll directly see inside the "Task Manager" that your graph is actually slowing down (i.e. the CPU-consumption will drop around 50% because of the many "starvations"). Most of the time, the computational-speed of your data-transformation-graph will decrease but it won't be easily noticeable inside the MSWindows "Task Manager" because you'll see a 100% CPU-consumption that might fool you (The CPU is actually busy performing many "context switching").

Adjusting the amount of CPU used inside a graph can be tricky: We are always tempted to design our graphs to use more and more CPU's. But, at the same time, if we try to go "above" 100% of CPU consumption, the data-transformation suddenly slows down dramatically. Finding the right settings sometime requires a small "trial-and-error" procedure to get it right. Simply "time" your transformation graph on a sample, using different settings and keep the settings with the shortest computation time.

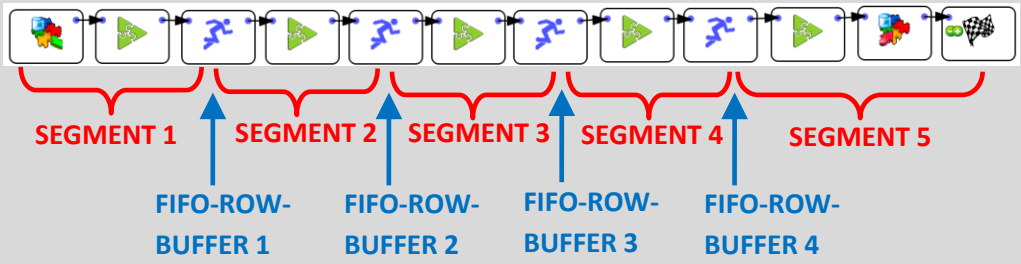
Be warned: There is a BIG penalty-hit if you try to use more CPU's than actually physically available inside your server. In the ideal situation, the "Task Manager" should display a number near 100% of CPU-consumption (i.e. 95% or 99%).

Why is it so bad to try to use more CPU's than the actual number of physically available CPU's inside your server?

One first, simple answer is that you'll lose CPU resources in "context switching". …but there is something even worse than that.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's assume that you have a 2-CPU server and you run this graph:



This graph is designed to use (at least) 5 CPU's. In between each of the 5 multithread segments, there is a Multithread Action that contains a FIFO-Row-Buffer (do you remember the "water tank" of section 5.3.2.1.?).

Let's now assume that SEGMENT 5 is very fast to compute. Thus, most of the time, the FIFO-ROW-BUFFER 4 will be empty (because, as soon, as there are a few rows inside the buffer, we'll process them all very quickly). In technical terms, SEGMENT 5 is "starving" most of the time. To give some rows to process to SEGMENT 5, we need to run SEGMENT 4 for a little while. …but we have only 2 *real* CPU's: The MSWindow task scheduler must assign one of the 2 *real* CPU's to SEGMENT 4. The MSWindow scheduler has no idea of the structure of your data-transformation-graph: It does not know about the **dependencies** that exist between the different SEGMENTS: e.g. It has no idea that, to un-block SEGMENT 5, it needs to run the SEGMENT 4 for a little while. Thus, it will choose **mostly randomly** one of the SEGMENTS to run. If you are lucky, it will be SEGMENT 4 and your data-transformation will "advance" a little bit further. If you are unlucky, it will choose a SEGMENT that has an input FIFO-ROW-BUFFER nearly empty: This SEGMENT will thus run for a very small amount of time (to process the small number of rows in input) and then stop (i.e. because of "starvation": no more rows to process), forcing the MSWindow task scheduler to execute one more CPU-costly "context switch" (searching for another SEGMENT to run). This will lead to a large amount of "context switching" (searching for the "right" multithread segments to SEGMENT to "advance" the data transformation) and, at the end, you obtain a very inefficient (i.e. really slow) data transformation.

Some poorly designed Multi-CPU multithreaded Anatella graphs might run slower than their 1-CPU single threaded equivalent (it's quite common for very small graphs, with only a few Actions). It's also common that some Anatella graphs should not be multithreaded at all (More precisely: Any graph can be multithreaded: The question is rather: Does the multithreading makes computation faster?).

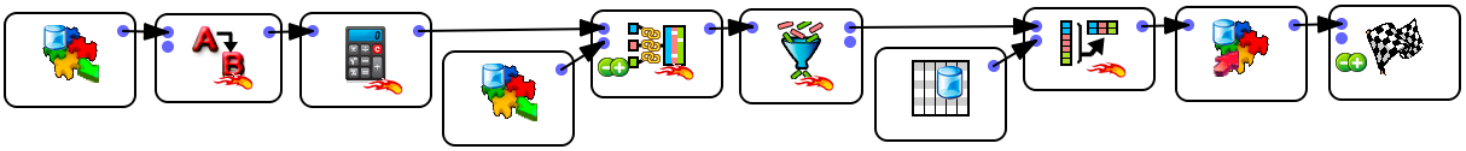
Summary: Because the MSWindow task scheduler does not know about the *dependencies* between the different multithread SEGMENTS inside your data transformation graph, it will lose a large amount of CPU-time in "context switching", searching (mostly randomly) for the "right" SEGMENT to execute. This, of course, leads to very slow computing speed for your data

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

transformation. Do not design your graph to use more CPU's than required. Sometime, 1-CPU single-threaded execution is the fastest.

5.3.2.9. About multithreading inside other ETL's

Most ETL's do not have multithreading functionalities. Some are using a very specific (and primitive) form of multithreading: They automatically insert a Multithread Action in-between every normal Action. For example, when you see the following data-transformation graph inside a "classical multithreaded ETL tool":



… this is, in reality, executed in the following way (note that we simply inserted a Multithread Action in-between every Action):

This means that, if you have a graph with twenty Actions, there will be (roughly) twenty CPU's used to run the data-transformation-graph. Twenty is a number that is (usually) significantly larger than the physical amount of available CPU's and **this graph will thus run very slowly** (because of the software "emulation" of the missing CPU's performed by MSWindows: See the previous section about this subject). **There is no way of preventing that**.

Each Multithread Action has an internal FIFO-row-buffer. The management of this FIFO-row-buffer consumes a big amount of precious CPU time. Thus, it's a good idea to reduce to the minimum the number of Multithread Actions in your graph: you can do that by designing your graph so that it has the longest Sections as possible (i.e. a section should include as many Actions as possible).

The data-transformation-engines that are used inside "classical" multithread ETL's are using a very large amount of FIFO-row-buffers (because they internally place a Multithread Action in-between every Action: All their sections are composed of only ONE Action: it's terrible!) and they are thus losing a very large amount of precious CPU time in managing all these un-necessary FIFO-row-buffers (and also in "context switching").

Most "classical" multithread ETL Engines do not even offer the equivalent of "N-Way Multithreaded Sections".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In opposition, inside Anatella, you have the complete control of:

a) how many CPU's your data-transformation is using.
b) how many FIFO-row-buffers your data-transformation is using.

This "total control" allows you to use in the most efficient way, all the CPU's inside your server.

5.3.2.10. Summary

The Multithread Action has 3 operating modes:

1. : This defines the border between 2 simple Multithread Sections. Simple Multithread Sections do not alter the order (i.e. the sorting) of the rows. It means that you can safely put these a little bit everywhere in the data transformation graph.
2. : This defines a N-Way Multithread Section (that is located just BEFORE the Multithread Action and NOT AFTER). The rows that exit a N-Way Multithread Section are not sorted anymore (i.e. N-Way Multithread Sections are removing all "sorting attributes" in the flow). You can use a partitioning variable to better control which rows are sent to which branch/CPU. Inside a N-Way Multithread Section you cannot have any HD cache.
3. : This is the multithreaded equivalent of the GlobalRunFlag Action. All the actions connected to the input pins of the Action are starting & running simultaneously.

Always design your data-transformation-graph without any multithreading. Once everything has been tested and is working properly, you can start adding Multithread Actions to improve processing speed.

You should try to use the minimum amount of Multithread Actions because the management of the FIFO-row-buffer (that is included inside the Multithread Actions) consumes precious CPU time (The Multithread Actions are a special case because they do not have any internal FIFO-row-buffer, because they output nothing, so you can have plenty of them). In the same spirit, always try to make the longest Sections as possible. A Section should include as many Actions as possible.

Combining together an Input-Action with a Multithread Action is usually a good idea (but not for the GelReader Action, the ColumnarGelFileReader Action, the readStat Action and the TcpIPReceiveTable Action).

Use the MSWindows "Task Manager" to:

a) Optimize your multithread parameters (i.e. the position and the content of the different Sections and the number of CPU assigned to each Section): Do not try to use more than 100% of CPU consumption: it will directly translate to a large efficiency penalty.

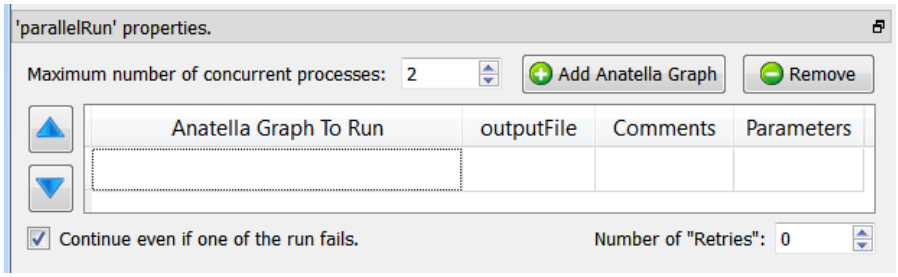
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

b) Optimize the memory consumption of your graph.
   I. When using Anatella 64-bit, do not use more RAM memory than the physical memory available inside your computer.
   II. When using Anatella 32-bit, do not use more than 2GB RAM memory.

5.3.3. Parallel Run

Icon:

Property window:



Short description:
Runs one or several Anatella (sub)Graphs.

Long Description:
Runs sequentially (When the parameter "*maximum number of concurrent processes*" is **one**) or in parallel (When the parameter "*maximum number of concurrent processes*" is **two or more**) several Anatella Graphs.

For large data-transformation, it can be interesting to divide all the transformations to perform into several graphs. Division is interesting for several reasons:

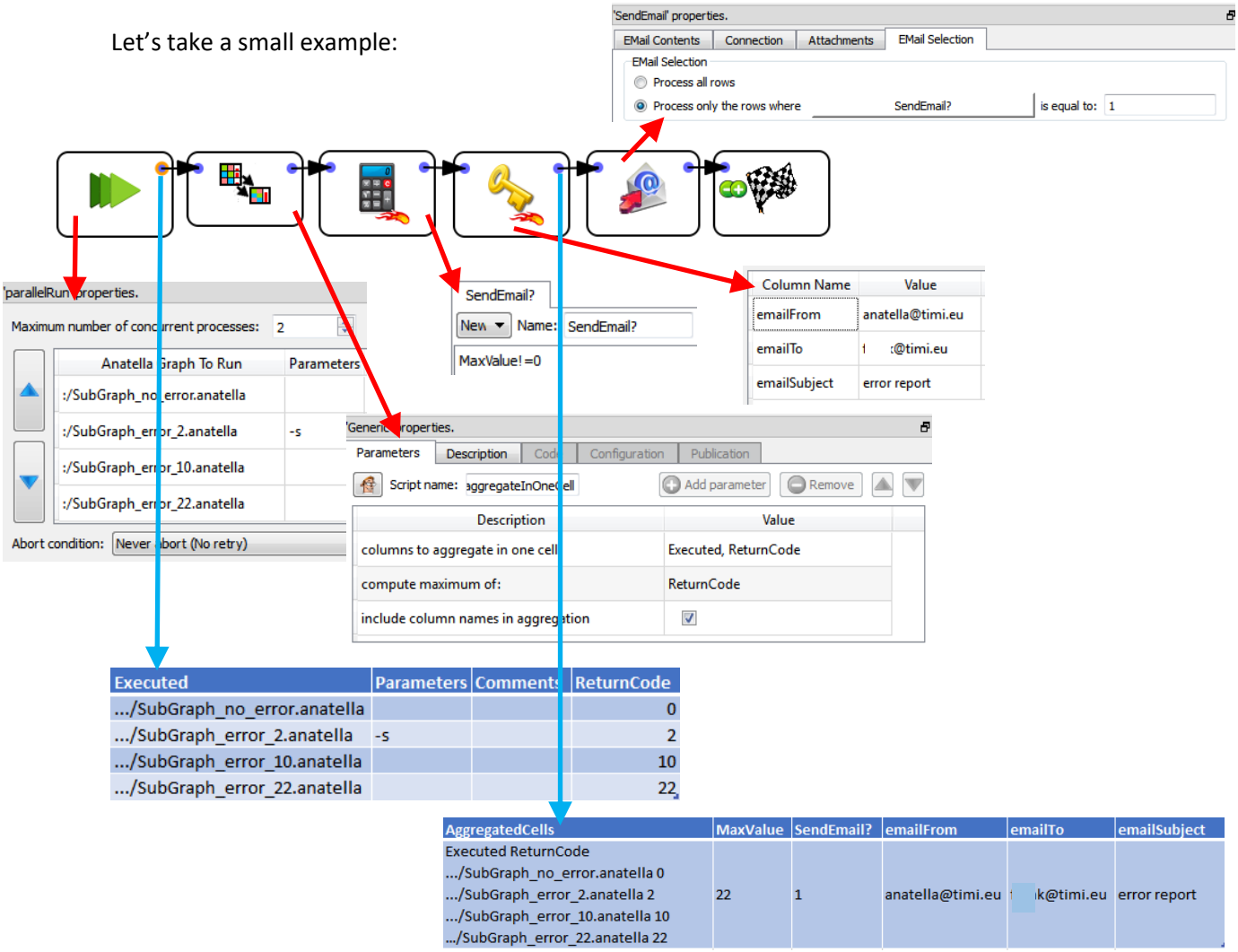
a) It allows you to better organize your work.
b) It allows easier collaboration between a team of different data miners (each one working on a different set of Anatella graphs).
c) It's an easy way to create synchronization-barriers: See section "5.3.2.6 Thread Synchronization".
d) It's an easy way to run the same data transformation inside a "loop" (with some "loop-variable" passed as parameter of the graph).
e) It's an easy way to monitor the success (or the failure) of each of your Anatella graphs (e.g. looking at the output table of the ParallelRun Action, you can automatically send an email to a site administrator, in case of failure).
f) It allows you to use more than 2GB RAM on a 32-bit OS (on a 32-bit OS, each different process can use maximum 2GB RAM, but you can have several processes running at the same time).
g) It's an easy way to reduce computation time, running in parallel several independent Anatella graphs.

If you want to use in parallel the many CPU's available on your computer, you can use the

Multithread Action **or** the ParallelRun Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's take a small example:



The above example graph will send an email to the system administrator if an error (or a warning) is detected during the execution of one of the Anatella-Graphs launched by the ParallelRun Action.

How does this work? The ouput of the ParallelRun Action is a table that contains the final "error level" returned after the execution of each of the Anatella-Graphs (See section 4.7. to know more about "error levels"). We look at all the "error levels" returned by the ParallelRun Action and we send an email if there exists a non-null "error level" (because a non-null "error level" signals an error or a warning during the execution of the Anatella-Graph). We could have achieved the same result using only Javascript: i.e. using the ProcessRunner Javascript Class and the SendMail Javascript Class.

When you run the above example graph, the ParallelRun Action launches the simultaneously execution of **two** Anatella-Data-Transformation-Graphs out of the 4 Anatella-Graphs to execute (because the parameter "*maximum number of concurrent processes*" is **two**). As soon as one of the 2 running Anatella-Graph are finished, Anatella directly launches the next Anatella-Graph, so that there

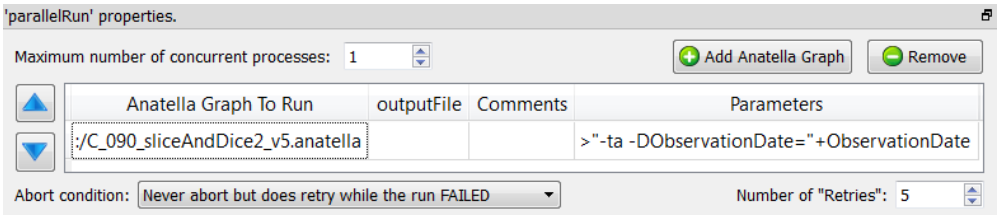
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

are always at least two Anatella-Graphs running at the same time. Internally, to launch the execution of the first graph, Anatella will run (you can see that line in the log-file window of Anatella):

```
Anatella.exe -e  sub_graph_no_error.anatella
```

When the parameter "*maximum number of concurrent processes*" is **one**, the Anatella graphs are executed sequentially in the exact order given by the user. When the parameter "*maximum number of concurrent processes*" is **greater than one**, the order in which the graphs are executed is not fixed (i.e. it's partially random).

Inside the "Parameters" field of the ParallelRun Action, you can pass some "initialization" parameters to the Anatella graph that will be launched. These "initialization" parameters are typically "Global Parameters": see section 4.7.1. to know how to define "Global Parameters" on the command-line (and also section 9.4.2.). If the "Parameters" field starts with a ">" character, then it contains a javascript program that computes a string that is used to define the "Parameters". For example:



This will run the Anatella graph "C_90_sliceAndDice2_v5.anatella" with the 2 command-line parameters: "-ta" (that creates a trace files) and "-DObservationDate=…" (that re-defines the value of the Graph-Global-Parameter "ObservationDate" that is used inside the "C_90_sliceAndDice2_v5.anatella" graph). Note that to compute the value of the "ObservationDate" parameter we used the value of the current "ObservationDate" Graph-Global-Parameter: **We are, in fact, "propagating" the value of the "ObservationDate" Graph-Global-Parameter to the child process** (i.e. to the "C_90_sliceAndDice2_v5.anatella" graph).

When you click the STOP button, Anatella automatically aborts all the child processes that were running (i.e. You don't need to "manually" abort each of the possibly many child processes currently running). For example, when you click the STOP button during execution of the above example graph, it will abort the current "main" graph but also the 4 other graphs: "SubGraph_no_error.anatella", "SubGraph_error_2.anatella", "SubGraph_error_10.anatella", "SubGraph_error_22.anatella".

The ParallelRun Action has also a parameter named "Abort Condition". This parameter can have the following values:

- Never abort (No retry)
- Never abort but does retry while the run FAILED
- Abort if the run still FAILED after some retry
- Abort if the run had still a WARNING after some retry

To detect a FAILURE or a WARNING during a graph execution, Anatella uses the "error level" of the process: See section 4.7. to know more about "error levels".

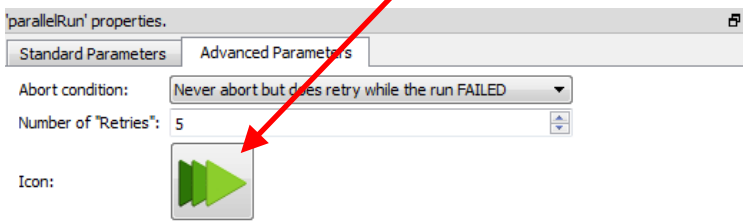
The typical usage of the option "*Abort if the run still FAILED after some retry*" is to to execute a FTP file transfer (e.g. using curl). Such type of task can easily fail. Failure is detected and Anatella re-attempt a

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

new FTP file transfer (the maximum number of attempt is specified using the parameter “number of retries”).

Let’s now assume that you are re-using all the time the same (sub)Graph.
In such situation, you want to:

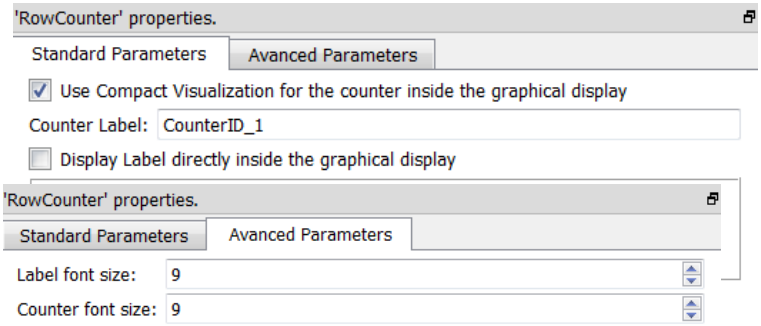
1. …place this (sub)graph inside a specific location on your hard drive (i.e. your graph library), to be sure to always be able to access it.
2. ...associate with this graph a specific icon, so that you can directly visualize, inside your data-transformation-graph, the call to your specific (sub)Graph. To associate an icon to the execution of a specific (sub)Graph, click here: … and select a .png file.



5.3.4. Row Counter

Icon:

Property window:



Short description:
Show a Row Counter inside the graphical display.

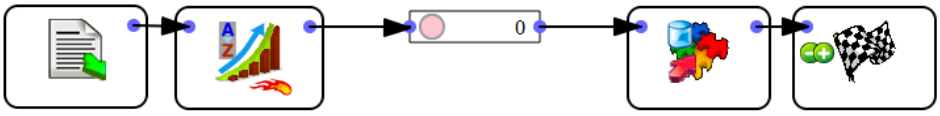
Long Description:

In addition to the counter visible inside graphical display, the RowCounter Action can produce an execution report that appears inside the Log Window. For example:

```
Counter "All Rows": Final State: Finished successfully.
  Number of Row processed: 73.2 M rows (Exactly 73 216 410 rows)
  Execution time: 17.4 seconds (99.914 % of Total execution time)
  Average Row Speed: 251 646 021.65 rows/minute (4 194 100.36 rows/second)
  First Run Time: 22:43:21 2013/8/23 (Started after 0.085852 % of Total execution time)
  Last Run Time: 22:43:38 2013/8/23 (Ended at 100 % of Total execution time)
```

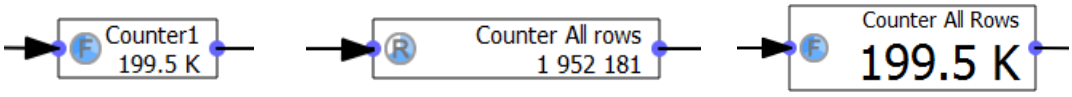
You can save these reports inside a trace file (i.e. a log file) for later consultation/reviewing: see section 4.7.4 to know how to automatically generate trace files at each run.

The RowCounter Action possesses a unique design that is very easy to “spot”. This allows you to directly find “in the blink of an eye” all the counters included inside an Anatella-Data-Transformation-Graph. Here is a very small example:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can change easily the look&feel of the Row Counters inside the graphical display.
Here are some examples:



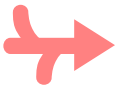
Here are the different meaning of the small icon on the left-side of the Row Counter:

- :Not Executed
- :Waiting for first row
- :Running
- :Finished successfully

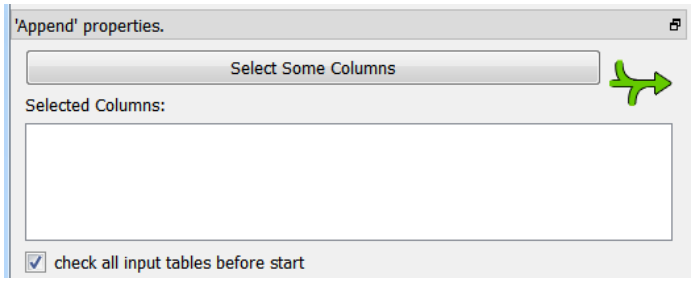
5.4. TA – Join Tables (TA=Transformations Actions)

5.4.1. Append (High-Speed action)

Icon:



Property window:



Short description:
union of several tables.

Long Description:
This operator appends several tables into one.
The tables are placed “below” each other.
Only the columns specified as parameter are written out.

For example: if we have:

- The selected columns: 'Field1', 'Field3'
- These input tables:

```
+--------------------+
|       TABLE 1      |
+------+------+------+
|Field1|Field2|Field3|
+------+------+------+
|     A|     B|     C|
|    AA|    BB|    CC|
|   AAA|   BBB|   CCC|
|  AAAA|  BBBB|  CCCC|
| AAAAA| BBBBB| CCCCC|
+------+------+------+
```

```
+--------------------+
|       TABLE 2      |
+------+------+------+
|Field3|Field4|Field1|
+------+------+------+
|     D|     E|     F|
|    DD|    EE|    FF|
+------+------+------+
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

...we obtain:

```
+-------------+
|OUTPUT TABLE |
+------+------+
|Field1|Field3|
+------+------+
|     A|     C|
|    AA|    CC|
|   AAA|   CCC|
|  AAAA|  CCCC|
| AAAAA| CCCCC|
|     F|     D|
|    FF|    DD|
+------+------+
```

NOTE:
The columns of the different input tables can be in different orders.
The meta-type of each of the selected columns must be the same inside all input tables.
The sort order of the input tables is lost.

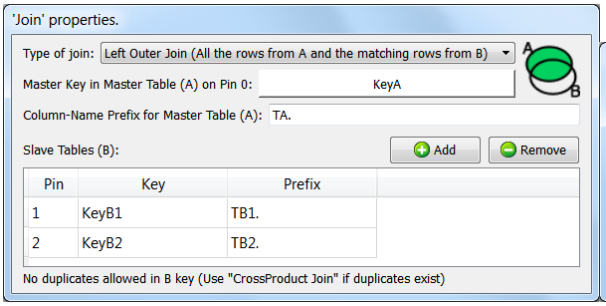
NOTE:

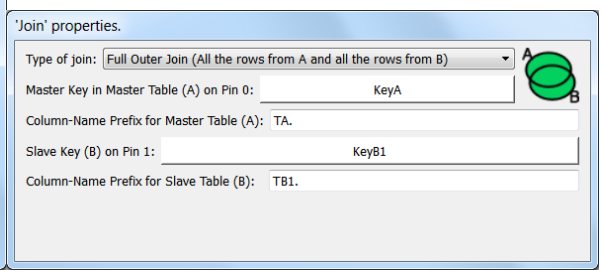
If each of the input table is sorted in the same way, you can use the MergeSort Action (instead of the Append) to obtain a sorted table as output.

5.4.2. Join (High-Speed action)

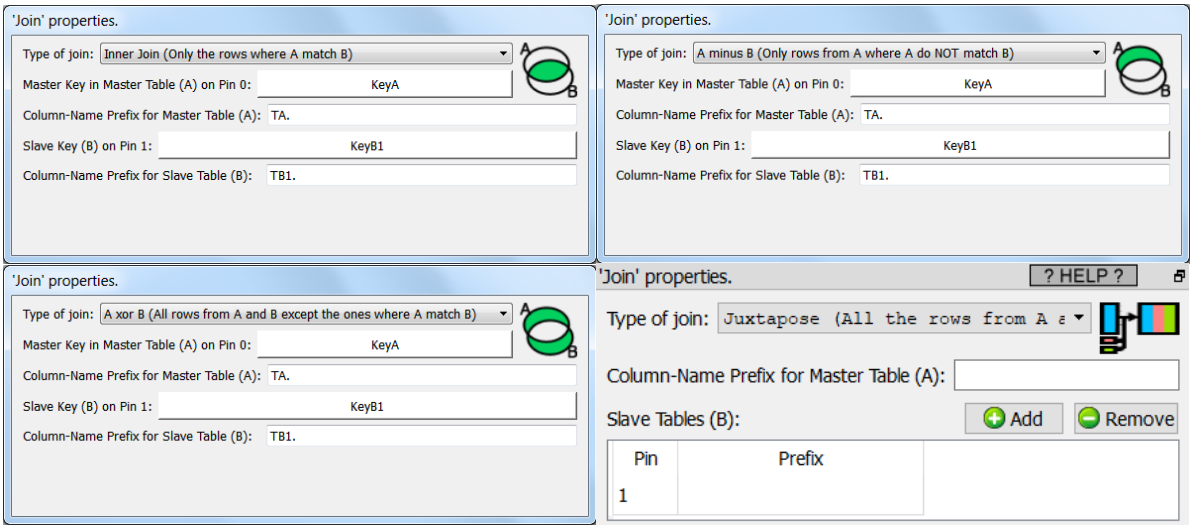
Icon:

Property window:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Short description:
Join several tables.
All the joins computed by the Join Action are using the same unique KEY column inside the MASTER table (Use the MultiJoin action if you have many different keys inside the MASTER table).

Long Description:

Pre-requisite
All the input tables must be sorted on the "Key" columns using the SAME sorting algorithm (all "numeric sort" or all "alpha-numeric sort").

This Action joins several tables on a key. Some definition:

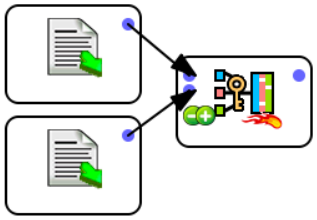
- The first input pin contains the MASTER table.
- The other input pins (second, third, etc.) contain the SLAVE tables.

During the join, the SLAVE tables are added to (i.e. “joined with”) the MASTER table.

Please refer to the section “*5.1.8. Composite Primary Keys*”, to know how to handle composite primary keys.

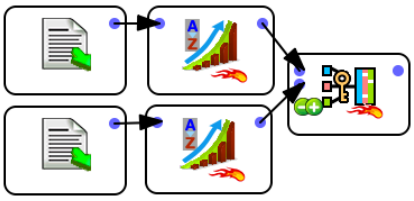
The Join Action checks that all the input tables are sorted in the same way (i.e. using the same sorting algorithm). Sorting all the input table is required for the Join Action to work (in opposition to the MultiJoin Action that does not require any sort at all).

Because the input tables of the Join Action MUST be sorted for the join to work, this produces an error:



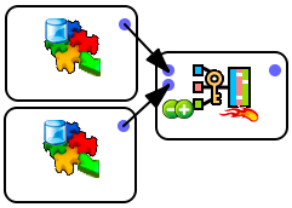
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

..but this will work (provided that the two Sort Actions are properly parameterized):



When you use the "sort action", it modifies the meta-data of the table, so that it now includes the information "*this table is now sorted in a specific way*". Using this meta-data, the "Join operator" is able to check if the input tables are correctly sorted.

This will work if the two .gel_anatella files are properly sorted:



This means that, if you intend to do several join with a specific table, it's better to SORT and thereafter save the table. In this way, you can easily compute many joins because the table is already sorted in the .gel_anatella file (and, thus, you don't need to sort it, again and again, anymore).

All the input tables used inside a Join Action are read simultaneously (in opposition to the MultiJoin Action that, first, reads completely all the SLAVE tables and then, once it's finished, starts reading the MASTER table).

The RAM Memory consumption of the Join Action is negligible (i.e. very small) (in opposition to the MultiJoin Action that, stores completely all the SLAVE tables in RAM).

NOTE:
You can only define one key inside the MASTER TABLE. If you want to define several different keys inside the MASTER TABLE, use the MultiJoin Action.

Let's give some examples: Here are the input tables:

MASTER TABLE(A)

KeyA	Field1	field2
1	A	B
2	AA	BB
3	AAA	BBB
3	AAAA	BBBB
5	AAAAA	BBBBB

SLAVE TABLE 1(B1)

KeyB1	Field1	field2
1	C	D
3	CC	DD

SLAVE TABLE 2(B2)

KeyB2	Field1	field2
1	E	F
2	EE	FF
4	EEE	FFF

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For the “Left Outer Join” (represented by the or icon), we obtain:

OUTPUT TABLE						
KeyM	Field1	field2	T01.Field1	T01.field2	T02.Field1	T02.field2
1	A	B	C	D	E	F
2	AA	BB			EE	FF
3	AAA	BBB	CC	DD		
3	AAAA	BBBB	CC	DD		
5	AAAAA	BBBBB				

For the “Full Outer Join” (represented by the icon) between Table A and Table B2, we obtain:

OUTPUT TABLE				
KeyM	Field1	field2	T02.Field1	T02.field2
1	A	B	E	F
2	AA	BB	EE	FF
3	AAA	BBB		
3	AAAA	BBBB		
4			EEE	FFF
5	AAAAA	BBBBB		

When computing a “Full Outer Join”, some fields from the table A might end up as NULL.

For the “Inner Join” (represented by the icon) between Table A and Table B1, we obtain:

OUTPUT TABLE				
KeyM	Field1	field2	T01.Field1	T01.field2
1	A	B	C	D
3	AAA	BBB	CC	DD
3	AAAA	BBBB	CC	DD

For the “A minus B Join” (represented by the icon) between Table A and Table B2, we obtain:

OUTPUT TABLE		
KeyM	Field1	field2
3	AAA	BBB
3	AAAA	BBBB
5	AAAAA	BBBBB

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For the “A xor B Join” (represented by the icon) between Table A and Table B2, we obtain:

```
+-----------------------------------------+
|              OUTPUT TABLE               |
+----+------+------+----------+----------+
|KeyM|Field1|field2|T02.Field1|T02.field2|
+----+------+------+----------+----------+
|   3|   AAA|   BBB|          |          |
|   3|  AAAA|  BBBB|          |          |
|   4|      |      |       EEE|       FFF|
|   5| AAAAA| BBBBB|          |          |
+----+------+------+----------+----------+
```

When computing a “A xor B Join”, many fields from the table A end up as NULL.

NOTE:

For the join, no row of the MASTER TABLE will ever be duplicated.

NOTE:

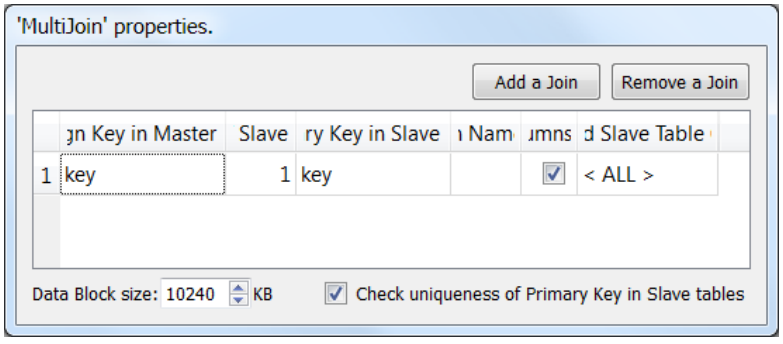
This Action is actually a good example of meta-data management: the "Join Action" looks at the "meta-data" of the input tables and refuses to join the input tables if the meta-data of the input tables says that "*these table are not sorted properly*".

This operator is an example of clever meta-data management. Most "Anatella-Action" are able to work without using any meta-data at all about the columns of the different tables (i.e. usually the actions don't need to know, for example, if the columns contain numbers or characters). ...but this does NOT mean that Anatella is not able to manipulate and manage meta-data information. On the contrary! For example: when the "Join operator" wants to test the meta-data of the input tables (to see if these input tables are properly sorted), the Anatella framework is providing ALL the required functionalities to do so. Most of the time, with Anatella, you don't need to do any meta-data management but when you really need to do it, it's really easy and powerful.

5.4.3. Multi Join (High-Speed action)

Icon:

Property window:



Short description:
Join several tables. You can have many different KEY columns inside the MASTER table.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:

This Action joins several tables based on different keys. Some definition:

- The first input pin contains the MASTER table.
- The other input pins (second, third, etc.) contain the SLAVE tables.

During the join, the SLAVE tables are added to (i.e. "joined with") the MASTER table.

The MultiJoin Action only computes "Left Outer Joins".

Please refer to the section "*5.1.8. Composite Primary Keys*", to know how to handle composite primary keys.

It is assumed that the column used as key inside the SLAVE tables have no duplicates (i.e. it can be used as primary key). Unexpected results may occur if that's not the case. Anatella checks for the uniqueness of the keys in the SLAVE tables. This check might take some time and you can thus de-activate it but it's strongly not recommended. If the check fails (i.e. if there are some duplicate keys inside one of the slave tables), you can investigate this data quality issue using the NaïveDeDuplicate Action.

The MultiJoin Action first loads all the SLAVE tables into RAM Memory and then, once the loading is finished, it starts processing the MASTER table row-by-row. Thus, we have the following:

- The MASTER table is processed row-by-row and can thus have any size (i.e. it can be as big as you want without increasing RAM memory consumption).
- The RAM Memory consumption of the MultiJoin Action is proportional to the size of the SLAVE tables. If your SLAVE tables are large, you'll have a large RAM Memory consumption. There are three ways to reduce RAM memory consumption (to be able to handle bigger SLAVE tables):
  - If the same SLAVE table S is used several times in different joins, connect the SLAVE table S **<u>one time to only one of the input pin</u>** of the MultiJoin Action.

    For example: We want to know the longitude & latitude of the two individuals A and B, each time they call each other. We have two tables:

TABLE 1: PHONE CALLS

PHONE_NUMBER_A	PHONE_NUMBER_B	ANTENNA_A	ANTENNA_B
555-0103	555-0153	1	3
555-2106	555-2368	2	1
555-6666	555-9999	1	3
555-2840	555-4782	4	2
555-6102	555-8258	1	2
...	...	...	...

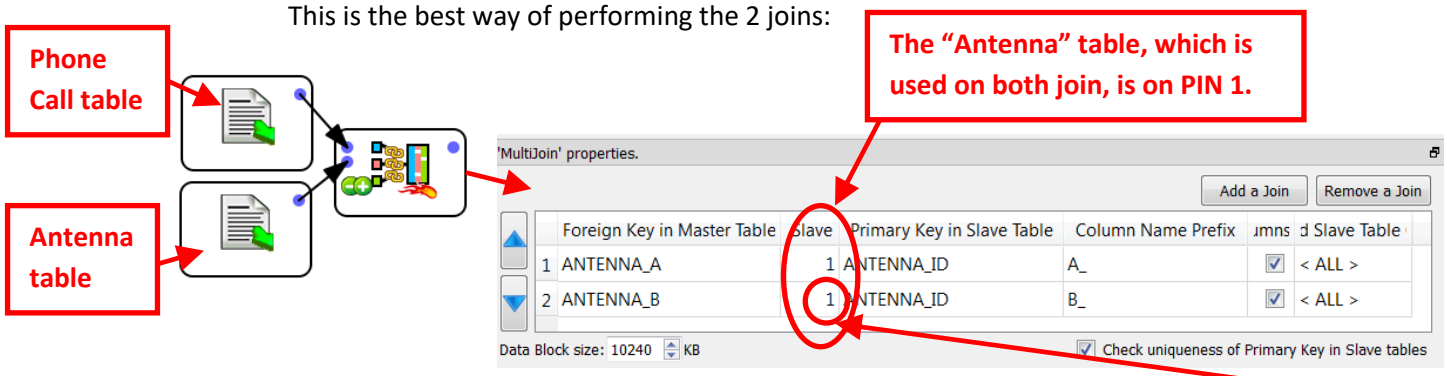
One row= one phone call between A and B using the 2 specified GSM antennas.

TABLE 2: ANTENNAS

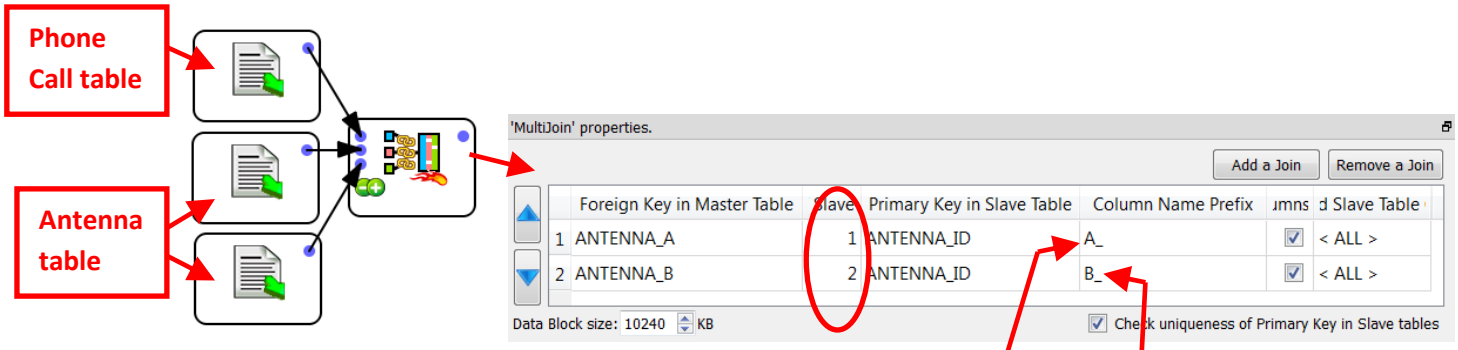
ANTENNA_ID	LONGITUDE	LATITUDE
1	50.615904	3.740324
2	50.826711	4.378888
3	48.786606	2.220418
...	...	...

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This is the best way of performing the 2 joins:



Do not use the following settings because it doubles the RAM memory consumption (although it gives the correct answer):



NOTE:

Each row of the above table (i.e. the table inside the MultiJoin properties window) is defining one join between two tables.

The number of rows inside the above table can thus be completely different from the number of tables used to compute the output of the MultiJoin Action:

e.g. It can happen that you compute 5 different joins using only 2 tables. In such situation, the above table will have 5 rows and the number of input pin is only 2.

The SLAVE table that is used inside a particular join is specified using its input **pin number**. In the example, the second join is using the SLAVE table on pin 1.

The correct answer for the join operation is the following table:

PHONE_NUMBER_A	PHONE_NUMBER_B	ANTENNA_A	ANTENNA_B	A_LONGITUDE	A_LATITUDE	B_LONGITUDE	B_LATITUDE
555-0103	555-0153	1	3	50.615904	3.740324	48.786606	2.220418
555-2106	555-2368	2	1	50.826711	4.378888	50.615904	3.740324
555-6666	555-9999	1	3	50.615904	3.740324	48.786606	2.220418
555-2840	555-4782	4	2	49.123456	3.123456	50.826711	4.378888
555-6102	555-8258	1	2	50.615904	3.740324	50.826711	4.378888
...	...	...	...	...	...	...	...

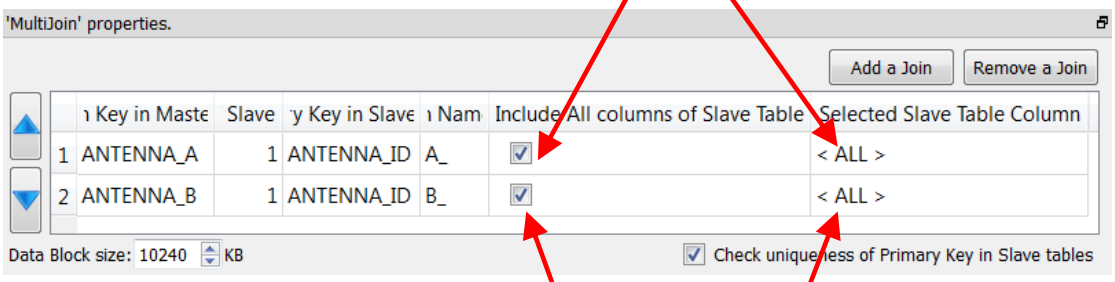
From the First Join (All the column names have the "A_" prefix)

From the Second Join (All the column names have the "B_" prefix)

- Use a ... he data in ... 5.1.2 about data-types). To remind you: The most memory-efficient data type is "Key" (4 bytes per cell) and after "Float" (8 bytes per cell).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- You should only select, inside the slave tables, the columns required inside the output table of the MultiJoin Action because only these columns will be loaded into RAM memory (thus reducing the memory required to store the SLAVE table). By default, **ALL** the columns of the SLAVE tables are loaded into RAM memory (this is thus very bad from the RAM memory consumption point-of-view).



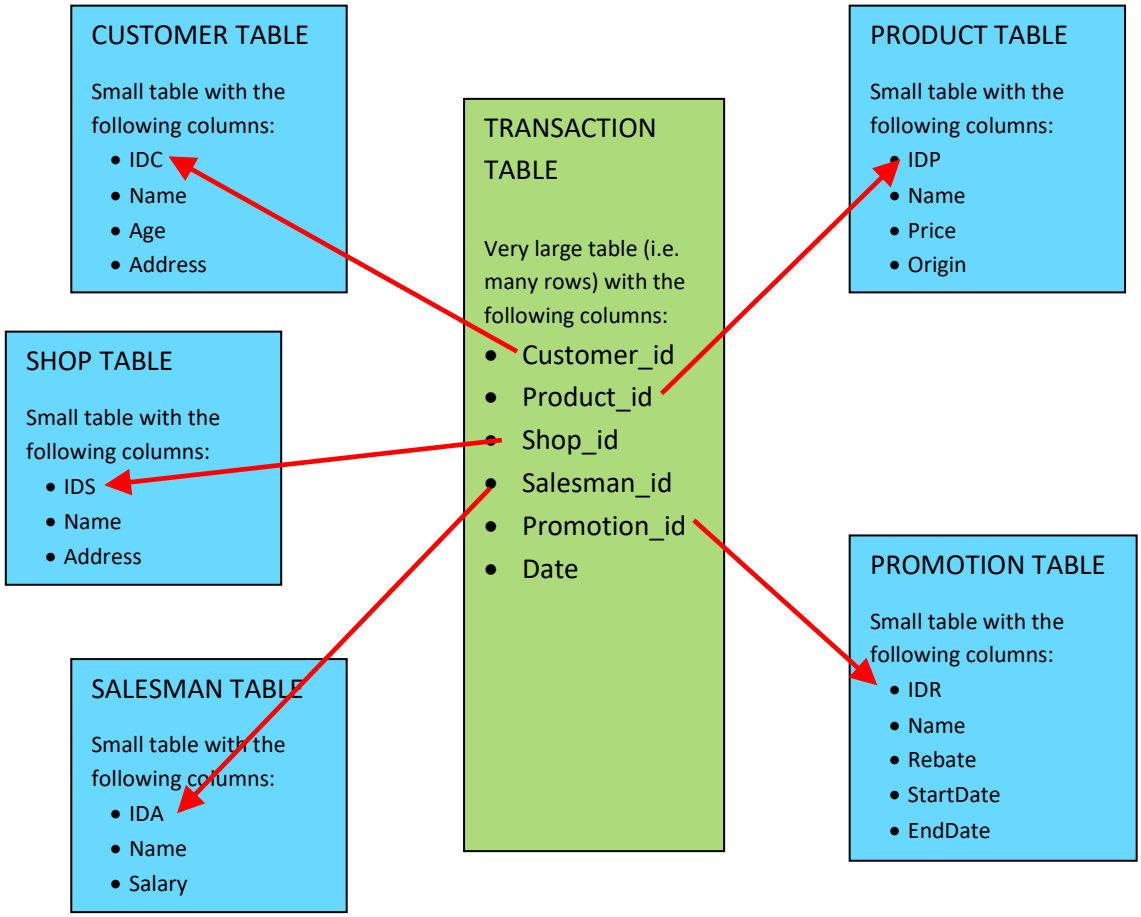
To select some particular columns from the SLAVE tables:

1. untick the checkboxes here:
2. click here to select some particular column:

Despite the above optimizations, the MultiJoin Action might still consume a very large amount of RAM memory: Please also refer to section "*5.3.2.7. Main RAM Memory Consumption*" to know more about this subject.

Typically, the MultiJoin Action is used to de-normalize (i.e. put everything inside a single table) the databases that are in "Star Schema". For example, we have:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



It should be obvious, from the look of the above schema, to guess why this schema is named “Star” schema. Inside the database literature, you’ll find that:

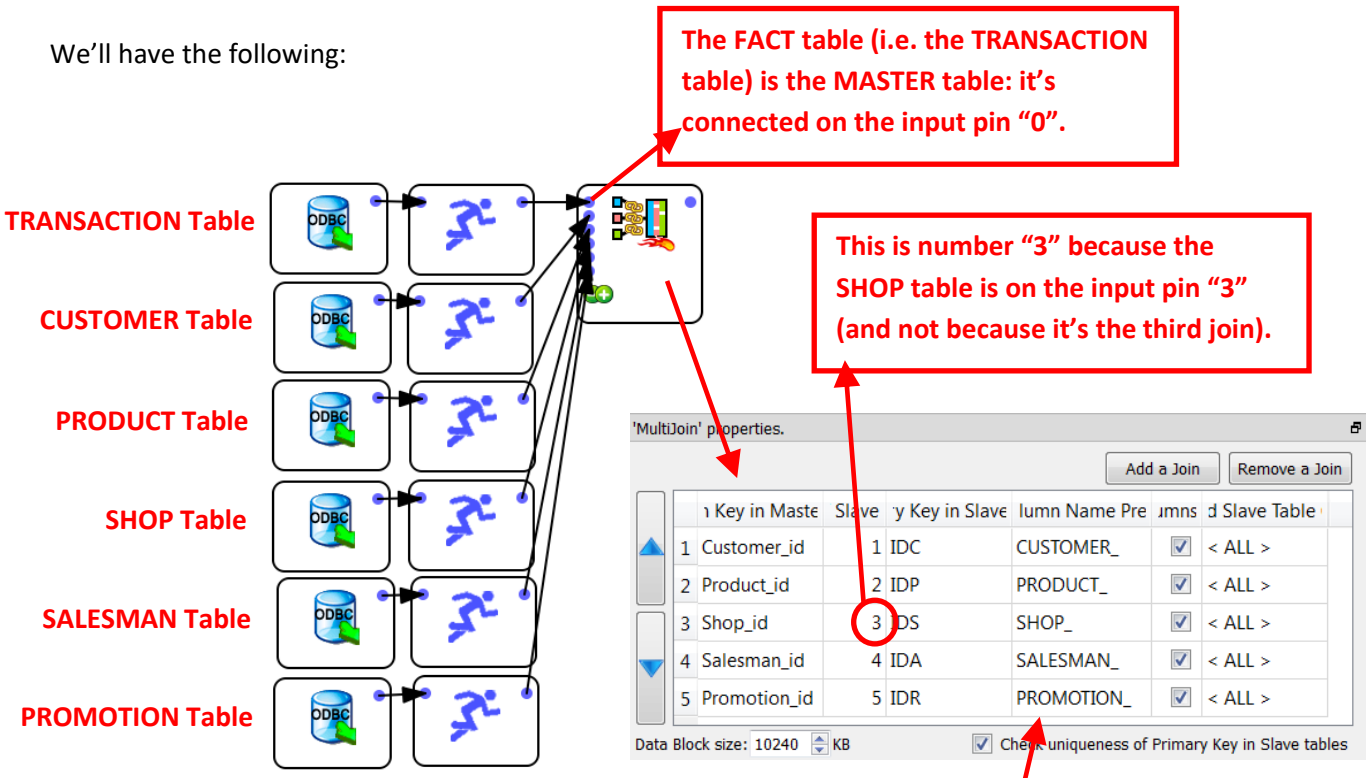
- The large table “in the center” of the schema (i.e. the table named TRANSACTION, in green) is referred as the “FACT” table in the literrature.
- The small tables “on the border” of the schema (i.e. the blue tables) are referred as the “DIMENSION” tables in the litterature.

To de-normalize “Star Schema” databases, you will:

- set as MASTER table of the MultiJoin Action the FACT table (in the example above: the TRANSACTION table).
- set as SLAVE tables of the MultiJoin Action all the DIMENSION tables (in the example above: the blue tables).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We'll have the following:



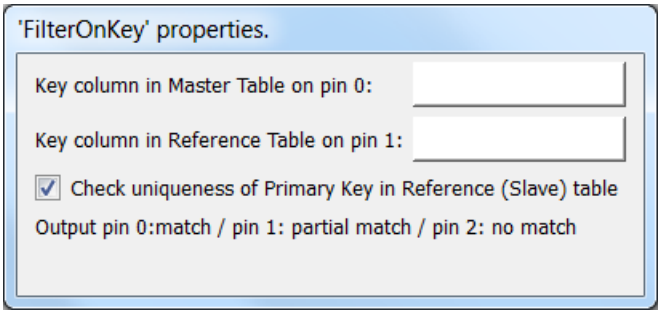
Please note that we used different column-name-prefixes for each join:

Using column-name-prefixes is important to avoid any "collision". In the above example, the output table from the MultiJoin Action contains the columns: *CUSTOMER_Name, PRODUCT_Name, SHOP_Name, SALESMAN_Name, PROMOTION_Name*. If we forgot to set any prefixes (Warning: This is the default Anatella behavior!), all these different columns ends up with exactly the same name (that is "*Name*") and we have many "column name collisions". Collisions are detected at runtime inside the CSVFileWriter Action and the GenericODBCWriter Action.

5.4.4. Filter On Key (High-Speed action)

Icon:

Property window:



Short description:
Filter Rows based on a set of "Reference" Keys.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

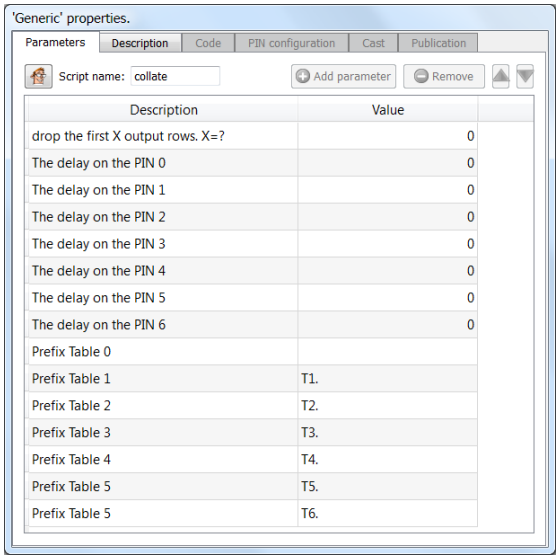
Long Description:
Filter Rows based on a set of “Reference” Keys.

It is assumed that the column used as key inside the SLAVE tables have no duplicates (i.e. it can be used as primary key). Unexpected results may occur if that’s not the case. Anatella checks for the uniqueness of the keys in the SLAVE tables. This check might take some time and you can thus de-activate it but it’s strongly not recommended. If the check fails (i.e. if there are some duplicate keys inside one of the slave tables), you can investigate this data quality issue using the NaïveDeDuplicate Action.

5.4.5. Collate

Icon:

Property window:



Short description:
Collate several tables.

Long Description:
The different tables will be glued together side-by-side.
Some empty rows can optionally be inserted at the beginning of the tables.
This Action is handy when manipulating time-series.
This Action is very similar to a simple Join Action (but it does not require any keys).

For example, if we have:

- The "delays" (that are defining the number of empty rows at the beginning of the table):
  - PIN 1= 1 delay
  - PIN 2= 3 delay
  - PIN 3= 0 delay
- These input tables:

```
+-------------+   +-------------+   +-------------+
|   TABLE 1   |   |   TABLE 2   |   |   TABLE 3   |
+------+------+   +------+------+   +------+------+
|Field1|field2|   |Field1|field2|   |Field1|field2|
+------+------+   +------+------+   +------+------+
|     A|     B|   |     C|     D|   |     E|     F|
|    AA|    BB|   |    CC|    DD|   |    EE|    FF|
|   AAA|   BBB|   +------+------+   |   EEE|   FFF|
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
|   AAAA|   BBBB|                          |   EEEE|   FFFF|
+-------+-------+                          | EEEEE| FFFFF|
                                           |EEEEEE|FFFFFF|
                                           +------+------+
```

...we obtain:

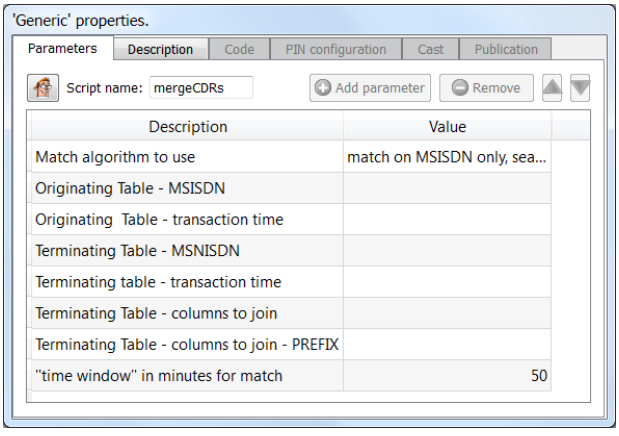
```
+-------------------------------------------------------------+
|                        OUTPUT TABLE                         |
+------+------+---------+---------+---------+---------+
|Field1|field2|T1.Field1|T1.field2|T2.Field1|T2.field2|
+------+------+---------+---------+---------+---------+
|      |      |         |         |        E|        F|
|     A|     B|         |         |       EE|       FF|
|    AA|    BB|         |         |      EEE|      FFF|
|   AAA|   BBB|        C|        D|     EEEE|     FFFF|
|  AAAA|  BBBB|       CC|       DD|    EEEEE|    FFFFF|
|      |      |         |         |   EEEEEE|   FFFFFF|
+------+------+---------+---------+---------+---------+
```

NOTE:
You don't need any primary key at all.
The rows are placed in the order in which they "arrive".

5.4.6. Merge CDRs

Icon:

Property window:



Short description:
Perform a "fuzzy" join between the 2 input tables.

Long Description:
A typical Mobile Money transaction goes, basically, the following way:

- a subscriber A wants to transfer some money to another subscriber B.
- the subscriber A sends a SMS to the central SMS server.
  This SMS is crypted and contains:
  a) the amount to transfer.
  b) the MSISDN from B.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- the whole transaction is logged inside 2 different tables:
  a) Each row of the "TABLE A" describes one money transfer between 2 subscribers identified by their MSISDN number. The columns are MSISDN_A, MSISDN_B, TRANSACTION_TIME, TRANSACTION_AMOUNT.
  b) Each row of the "TABLE B" describes one SMS between a subscriber MSISDN_A and the central SMS server. The columns are MSISDN, TIME, ANTENNA_ID.

We want to find, for each transaction of TABLE A, the ANTENNA_ID from the subscriber that started the transaction. In other word, we need to compute a join between the TABLE A and the TABLE B.

We could try to compute the join using the standard Join Action. More precisely:

2. We use the columns (MSISDN_A, TRANSACTION_TIME) from "TABLE A" as Composite-Primary Key.
3. We use the columns (MSISDN, TIME) from "TABLE B" as Composite-Primary Key.

This approach is described in more detail in section 5.1.6 about Composite primary Keys. This naïve approach will fail because the "TIME" column is not exactly the same inside each of the two tables (because these are two different servers that are generating the two tables A and B and their clock is not tightly synchronized).

To be able to compute the join, we need a "Fuzzy" match on the TIME dimension. Meaning that, if the TRANSATION_TIME and the TIME are separated by less than 3 minutes (this is the "Time Window" parameter), we'll decide that the TIME columns are "matching" (and we'll thus produce a join and retrieve the ANTENNA_ID associated with the transaction).

The parameters of the

MergeCDR Action are:

4. The algorithm used to compute the Fuzzy match.
5. The Originating Table: the Table A: the Transaction Table.
6. The Terminating Table: the Table B: the SMS Table.
7. The Time window (in the example above: 3 minutes).

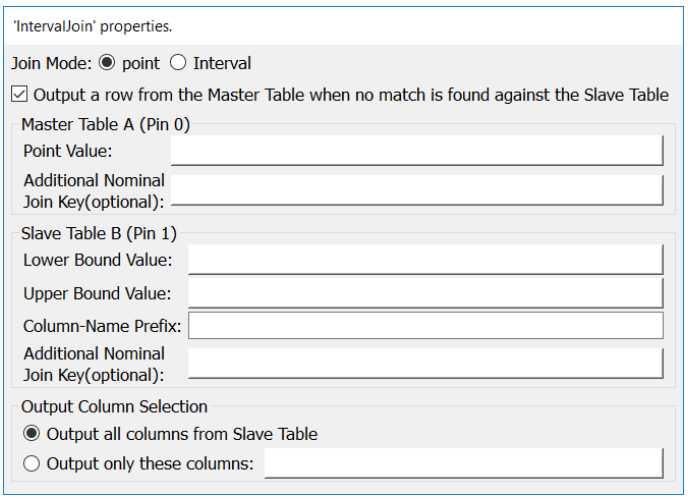
The

MergeCDR Action is written in JavaScript so you can easily tweak/modify it to suit your specific needs.

5.4.7 Interval Join (High-Speed action)

Icon:

Property window:



Short description:
Joins two tables based on an interval

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Change the Data Type of some columns, for example time range, income range, age range, etc.

There are two operating modes: **point mode**, and **interval mode**.

In the **point mode**, the master table contains a single continuous value that will seek an interval match in the slave table.

Time
13:27
14:18
17:41

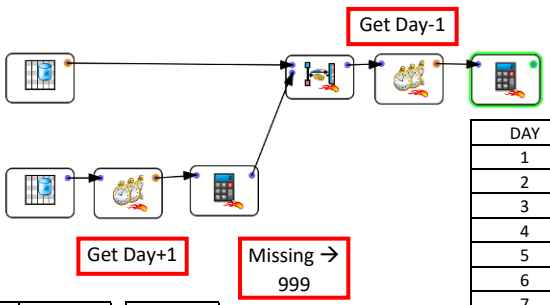
Lower	Higher	Price
13:00	13:29	$175.00
13:30	13:59	$178.00
14:00	14:29	$165.00
14:30	14:59	$162.00
15:00	15:29	$175.00
15:30	15:59	$181.00
16:00	16:29	$182.00
16:30	16:59	$181.00
17:00	17:29	$180.00
17:30	17:59	$179.00
18:00	18:29	$175.00
18:30	18:59	$177.00

For example, let's imagine we need to get a quote price depending on the time of the day, we would look in the "slave" table which interval corresponds to the transaction time. The interval is considered as [LOWER, UPPER[, which means you always need to consider that your upper bound is excluded. The additional nominal key typically identifies a customer or a transaction ID that is present in both tables

A common setting to use the interval join (and probably one of the conceptually most complex transformation necessary in time-based predictive modeling – hint: certification question) is to prepare data for predictive modelling in which a day is a predictive reference, and we need to keep the information of the last purchase in each prediction window, with only a variation of "time passed". For this, we create an artificial table in which there is a possible transaction every day and make some relatively complex computations to account for the "upper bound excluded" constraint.

In this example, the prefix "T." applies to the slave table.

Day
1
2
3
4
5
6
7



DAY	T.DAY	T.Q	T.P	T.DAY_T+1	T.DAY_T-1	Days Passed	TARGET
1	1	2	17	5			1
2	1	2	17	5	1	1	0
3	1	2	17	5	1	2	0
4	1	2	17	5	1	3	0
5	5	3	16	7	1	4	1
6	5	3	16	7	5	1	0
7	7	7	18	999	5	2	1

Day	Quantity	Price
1	2	$ 17.00
5	3	$ 16.00
7	2	$ 18.00

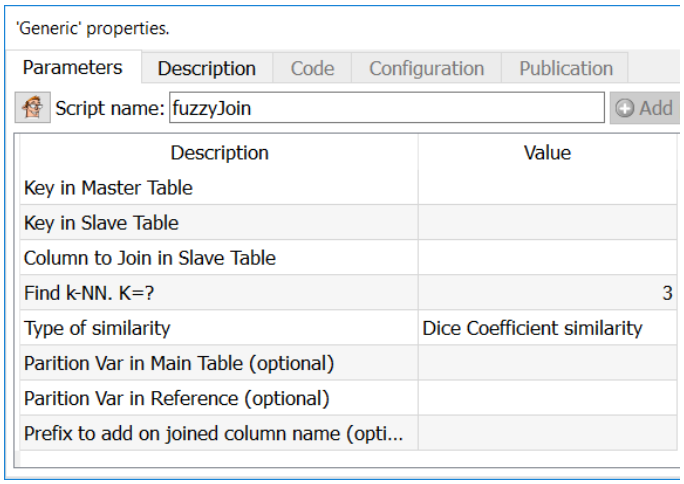
Day +1
5
7

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.4.8 Fuzzy Join (JavaScript action)

Icon:

Property window:



Short description:
Joins two tables based on text approximation

Long Description:

Uses either **Jaro Winkler**, **Damereau Levenstein** or **Dice Coefficient** to establish if two keys are *probably* identical.

Let's imagine we have three tables in which a key has not been defined, and we want to use names to join them.

Name	Company
Daniel Soto Zeevaert	Timi

Name	Region
Daniel Soto Zevart	LatAm

Name	Title
Soto Zeevaert Daniel	Director

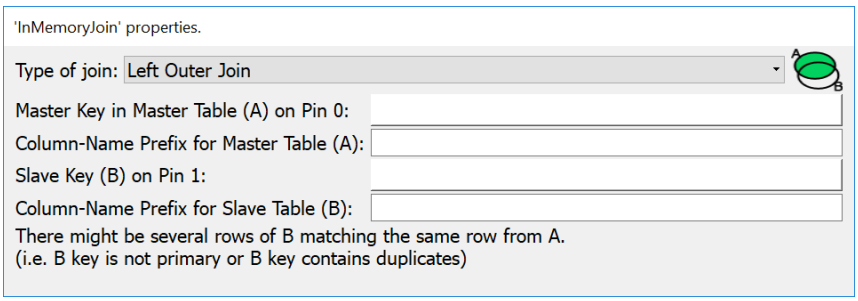
We have:
Three tables contain different information about an employee.
Unfortunately, humans sometimes input data incorrectly.
in this setting, both J-W and D-L will do a good job to recognize the Table 1 and Table 2 are similar but will miss the information of table 3.
Dice Coefficient will work in all situations.

For more information, see 5.9.1 Correct Spelling

5.4.9. In-Memory Join (High-Speed action)

Icon:

Property window:



Short description:
Perform a left-join or inner-join in-memory

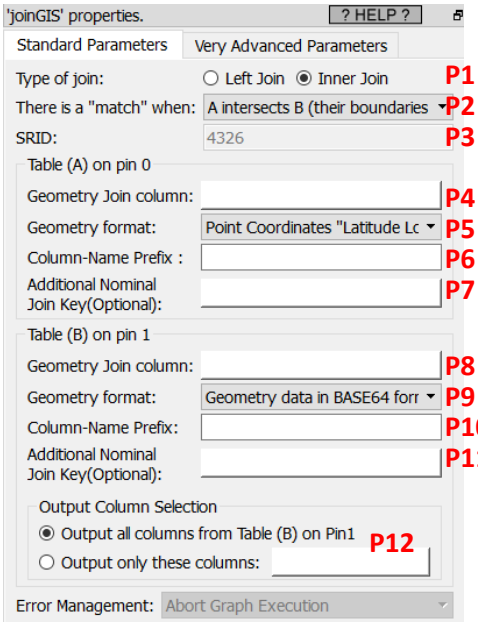
Long Description:
Perform a left-join or inner-join in-memory. This is similar to the "multi-join" in the sense that it is not required to sort the tables before the join, but the slave table may include duplicate keys.

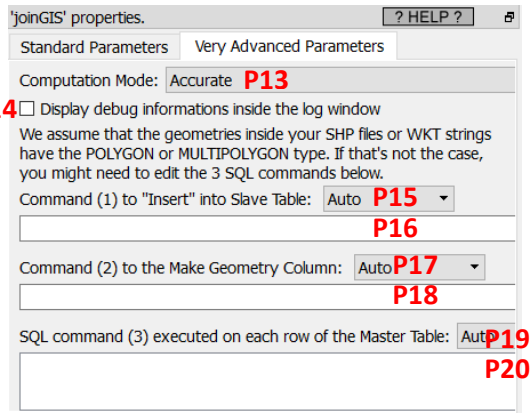
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.4.10. GIS Join (High-Speed action)

Icon:

Property window:





Short description:
Computes a GIS join

Long Description:
Computes a join between two table (“Table A” and “Table B”) using geographical informations.

As usual, the “Table B” is the “Slave” table: It’s completely loaded into memory at the start of the execution, so it should be “small enough” to fit into your RAM memory. To use less RAM, you can load into memory a sub-selection of the columns from “Table B” using the parameter **P12**.

The different options for the parameters **P5** and **P9** are:

- **Point Coordinates "Latitude Longitude".**
  This format is self-explanatory. The Latitude and Longitude must be stored in the same column, separated with a space char, in decimal degree.
- **Rectangle "Latitude1 Longitude1 Latitude2 Longitude2"**
  This format is self-explanatory. The 4 numbers must be stored in the same column, separated with a space char, in decimal degree.
- **Geometry data in WKT format**
  You can create WKT strings very easily using this interactive mouse-based editor:
  https://timi.eu/wkt.html
- **Geometry data in HEX format**
  This format is for compatibility with PostGreGIS.
- **Geometry data in BASE64 format**
  This is the default geometry format that is given as output of the loadShape action (see the section 5.2.31 for more details about this action). Basically, you get a “Geometry” column from your .shp Shape files using the loadShape action and you use it here.

The parameter **P3** defines the SRID that is used for the computation (more details on this subjection in the section 5.2.31). Most of the time, you must use the 4326 SRID. When you compute a join between two .shp Shape files, you can freely choose the SRID (but please make sure that both shape files are using the same SRID).

Behind the scene, this Anatella action uses the very efficient “Spatial Index” included inside the SpatialiteGIS library to speed-up the computations.

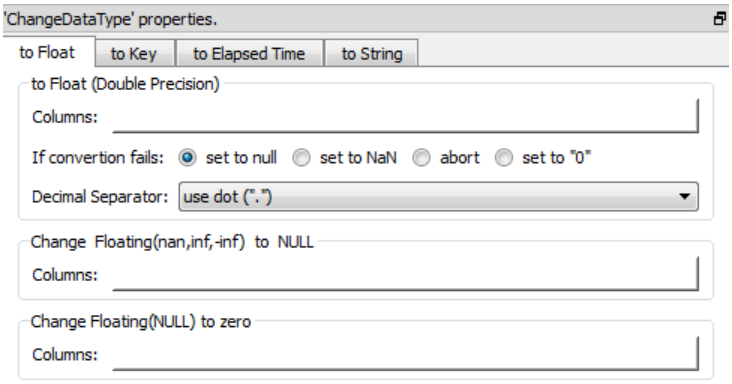
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

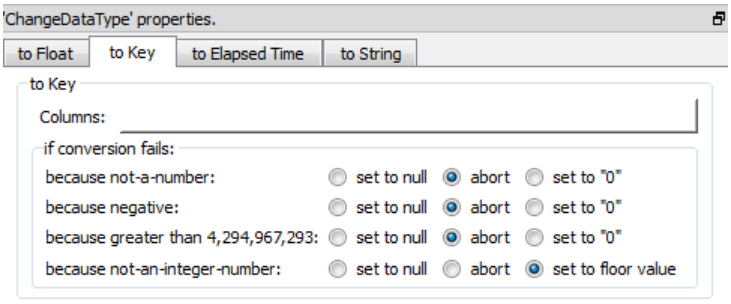
5.5. TA – Standard (TA=Transformations Actions)

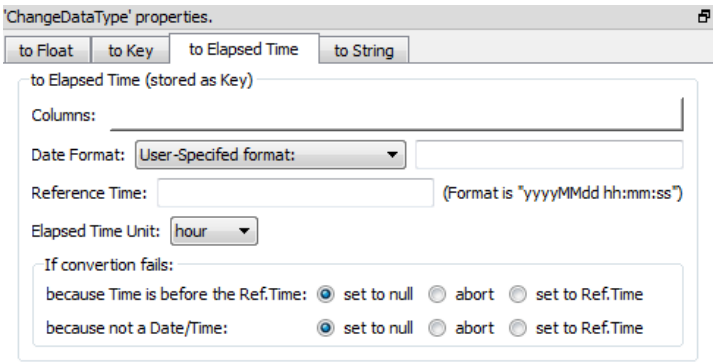
5.5.1. Change Data Type (High-Speed action)

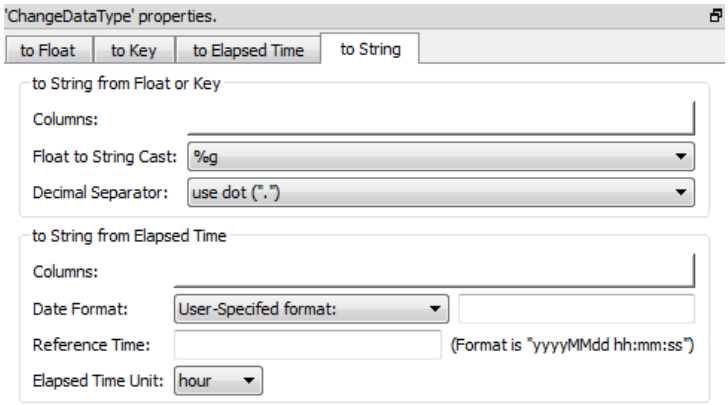
Icon:

Property window:









Short description:
Change the Data Type of some columns

Long Description:
Change the Data Type of some columns

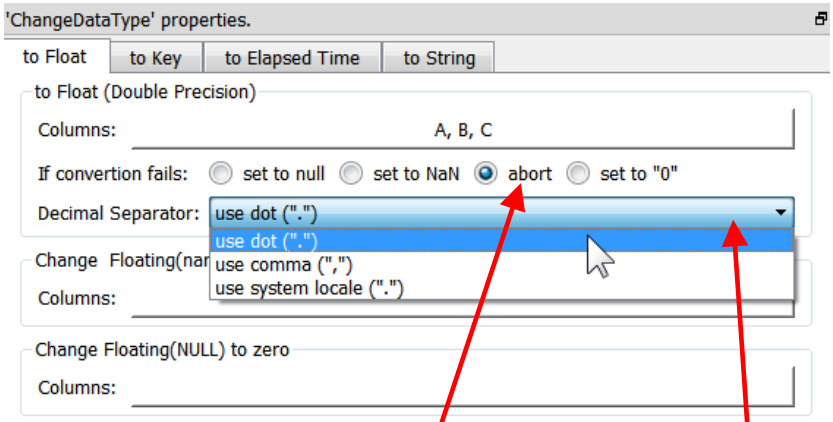
Use the ChangeDataType Action to convert columns from one data-type to another.

5.5.1.1. Converting to (& checking) floating-point numbers

You can also use the ChangeDataType Action to check if a column really contains floating-point numbers. This “sanity check” can potentially detect and prevent difficulties when manipulating tables of doubtful origin.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example, to CHECK & convert the column A, B and C to floating-point numbers, we'll have:

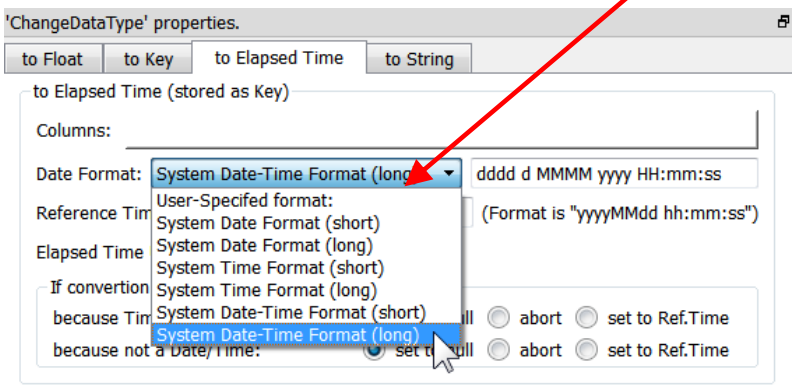


We decided to completely abort the data transformation if some non-numbers are found inside the column A, B and C. For example, if the column A contains "0,5", Anatella will abort the execution of the data transformation graph.

One very common pitfall when importing columns with numbers inside Anatella is that: In some countries the decimal separator is the dot (".") and in some other it's the comma (","). By default, Anatella is using the dot as decimal separator (i.e. Anatella uses the English notation). Inside the ChangeDataType Action, you can change the Decimal Separator here: . The third option (that is named "*use system locale*") is special: When using this option, Anatella uses the decimal separator that is globally defined inside the MSWindows Operating System (inside the "format" tab of the "Region and Language" option found inside the MSWindows "control panel"). This means that Anatella will dynamically change, at run-time, the decimal separator to always use the globally defined decimal separator from MSWindows.

5.5.1.2. Converting to (& checking) dates

The same importation difficulties that appeared for numbers also appear when importing dates and times. Each country has a different way of writing dates (and times). One again, you can use the "*system locale*" settings (more precisely, use one of the six "*System*" options:) to dynamically change, at run-time, the importation settings to reflect the Date & Time settings globally defined inside MSWindows:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.1.3. Converting Numbers to Strings

Inside Anatella, there are basically two different notations to display floating-point numbers:

- Standard notation.
- Scientific notation.
- "Short" notation (that is the shortest of the 2 above notations)

Here are some examples:

	Example 1 (a primary key)	Example 2 (a small number)	Example 3 (a large number)	Example 4 (high precision number)	Example 5 (procent or any number)
Number	123456789	0.0000002	120000000	1-0.9-0.1	5.3
Standard notation limited to 16 chars (%.16f)	123456789	0.0000002	120000000	-0	5.3
Scientific notation limited to 16 chars (%.16e)	1.23456789E+08	2e-007	1.2e+008	-2.775557561562891e-017	5.3e+000
Shortest notation limited to 16 chars (%.16g)	123456789	2e-007	1.2e+008	-2.775557561562891e-017	5.3
Standard notation using 6 chars to represent the fractional part (%f)	123456789.000000	0.000000	120000000.000000	-0.000000	5.300000
Scientific notation limited to 6 chars (%e)	1.234567e+008	2e-007	1.2e+008	-2.77556e-017	5.3e+000
Shortest notation limited to 6 chars (%g)	1.234567e+008	2e-007	1.2e+008	-2.77556e-017	5.3
Standard notation limited to 5 chars (and 3 chars are used to represent the fractional part) (%5.3f)	1.23E+08	0.000	120000000.000	-0.000	5.300

The "shortest" notation is noted **(%g)**. It's based on either the standard **(%f)** or the scientific **(%e)** notation. The "shortest" notation is the more "compact" of the 2 basic notations (standard or scientific). Furthermore, trailing zeros are truncated, and the decimal point appears only if one or more digits follow it.

For the **(%.16g)**: The scientific notation is used only when the exponent of the value is less than –4 or greater than **16**.

For the **(%g)**: The scientific notation is used only when the exponent of the value is less than –4 or greater than **6**.

To select a number notation follow these guidelines:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- The easiest to read and the most compact notation is usually the **(%g)** notation.
  This is the default notation that is used everywhere inside Anatella.
  The **(%g)** notation is used:
  - inside the data preview window.
  - as default notation when exporting your data inside a simple text file.
  - to (automatically) convert a number to a string inside the Calculator Action.
- The **(%g)** notation works very well almost all the time. There however are 2 cases when it's better to use the **(%.16g)** notation rather than the **(%g)** notation:
  - When the number that you want to display is a primary key. In such case, you never want to use the scientific notation (especially for large numbers): See the example 1 here above.
  - When you are exporting your data to another software for further processing, you don't want any "rounding" or "truncation" errors that might jeopardize the next computations: See the example 4 here above.
- When your numbers are "percents", then it's sometime better to use the **(%5.3f)** notation to obtain a nicer and clearer display.

If you want to use other notations inside Anatella, you can manually edit the .anatella files with a text editor and write the requested notation in it.

If you want more information about the different notations (%g,%e,%f) and the different precisions (%.16g or %5.3f), please refer to the following webpages:

- About printf:
  http://msdn.microsoft.com/en-us/library/wc7014hz%28v=vs.90%29.aspx
- About notations (%g,%e,%f):
  http://msdn.microsoft.com/en-us/library/hf4y5e3w%28v=vs.90%29.aspx
- About precision:
  http://msdn.microsoft.com/en-us/library/0ecbz014%28v=vs.90%29.aspx

5.1.1.4. Dynamically changing the Data Type

On the second input pin (pin 1) of the ChangeDataType Action, you can connect a small table with 2 columns:

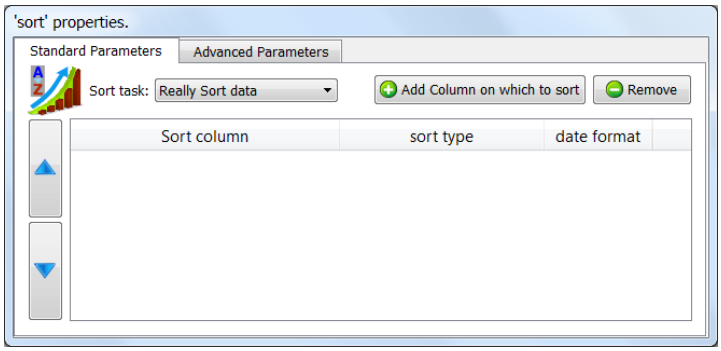
- The first column contains the names of the columns to process on the first input pin
- The second column contains a string that describes the transformation to apply.
  The accepted values are:
  - KEY
  - FLOAT
  - ET
  - STRING
  - STRINGFROMET
  - ZERO or 0
  - NULL

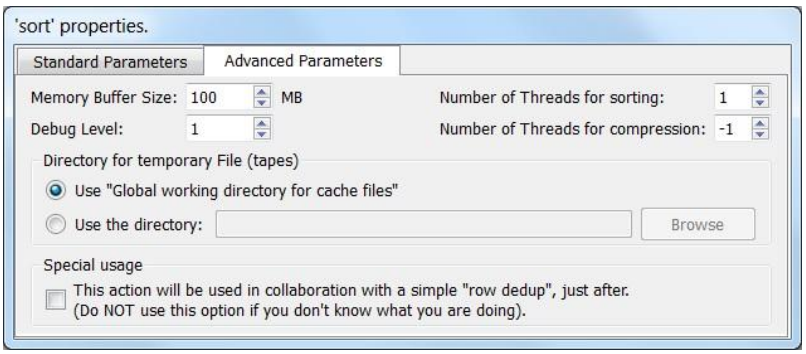
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.2. Sort (High-Speed action)

Icon:

Property window:





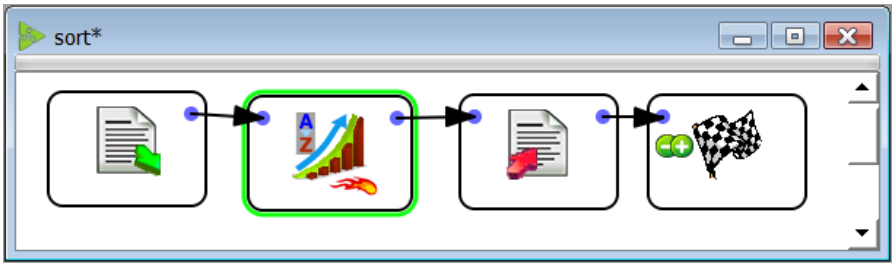
Long Description:

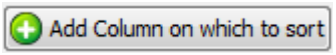
Sorts the table and modifies the meta-data of the table to reflect that the table is now sorted in specific way. The "Sort task" can be:

a) *Really Sort data* (This modifies the meta-data)
b) *Only check sort (with errors)* (This modifies the meta-data)
c) *Only check sort (with warning)* (This does NOT modify the meta-data)

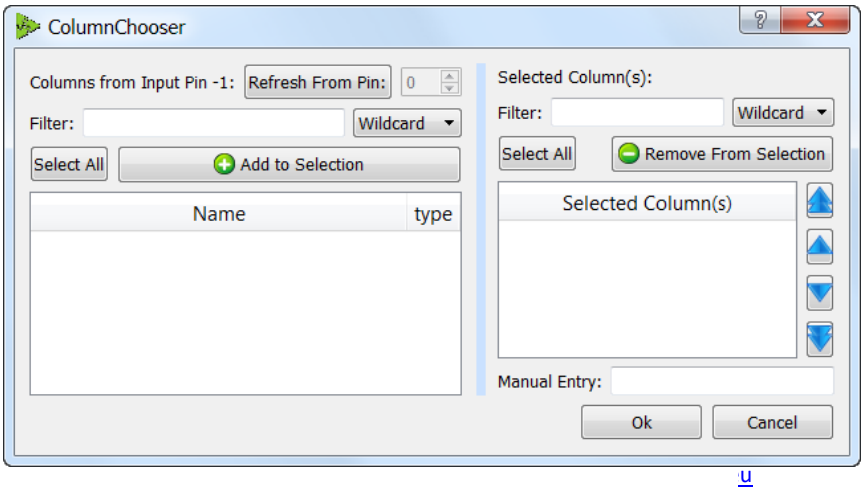
Thus, using the "*Only check sort (with errors)*", you can modify the meta-data of the table without actually sorting the table (which is usually quite a big job to do).

Let's give a small example! Let's assume that we want to simply sort a CSV file. We want to sort the CSV file on the column 'revenue" in increasing numerical order. We will have the following Anatella-Graph:

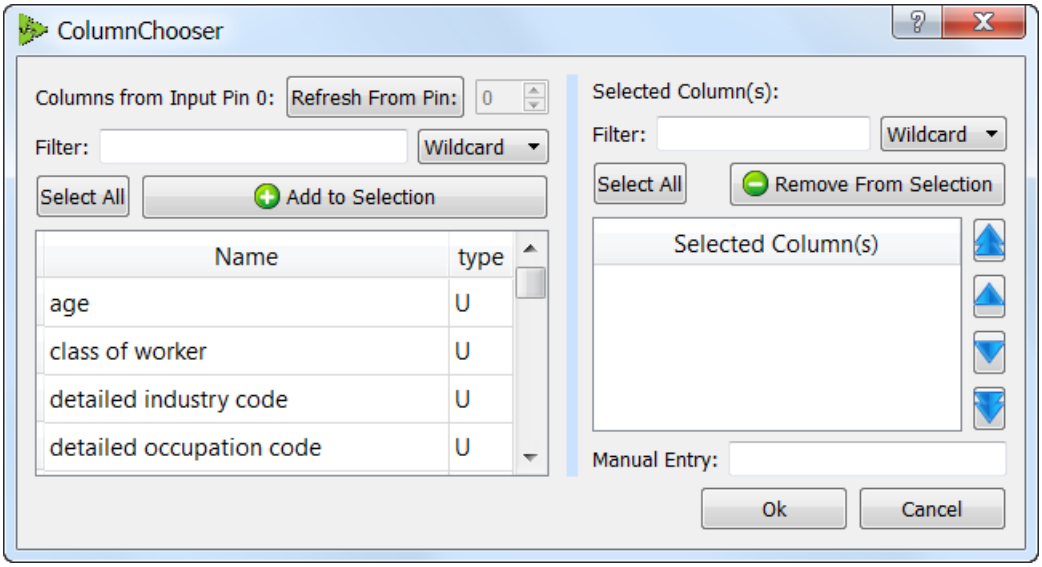


Let's configure the properties of the Sort Action. Click on the 

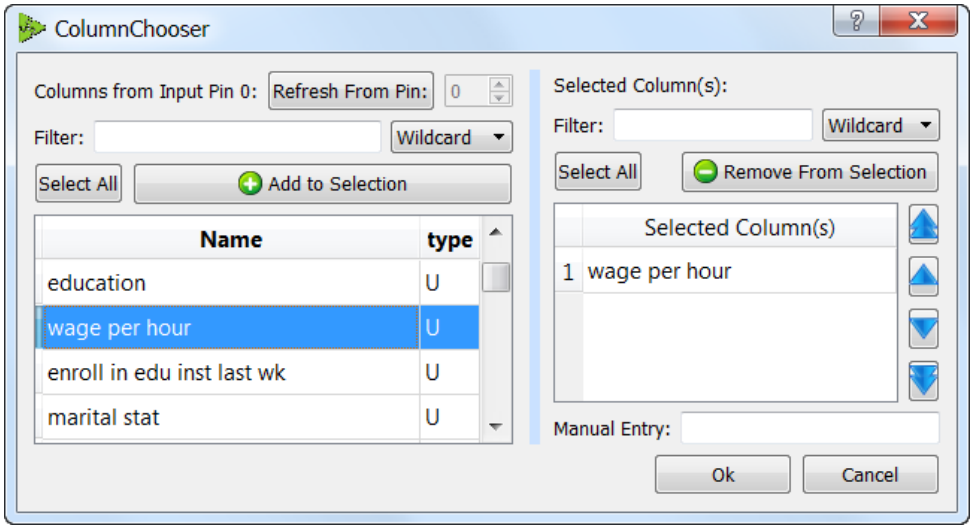
button. The following standard "Column Chooser" window opens (see section 5.1.4. about the "Column Chooser" window):



Click on the Refresh From Pin: button. We obtain the following list of columns:

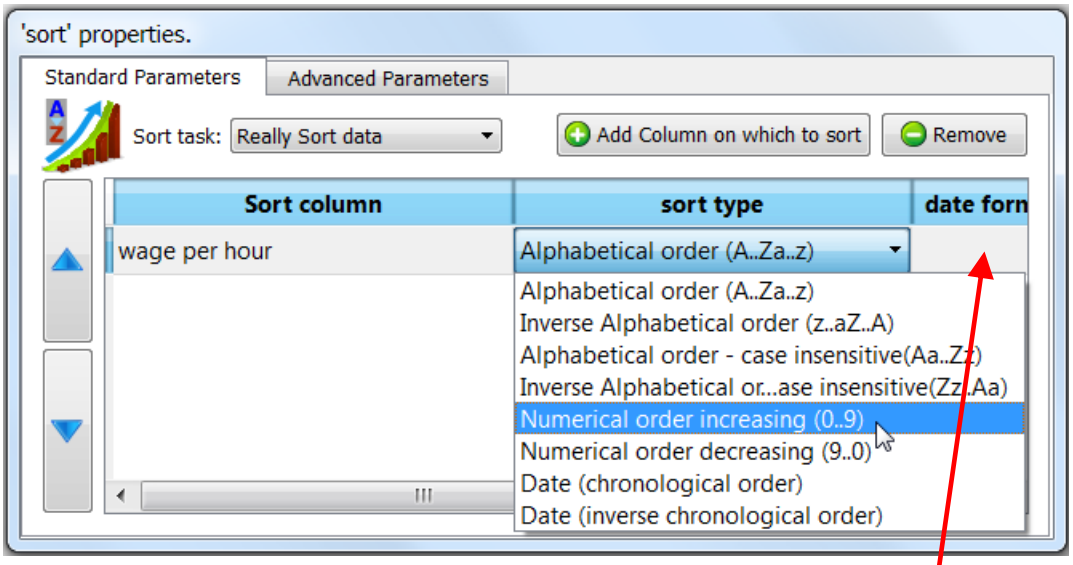


Double-Click on the “wage per hour” column name: you obtain:



Click on the Ok button: the “Column Chooser” window closes. You can now choose the type of sort that you want: we will select “Numerical order increasing (0..9)”:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



As you can see, Anatella currently offers you height "Sort types":

1) Alphabetical order (A..Za..z)
2) Inverse Alphabetical order (z..aZ..A)
3) Alphabetical order - case insensitive(Aa..Zz)
4) Inverse Alphabetical order - case insensitive(Zz..Aa)
5) Numerical order increasing (0..9)
6) Numerical order decreasing (9..0)
7) Date (chronological order)
8) Date (inverse chronological order)

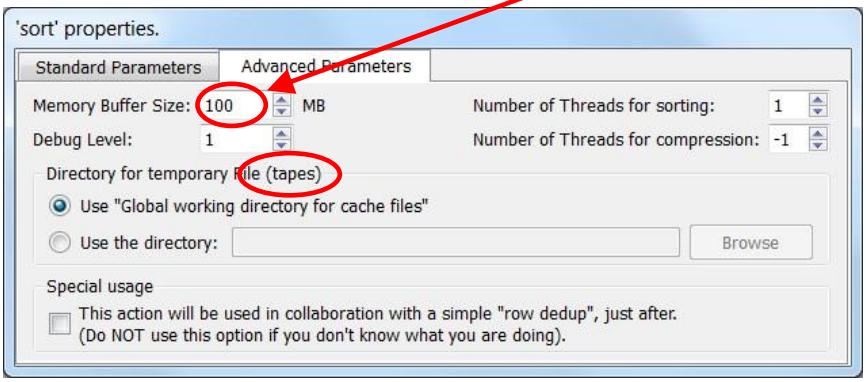
More "Sort types" will be added shortly.

When you use the "Date" sort types, you must specify a "date format" here: See section 5.1.3. for more information about "date formats".

5.5.2.1. What are tapes? What is "Merge Sort"?

In the "Advanced Parameters" tab of the "Sort" Properties window, you will find many references to "tapes". What are these "tapes"? The word "tape" is commonly used in the description of the sorting algorithm used inside Anatella. This algorithm is the following:

1. Create a large RAM memory buffer (whose size is given here:) and fill this buffer with as many rows from the input table as possible.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Sort the in-memory Buffer using a classical sorting algorithm (such as Quicksort, HeapSort, etc.). Write the in-memory buffer (that is now properly sorted) on the hard drive inside a file that is typically named a “tape”. The word “tape” is used because, in the old days, the in-memory buffer was “flushed out” on real tapes such as these:

3. Repeat step 1 and 2 (creating many “tape” files) up to the point that there are no more rows to read in input. At this point, the whole input table has been written into different “tape” files. Each tape is properly sorted.

We will now use an algorithm, named “Merge Sort” that reads all the different tapes and produces, as output, a table that is properly sorted. Let’s now assume (without loss of generality) that we are sorting from the smallest number to the largest number. The “Merge Sort” algorithm is the following:

4. Create a small in-memory RAM table T.
   Use the first row of each of the tape files to fill-in the table T.
   You’ll find on row N from table T, the first row of the tape number N.
   We will now start reading all the tape files, row-by-row.

5. Search for the smallest of the rows in the table T (To remind you: we are sorting from the smallest to the largest): Let’s assume that it’s the row number X.
   5.1. Send as output of the

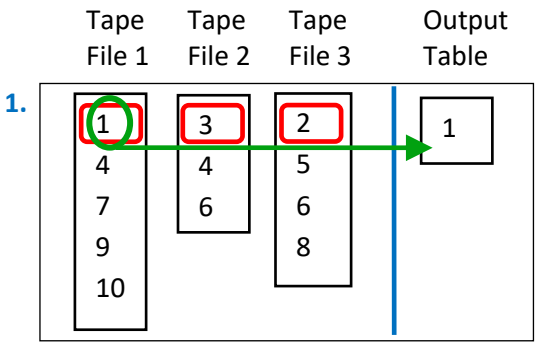
   Sort Action the row number X (i.e. we are adding one row to the output table).
   5.2. Replace the row X from table T with the next row from the tape number X.

6. Repeat step 5 up to the point that we have read all the rows from all the tape files.

To summarize: We iterate. At each iteration, we search for the smallest row amongst all opened tape files, send it as output and replace it with the next row from the same tape file.

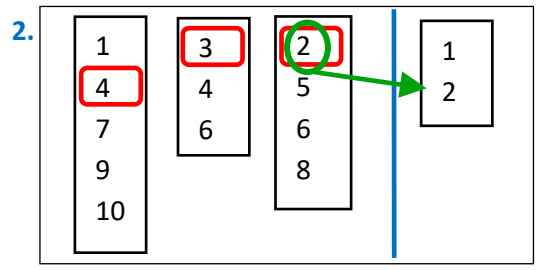
Here is a small illustration of the “Merge Sort” algorithm on a small example:

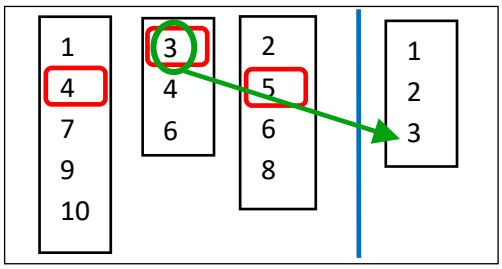
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

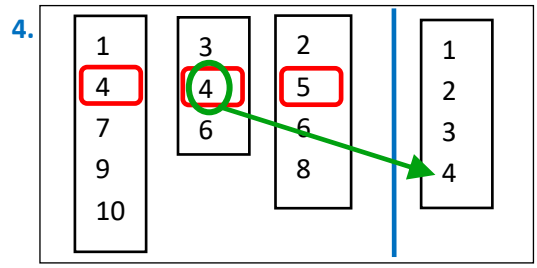


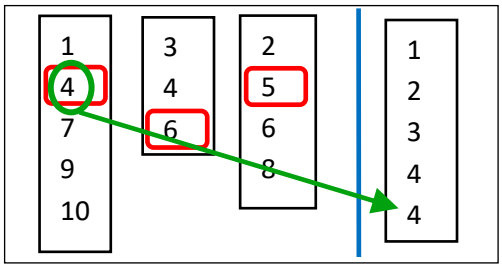
Let's assume that we have three tapes files. The first row of these three tapes file contains, respectively, the numbers 1, 3 and 2. Since we are sorting from the smallest number to the largest, we always select the smallest number and place it inside the "output" table. In the example on the left, the smallest number is "1", so we place "1" inside the output table.

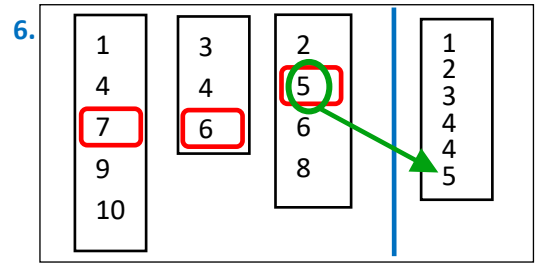
Here are the next iterations of the "Merge Sort" algorithm:

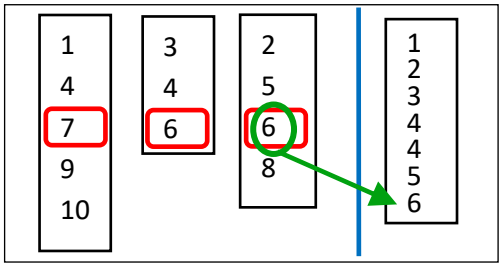


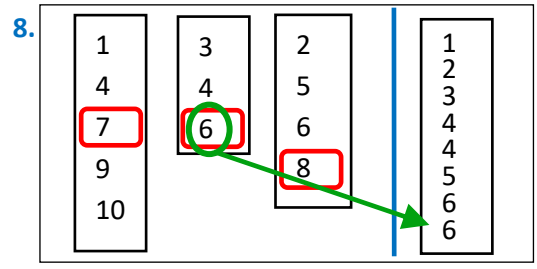


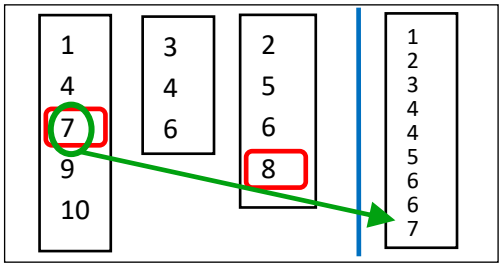


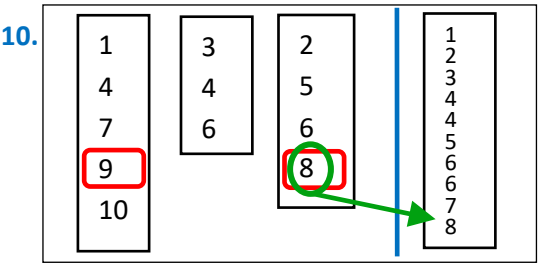


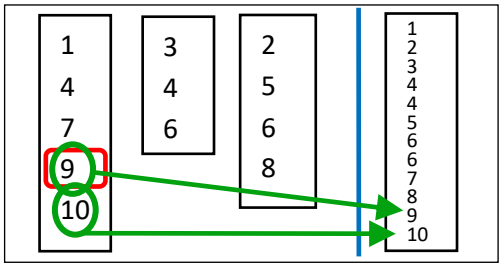












©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.2.2. Faster Alternatives to the "simple" Sort

Sorting is one of the slowest operation that you can perform inside any ETL because it involves writing all the data on the hard drive (in tape files) and, just after, reading all the same data again from the hard drive (during the "Merge Sort"). All these I/O operations (writing and reading from the hard drive) take a considerable amount of time (especially for large input tables). Thus, when optimizing your Anatella graph for speed, you should avoid to use any

Sort Action. There exists many different ways to obtain a sorted table without using a plain

Sort Action: I strongly suggest you to use instead (if possible):

1. a MergeSort Action.
2. a MergeSortInput Action.
3. a partitionedSort Action.

These 3 Actions are **incomparably** faster than a plain

Sort Action.

Sorting is required for the simple Join Action to work properly. If possible, replace the simple Join Action with a MultiJoin Action, to avoid sorting all the data.

Sorting is also required for the "out-of-memory" mode of the

Aggregate Action to work properly. If possible, replace the "out-of-memory" mode with the "in-memory" mode, to avoid sorting all the data.

5.5.2.3. Sorting big data

When sorting big data, you need to properly set the "Memory Buffer Size" parameter of the

Sort Action. This parameter defines:

- the size of the (uncompressed) tapes (The tape files on the HD are typically much smaller because they are in compressed format).
- the size of the memory buffer that is used to create each tape: This means that a value of "1000 MB" consumes 1000 MB of main RAM memory.

If you increase this parameter:

- you'll consume more RAM memory.
- you'll have a smaller number of tape files.

What happens when you want to sort a 4TB table with the default value of "Memory Buffer Size"=100MB? To compute the sort:

1. Anatella creates 4TB/100MB=40.000 tape files (to remind you: 4TB= 4,000GB= 4,000,000MB).
2. Anatella uses the "Merge Sort" algorithm to fusion all the 40.000 tapes into one sorted table. This means that Anatella needs to read simultaneously from 40.000 files. . This will simply not work. This won't work for, basically, 2 reasons:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

a. In Win7, the maximum limit of simultaneously opened file is around 20.000 (and it's even lower on older Windows). Furthermore, reading simultaneously from many different files strongly degrades the performances (because the hard-drive-heads have to constantly physically "jump" from one file to another on the surface of the disk).
b. Opening one .gel_anatella file consumes about 10MB of main RAM memory. Thus, to open the 40.000 tape files, Anatella needs 40.000x10MB=400GB of main RAM memory. This amount of RAM memory is typically not available on standard systems and Anatella refuses to sort the data.

To sort a 4TB table, you should set "Memory Buffer Size"=10GB (i.e. you need to have 10GB of main RAM memory). This will create 4TB/10GB=400 tape files. Thereafter, to "Merge Sort" these 400 tape files, Anatella will only need 400x10MB=4GB RAM memory (which is ok).

To summarize:

- To obtain the best Sorting speed, you should reduce to the minimum the number of tapes. (i.e. you should increase to the maximum the parameter "*Memory Buffer Size*").
- To sort large tables, you must set properly the "*Memory Buffer Size*" parameter.
- You need a large amount of RAM when sorting large table. In general, the more RAM memory you give to the

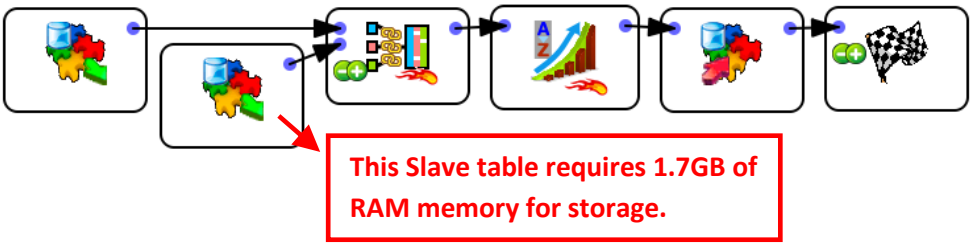
Sort Action, the faster the sort will be.

If you set the parameter "*Memory Buffer Size*" to 10 GB (as in the above example), it also means that the RAM memory consumption of the

Sort Action is around **10GB**. This is not negligible and it might lead to serious difficulties if you are using the multithreading capabilities of Anatella. For more information about this subject see the section 5.3.2.7.

5.5.2.4. Managing RAM memory 1.

We are always tempted to increase the "*Memory Buffer Size*" parameter to the maximum, to obtain the fastest sort. Let's consider the following data transformation graph:



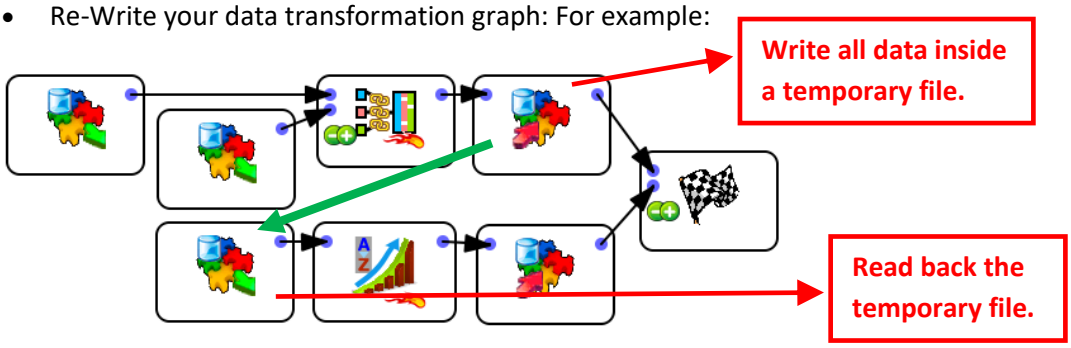
Let's assume that:

- The total available RAM memory inside the computer is 2GB (i.e. it's a 32-bit computer).
- The memory used inside the MultiJoin Action to store the Slave Table is 1.7 GB.

This means that the remaining memory to perform the sort is only around 0.3GB=300MB. This is not much and the sort might be quite slow. Even worse, if you set the "*Memory Buffer Size*" parameter to a value above or near 300MB, the sort will directly fail with a "out-of-memory" message (because you cannot use more than 2GB RAM on a 32-bit computer). There are several solutions:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Optimize the MultiJoin Action: Typically: change the Data Type of some columns in the Slave table so that less RAM memory are required to store the Slave table in memory.
- Re-Write your data transformation graph: For example:



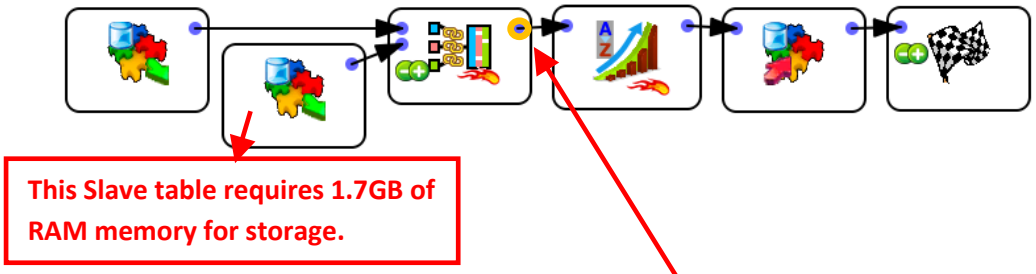
To execute the above data transformation graph, Anatella will:

- Load into RAM memory the Slave Table to compute the MultiJoin Action. The amount of free RAM memory is now only 300MB.
- Compute the join and save the results inside a temporary file.
- As soon as the computation of the temporary file is complete, Anatella will un-load from memory the Slave Table (because we don't need it anymore to compute the join because there is no join to compute anymore). **The amount of free RAM memory is now back to 2GB.**
- Read the temporary file and sort it. You can use 1.5GB of RAM memory to compute the sort (i.e. "Memory Buffer Size"=1500MB).

Please note that this last solution should ideally be used only for a "quick fix" because it's not speed efficient because it implies writing all the data inside a temporary file (and then re-reading it). This extra I/O (i.e. reading & writing all the data on the hard drive) costs a large amount of time.

5.5.2.5. Managing RAM memory 2.

Let's go back to the same data-transformation-graph as in the previous section:



We already assumed in the previous section that:

- The total available RAM memory inside the computer is 2GB (i.e. it's a 32-bit computer).
- The memory used inside the MultiJoin Action to store the Slave Table is 1.7 GB.

Let's do two more assumptions: Let's assume that:

- The user already computed a HD cache at the following location:
  This HD cache has been created by the user to help him during the interactive development of the data transformation graph.
- The "Memory Buffer Size" parameter of the

Sort Action is 1GB.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When we run the graph by clicking the button on the toolbar (i.e. start without deleting any HD cache), Anatella re-uses the available HD cache and it can completely skip the (Multi)Join computation. This means that the amount of free RAM memory is 2GB. Since the Sort Action only requires 1GB of RAM memory, Anatella can easily sort the data.

When we run the graph by clicking the button on the toolbar (i.e. delete all HD caches and then start), Anatella must re-compute the MultiJoin Action (because the HD cache is not available anymore). This means that the amount of free RAM memory available for the sort is only 2GB-1.7GB=300MB. Since the Sort Action only requires 1GB of RAM memory, it fails.

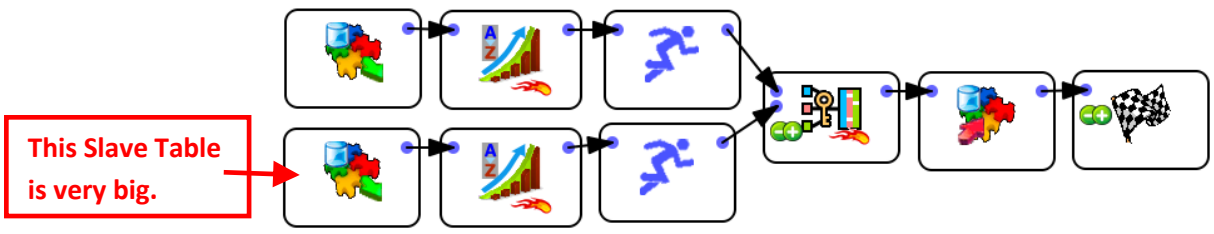
For batch execution, Anatella never uses any available HD cache: it always re-computes everything from scratch (this is to ensure that we are not using old and inconsistent data). This means that, during batch execution, the Sort Action will also fail.

During interactive development, thanks to the usage of HD caches, the RAM memory consumption required to execute your data transformations is usually lower (because you can skip some computations such as the MultiJoin Action in the example here above) than in batch mode or "normal" execution mode. This means that some memory intensive Anatella graph might perfectly run in interactive mode but not in batch/production mode (because of the higher RAM memory consumption in batch/production mode).

Before using any Anatella data-transformation-graph in production, clear all the HD Cache and run it one last time, to ensure that the RAM memory consumption is not excessive.

5.5.2.6. Managing RAM memory 3.

Let's consider the following data transformation graph:



This is a very common situation: We need to join the two tables using the simple Join Action (because the Slave Table is very big and we can't use the MultiJoin Action with very big Slave Table). To use the simple Join Action, we first need to sort the two tables. To (roughly) divide the computation time by two, we'll sort the two tables in parallel (i.e. at the same time), using two

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Multithread Actions. Indeed, in the above graph, 95% of the computation time is used to compute the 2 sorts (sorting is nearly always the slowest Action that you can do inside any ETL).

Let's assume that, for each of the Sort Action, the "*Memory Buffer Size*" parameter is 1200MB. This means that the total amount of RAM memory used by Anatella to execute this graph is 2x1200MB=2400MB=2.4GB (i.e. this graph won't run on a 32-bit computer)(because the two Sort Actions are running simultaneously and their respective memory consumption adds up).

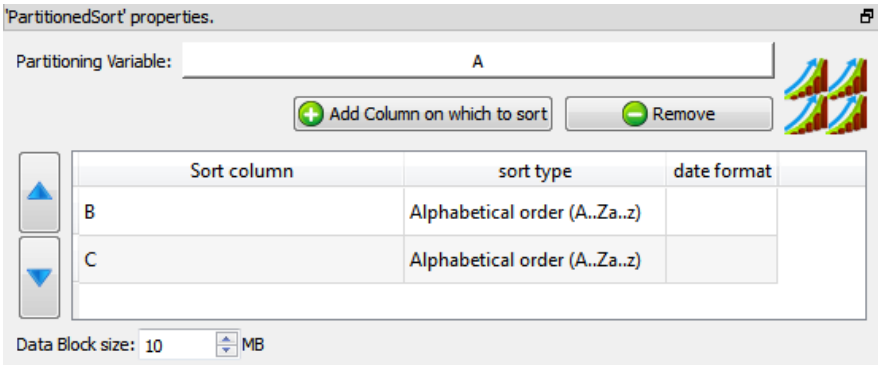
To be able to run this graph on a 32-bit computer (i.e. using less than 2GB RAM), you can either:

- Decrease the "*Memory Buffer Size*" parameter of the two Sort Actions to 800MB.
- Remove the two Multithread Actions. The whole graph will then execute sequentially. More precisely, we'll have:
    1. Anatella creates all the tape files to sort the **first** table. To compute the tape files, Anatella requires 1200MB (i.e. the "*Memory Buffer Size*" is 1200MB). The amount of free RAM memory is 2GB-1200MB=800MB.
    2. Let's assume that there are 30 tape files and that they are now all computed. Anatella free's the RAM memory that was required to compute the tape files and opens all the tape files to fusion them (using the Merge Sort algorithm). The amount of free RAM memory is now 2GB-30x10MB=1700MB (i.e. we lost 300MB to open the 30 tape files).
    3. Anatella creates all the tape files to sort the **second** table. To compute the tape files, Anatella requires 1200MB. The amount of free RAM memory is 1700MB-1200MB=500MB.
    4. Let's assume that there are 25 tape files and that they are now all computed. Anatella free's the RAM memory that was required to compute the tape files and opens all the tape files to fusion them (using the Merge Sort algorithm). The amount of free RAM memory is now 1700MB-25x10MB=1450MB (i.e. we lost 250MB to open the 25 tape files).
    5. The two Sort Actions are using the 30+25=55 tape files to produce, row-by-row, the two sorted table (All these 55 tape files are opened simultaneously). The 2 sorted tables are used, row-by-row, to compute the simple Join Action. The amount of free RAM memory during the join is 1450MB (i.e. we lost 550MB out of 2GB to open the 55 tape files).

5.5.3. Partitioned Sort

Icon:

Property window:



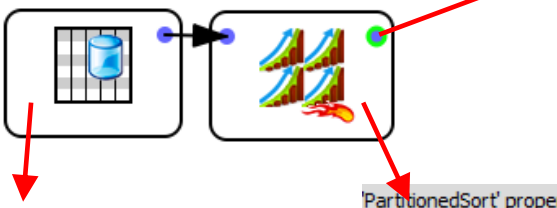
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Sort the input table, partition-by-partition.

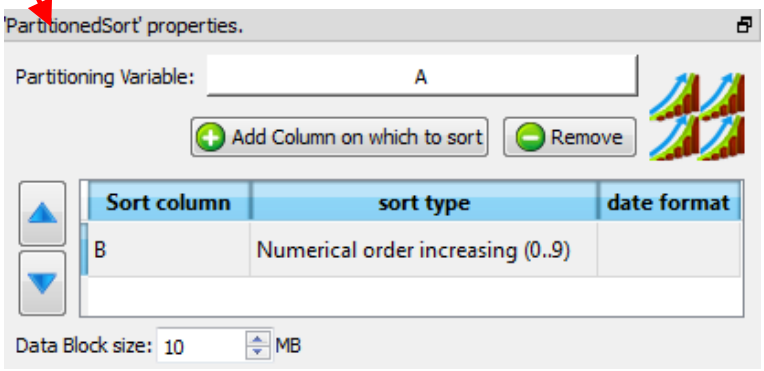
Long Description:
Sort the input table, partition-by-partition. A partition is a set of contiguous rows inside the input table. The end of a partition (i.e. the last row of a partition) is marked by a change of value inside the "partitioning variable" (i.e. The partitioning variable is constant inside one partition).

For example:

A	B	C
1	0	B
1	8	A
2	3	C
2	4	D
0	10	E
3	2	G
3	4	F



A (Partitioning Variable)	B (Most Significant Sort Variable)	C
1	8	A
1	0	B
2	3	C
2	4	D
0	10	E
3	4	F
3	2	G

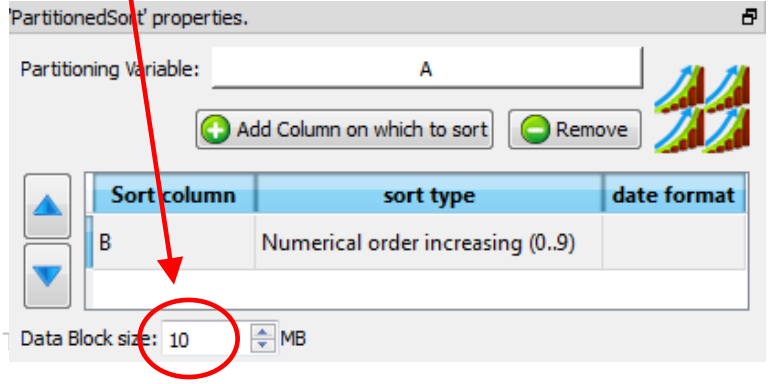


The algorithm is the following:

1. Anatella loads into an in-memory RAM buffer a whole partition.
2. Anatella sorts the in-memory RAM buffer.
3. Anatella outputs each row of the (sorted) buffer.
4. Anatella clears the buffer and, if there are some more input rows (i.e. some more partitions), it goes back to step 1.

The size of the in-memory RAM buffer automatically increases to be able to store a whole partition.

The size increment is given here: . The memory consumption of the Partitioned Sort Action is equal to the amount of RAM memory required to store the largest partition.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In the general case, the output table does not contain any sort-meta-data (to indicate that the output table is, in the general case, not sorted), unless we are in the following "special" case: If the most significant sort-variable of the **input** table is equal to the partitioning-variable, then the sort-meta-data of the output table is set to the following:

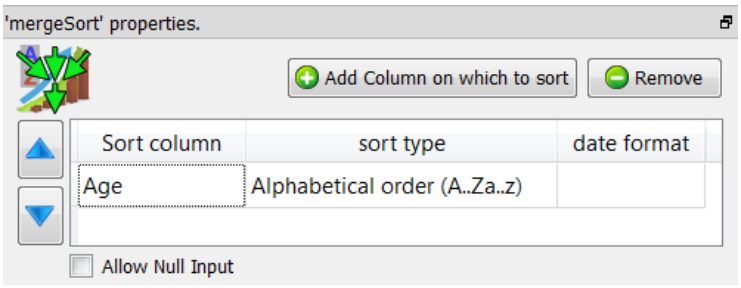
- The most significant sort-variable of the output table is equal to the partitioning-variable.
- The other less-significant sort-variables are set accordingly to the sorting parameters of the Partitioned Sort Action.

The Partitioned Sort Action can run inside a N-Way Multithread Section only if the partitioning variable of the N-Way Multithread Section is equal to the partitioning variable of the Partitioned Sort Action. When executing a Partitioned Sort Action using N CPUs, the RAM memory consumption is multiplied by N.

5.5.4. Merge Sort (High-Speed action)

Icon:

Property window:



Short description:
Append several tables, keeping the sort that is present inside the input tables.

Long Description:

The MergeSort Action takes as input many SORTED tables (on the different input pins) and it gives as output one SORTED table that includes all the rows from all the input tables. The MergeSort Action has basically the same functionality as the Append Action except that it interleaves the rows of the input tables to produce a SORTED table as output. The sort-order specified inside the MergeSort Action must match the sort-order of the input tables (…and all input tables must be sorted in the same way). At run-time, the MergeSort Action analyzes the meta-data of all the input tables to check if they are properly sorted (…and abort if that's not the case).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

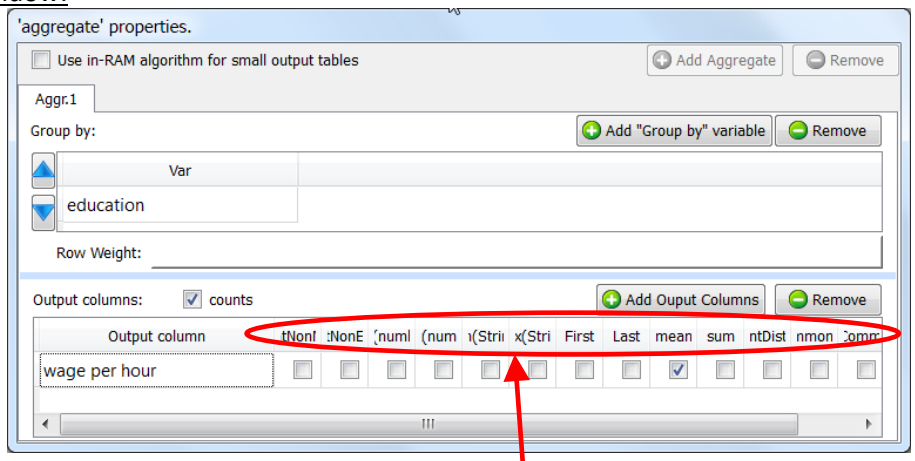
The algorithm used to produce the output table is the "Merge Sort" algorithm that has been described in section 5.5.2.1. (about the

Sort Action).

5.5.5. Aggregate (Group by) (High-Speed action)

Icon:

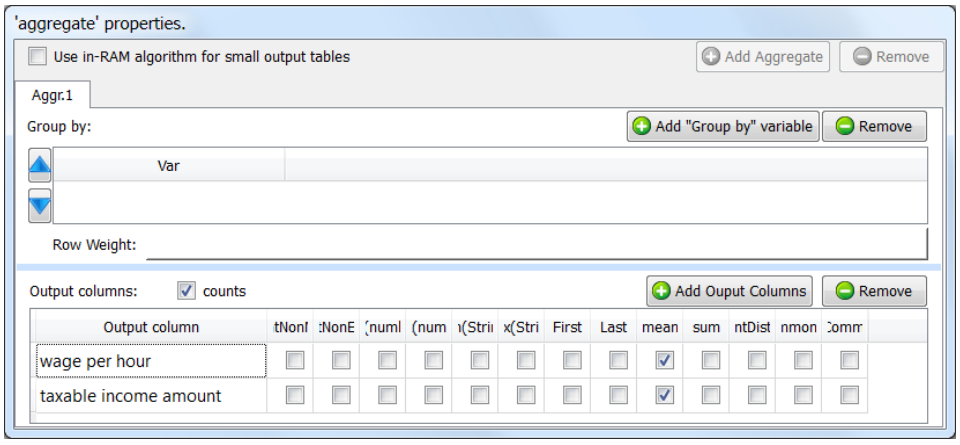
Property window:



Short description:
Aggregate (or Group By) rows.

Long Description:
This action aggregates (or Group By) rows.

You can click on the header of the "output column table" to check all the checkboxes of the column. If you don't provide any "group by" variables, the whole input table will be used to compute only one output row. For example, these settings:



… will generate a one-row output table with the 2 columns "wage per hour_mean" and "taxable income amount_mean".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can give several "Group by" variables, there will be as many output rows as there are of modality-combinations into your data. For example, if your "Goup-By-variables" are: Wealth (Poor, Rich), Age (Young, Middle, Old), Sex (Woman, Man), then you will have as output these 12 rows:

Idx	Wealth	Age	Sex
1	Poor	Young	Woman
2	Poor	Young	Woman
3	Poor	Middle	Woman
4	Poor	Middle	Woman
5	Poor	Old	Woman
6	Poor	Old	Woman
7	Rich	Young	Man
8	Rich	Young	Man
9	Rich	Middle	Man
10	Rich	Middle	Man
11	Rich	Old	Man
12	Rich	Old	Man

Note that the combination (Rich,Young,Man) is very un-likely, so it might not appear as output in the result table.

There are two operating modes for this Action:

1. **In-Memory Mode.**

   In this mode, Anatella is building "in-memory" all the output table(s) (…and there can be many output table(s)).

   Anatella read the input table row-by-row: Each time a **new** combination (in technical term: a new "t-uple") is found in the input table, Anatella extends the "in-memory" output/result table (it adds a new row) to store the new aggregations for this particular t-uple. If an **already known** combination is found, Anatella simply updates the aggregations to take into account the new input row.

   In this mode, Anatella reads the whole input table before producing any output row.

   This mode is very memory-hungry because it needs to store in RAM memory all the output tables. If your output tables do NOT fit into RAM memory (e.g. on a 32 bit computer: they are larger than 2GB), then Anatella won't be able to compute the aggregation (or, it will be intolerably slow because of "memory swapping"). Thus, if you have very big output tables, you must use the Out-of-Memory Mode.

   When using the "in-memory" mode, the Aggregate Action cannot be included inside a "N-Way Multithread Section" (see section 5.3.2.5. to know more about this subject).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. **Out-of-Memory Mode.**

This mode allows Anatella to compute any aggregation, whatever the size of the (unique) output table.

In order to use this mode, your input table must be sorted on the "Group by" variables. This is somewhat annoying because "sorting" is always a very slow operation (that you should thus avoid). Thus, if you have small output tables, you should rather use the "in-memory mode" (unless your input table is already sorted "for free").

Anatella read the input table row-by-row: Because of the sort, all the rows that have the same combination (i.e. the same value for the "Group by" variables)(in technical term: the same "t-uple") are contiguous. If an **already known** combination is found, Anatella simply updates the aggregations to take into account the new input row. If the combination **changes**, then Anatella directly produces a new output row with the aggregation values of the t-uple that "just ended".

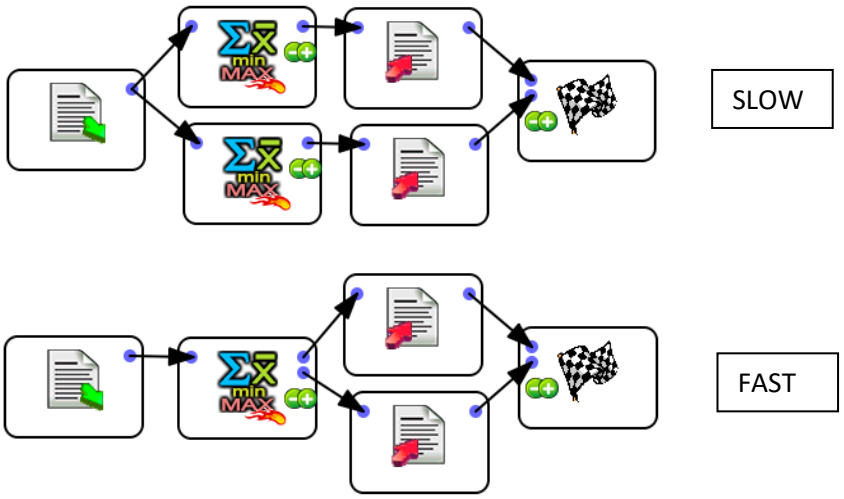
In this mode, Anatella is producing output rows while reading the input table.

This mode consumes a negligible amount of memory space (in opposition to the "in-memory mode").

When using the "out-of-memory" mode, the Aggregate Action can be included inside a "N-Way Multithread Section" (see section 5.3.2.5. to know more about this subject), provided that the input table is still correctly sorted on the "Group by" variables.

5.5.5.1. Using the "in-memory mode" to compute several aggregations simultaneously

When using the "in-memory mode" of the Aggregate Action, you can compute **several** aggregations (i.e. output tables) at the same time. For example, the two following Anatella-Graphs are equivalent but the second one is (at least) two times faster than the first one:



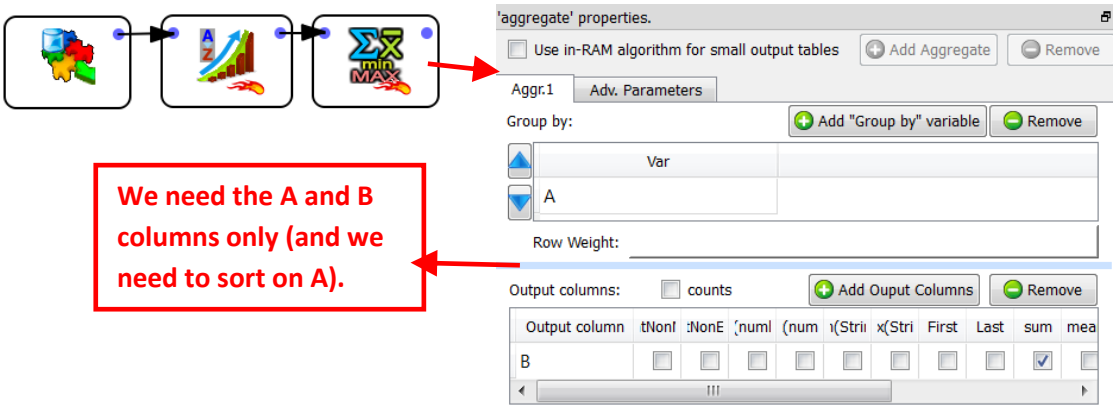
Inside the second graph here above, we used the fact that the Anatella "Aggregate Action" is able to compute several aggregation tables "in parallel", requiring only one pass over the database to compute

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

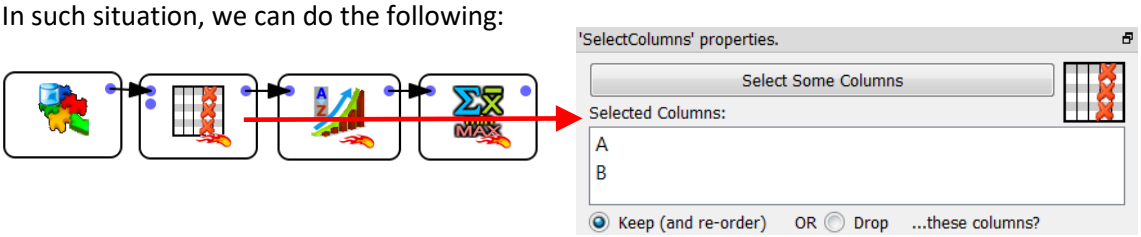
all the aggregation tables. This is a unique functionality of Anatella. Thanks to this functionality, the running-time of an Anatella-transformation-graphs containing several aggregations is usually divided by 2 or 3 (or even 4) compared to a standard implementation based on ELT techniques and simple SQL scripts. Anatella delivers you high-performance & ultra-fast computations!

5.5.5.2. Optimizing the sort before the "out-of-memory" Aggregate

Let's assume that we have:



In such situation, we can do the following:



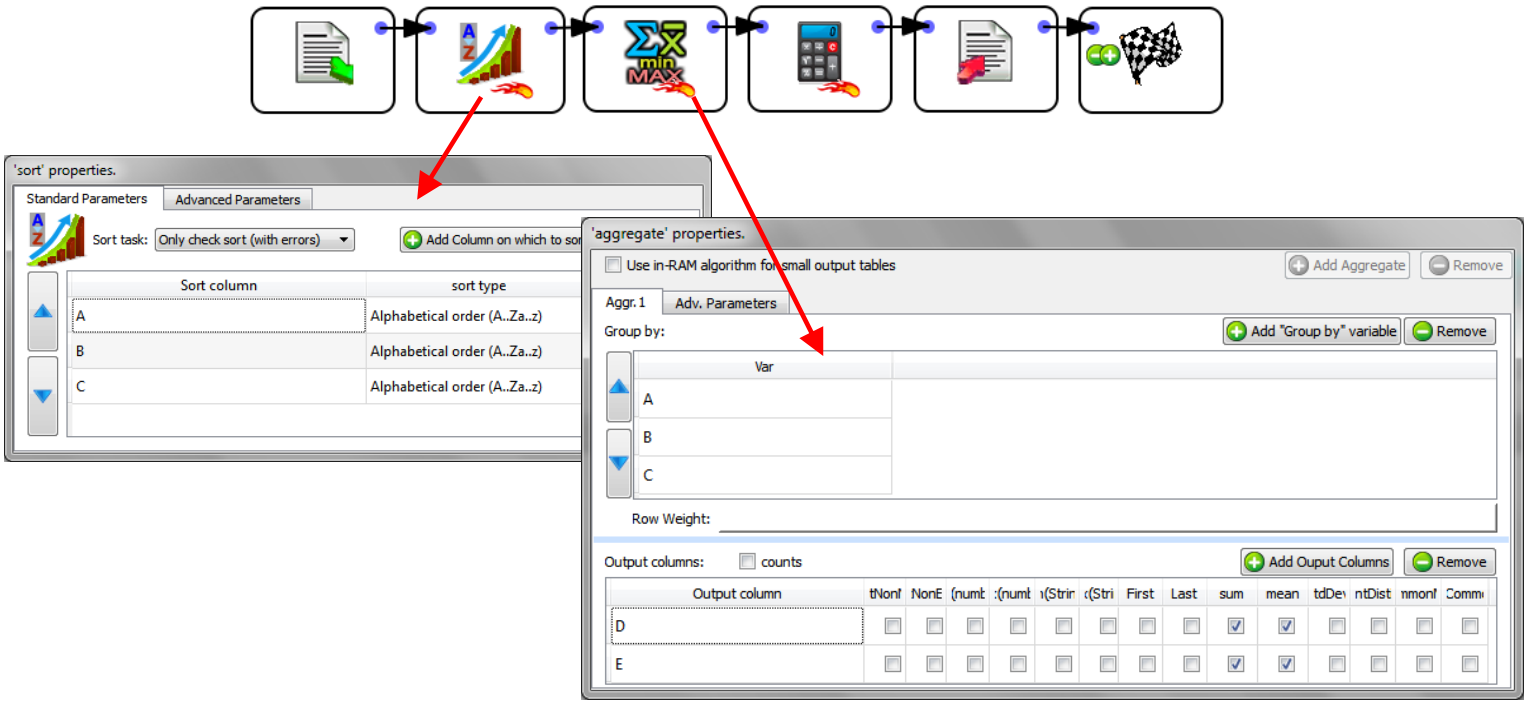
To compute the above aggregate, we only need to sort a table with the A and B columns only (and not the whole table, with all the columns). Thus, before the

Sort Action, it's interesting to place a ColumnFilter Action. To have the highest computation speed, you should always reduce to the minimum the volume of data to sort.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.5.3. Using the "out-of-memory mode" inside a N-Way Multithreaded Section

Let's assume that we want to multithread/parallelize this simple Anatella Transformation Graph:



When the Aggregate Action starts, it looks at the meta-data of the input table to check if it's properly sorted on the columns A, B & C. …And since that's the case (because of the Sort Action running in "*Check sort with error*" mode), it can proceed computing the aggregations.

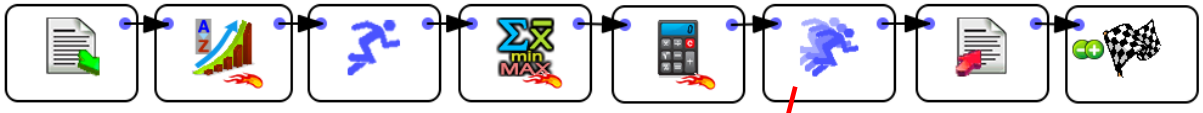
The above graph is equivalent to the SQL command:

"SELECT *sum(D) as D_sum,*
sum(E) as E_sum,
mean(D) as D_mean,
mean(E) as E_mean,
FROM *table*
GROUP BY *A,B,C*"

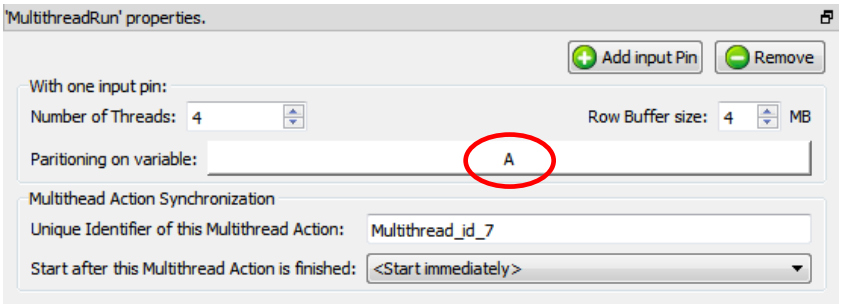
…followed by some small computation based on A, B, C, D, D_sum, E_sum, D_mean, E_mean.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's now include the Aggregate Action, inside a N-Way Multithread Section:

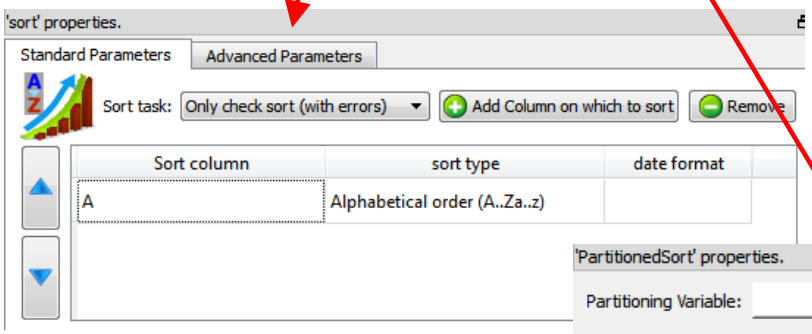


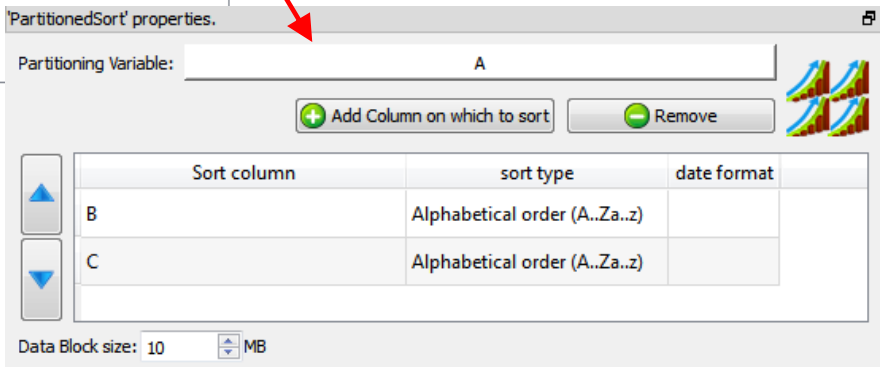
By default, the above graph won't run because the meta-data of the input table of the Aggregate Action says that the input table is not sorted (by default, any sort-meta-data is lost at the start of the N-Way Multithread Section). To "keep" the sort-meta-data inside the interior of the N-Way Multithread section, we must set the partitioning parameter of the second Multithread Action to "A":



This is a special case: When the partitioning parameter is equal to the most significant column of the sort-meta-data, then the sort meta-data is kept inside the interior of the N-Way Multithread section. (So that the Aggregate Action works, once again, properly)

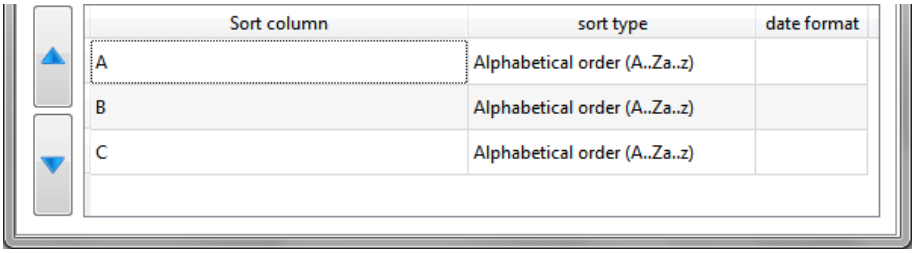
Let's now assume that the text file that is used as source data for the Anatella graph is sorted on the column A **only** (and not on the columns A, B & C, as previously). We'll thus have:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In the general case, the output table of the Partitioned Sort Action is not sorted (i.e. it does not contain any sort-meta-data at all). In the above example, we are in a special case: The partitioning variable of the Partitioned Sort Action is equal to the most-significant sort-variable of the input table. In this special case, the sort-meta-data of the output table is not empty (see section 5.5.2). In the above example, the sort-meta-data is automatically set to:



Sort column	sort type	date format
A	Alphabetical order (A..Za..z)	
B	Alphabetical order (A..Za..z)	
C	Alphabetical order (A..Za..z)	

(So that the Aggregate Action works, once again, properly!)

The above Anatella graph is very efficient because:

a) It **computes aggregations using many CPUs** (because the Aggregate Action is inside a N-Way Multithread section).

b) The aggregations are computed using the "out-of-memory" mode, meaning that we can handle **output tables of unlimited size**.

c) When using the "out-of-memory" mode to compute aggregation, the input table must be sorted on all the "group by" variables. In the above example, the input table was only partially sorted on one of the "group by" variable (i.e. it was sorted only on the column A, but not on the columns B and C). Although **the source table was not properly sorted, we were nevertheless able to compute the aggregations**, thanks to the Partitioned Sort Action that "extended" the sort-meta-data (to include the columns B and C) so that the Aggregate Action still works properly.

d) We were able to run on many CPUs the Partitioned Sort Action (because it's inside a N-Way Multithread section). Usually, sorting is a very slow operation and thus it's very nice to be able to **easily use many CPU's to sort the data** because it reduces considerably the computation time.

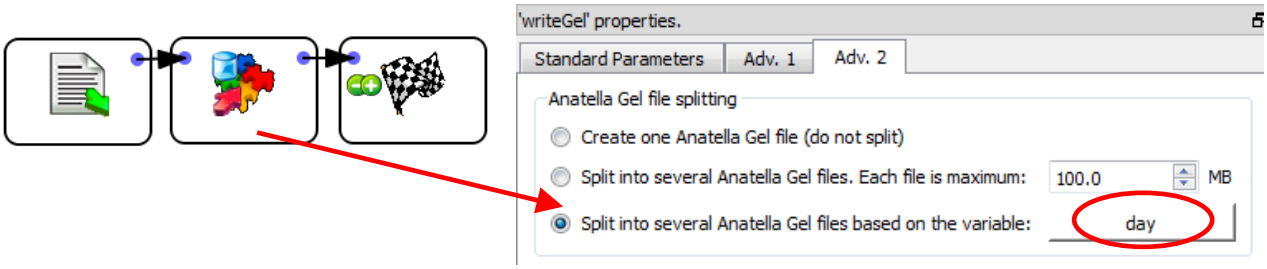
e) It's **using a very small amount of RAM memory** (there are no Sort Action and the Aggregate Action are running in "out-of-memory" mode).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.5.4. Avoiding a costly "sort"

In the above example, we assumed that the text file that is used as source data for the Anatella graph is sorted on the column A. If that's not the case, we can use the following procedure:

- Read the source text file and write it back inside many different .gel_anatella files using the "split variable" option of the writeGel Action. For example, this Anatella graph splits the data in different .gel_anatella file, one file per each different day:



- Sort all the different .gel_anatella files that were produced at the precedent step. This sort can easily be run in parallel (one CPU for each different day/file).

  I suggest you to use the "ProcessRunner" JavaScript class to run on parallel on several CPUs an Anatella graph that sorts (using a simple Sort Action) one particular day that is specified as command-line parameter to the graph (i.e. as "*Graph Global Parameter*", see section 5.2.5.2.).

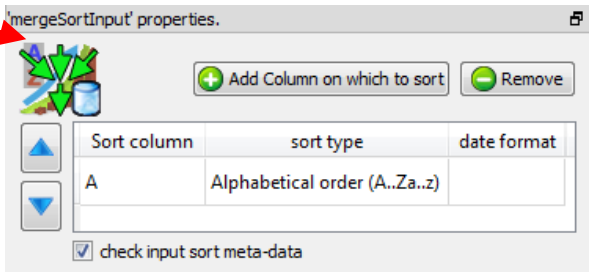
  Although we are using the simple Sort Action, we still get high speed because:

  - We are able to run the sort on many CPUs (one CPU per each different day/file).
  - The volume of data to sort is limited (it's only **one day**) and it can happen that the Sort Action is able to perform the sort entirely in RAM-memory (without using any tape files). When this happens, the sort is a lot faster.
  - If the data arrives on a daily basis, we can avoid re-computing all the sorts for all the previous days (by keeping on the hard drive the sorted .gel_anatella files that were previously computed) and we only sort the "last" file/day that we just received. This leads to a somewhat **"incremental" sorting algorithm** that only needs to sort a very small quantity of data before being able to compute very efficiently large aggregates.

- Use the MergeSortInput Action (or the MergeSort Action) to obtain, from the different "locally sorted" .gel_anatella files, one "**globally sorted**" table. We'll have:

List of filenames. This list contains the filename of all the sorted .gel_anatella files.

You can use the fileListFromObsDate Action (see section 5.21.5.) to automatically generate the file list.



This last data-transformation-graph is even more efficient than the previous one because:

a) it starts from a set of "locally sorted" file (i.e. we used the term "locally" because we sorted "locally" each day/file on the column A and we managed to avoid sorting "globally" ALL the data from ALL the days).
b) Each day/file is only "partially" sorted (i.e. each day is sorted on the column A **only**).
c) It allows you to use an **"incremental" sorting algorithm** that reduces the computing time by several orders of magnitude. We used the term "incremental" because we only need to sort the small quantity of **new** days of data that we just received (and not all the days).
d) It manages to keep the 5 advantages that were explained for the previous Anatella graph:
    - It **computes aggregations using many CPUs** (still reducing computing time).
    - You can have **output tables of unlimited size.**
    - It's not using any

Sort Actions (that are slow and memory hungry).
    - The little amount of **sorting is performed on many CPUs** (still reducing computing time).
    - It's **using a very small amount of RAM memory.**

5.5.5.5. Dynamic Aggregation

It can happen on some occasion that you don't know "in advance" the precise nature of the aggregation that you want to compute: i.e. you need to run some computation to now exactly which aggregations to compute. In such situation, you should use the second and third input pin (pin 1 and 2) from the

Aggregate Action. More precisely,

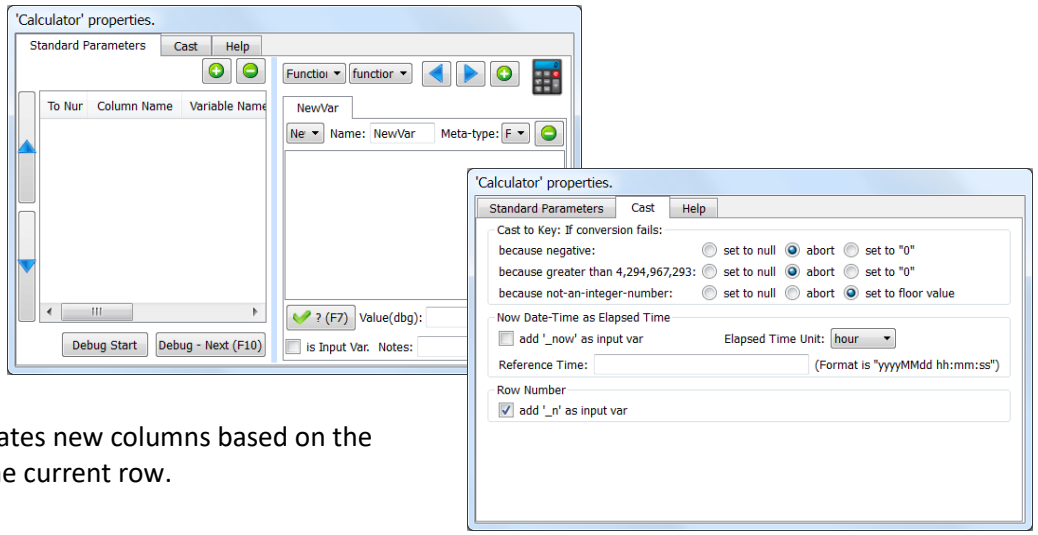
- The second table contains the "group by" information. It only has one column that contains the names of all the "group by" columns.
- The third table contains the "output column" information. It only has two columns:
    - The first column is the name of the variable to process/aggregate.
    - The second column contains a string that describes the aggregate to compute. The accepted values are (case insensitive):
        - IMin (number minimum)
        - IMax (number maximum)
        - SMin (string minimum)
        - SMax (string maximum)
        - First
        - Last
        - Sum
        - Mean
        - StdDev
        - CountDistinct
        - CountNonEmpty
        - CountNonNull
        - MostCommonModality
        - CountMostCommonModality

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.6. Calculator (High-Speed action)

Icon:

Property window:

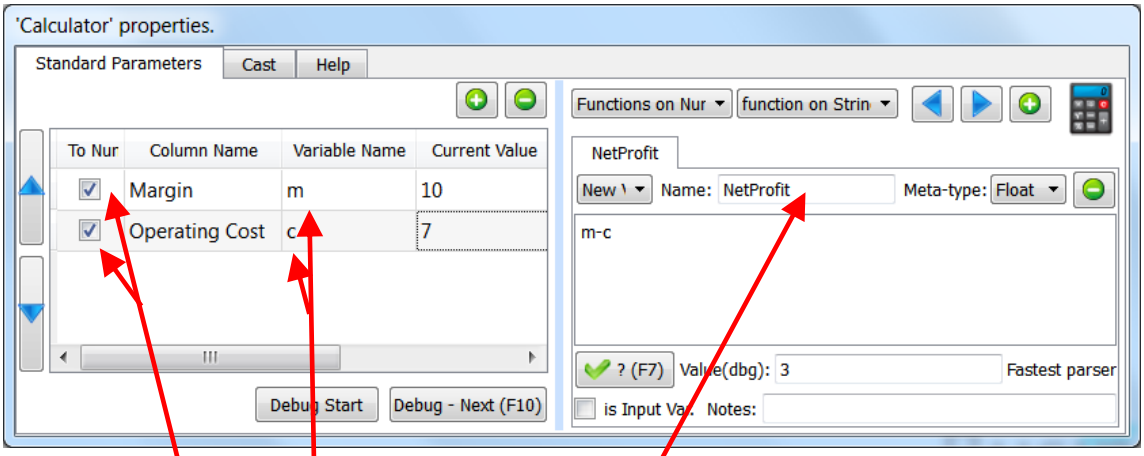


Short description:

This creates or updates new columns based on the other columns in the current row.

Long Description:

Here is a first example: We want to compute the "NetProfit" based on the columns "Margin" and "Operating Cost" (NetProfit=Margin-Operating Cost). We'll have:



Some remarks about the above example:

a) We created a new column named "NetProfit"

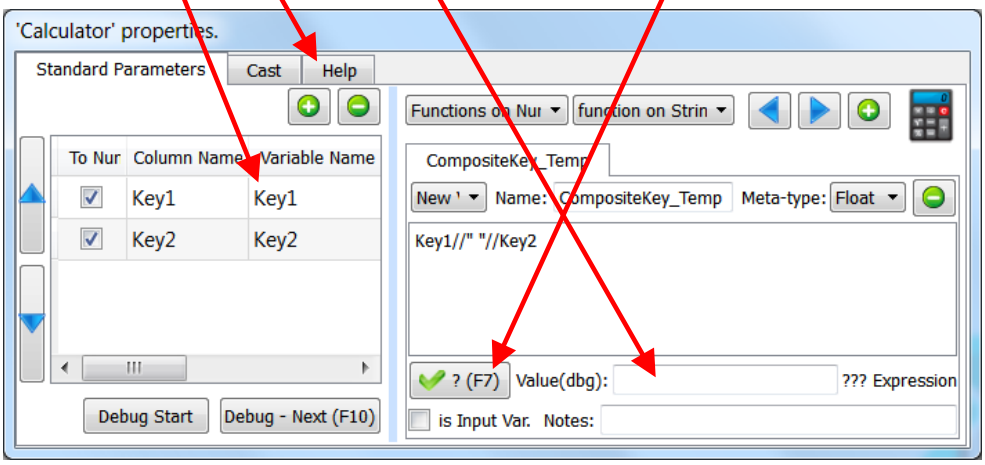
b) To compute "NetProfit", we used two other columns: "Margin" and "Operating Cost". Inside the mathematical expression defining the "NetProfit" these 2 columns were, respectively, named "m" and "c": "m" and "c" are **variables** inside the mathematical expression "m-c" defining the "NetProfit". Variable names can use the following characters: a-z,A-Z,0-9,_ (spaces are not allowed in variables names). Variable names cannot start with a number (thus "1A" is an invalid variable name but "_1A" is ok).

c) Variables (such as "m" and "c") can either be of the numerical type or of string type.
You define here: the type (number or string) of each of the variables (in the above example, the 2 variables are of the number type, thus the 2 checkboxes are both enabled).
There are no automatic conversions from one type to another: you must use the function **atof** (to convert a string to a floating-point number), **atoi** (to convert a string to an integer number), **ftoa** (to convert a floating-point number to a string) to compute the conversions.
Most operators require a specific type to work: For example, you cannot multiply together two variables of the string type: only variables of the numerical type can be multiplied together.

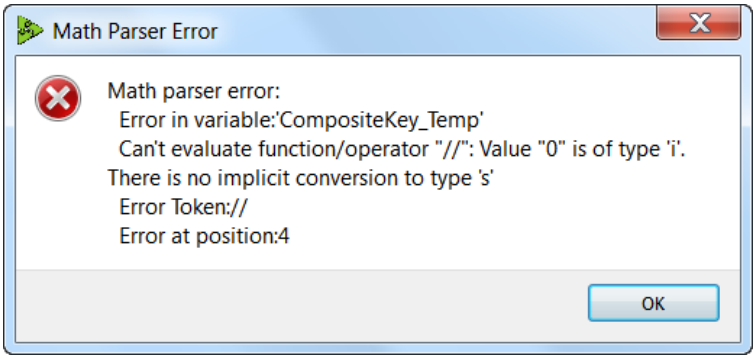
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

d) You have a description of all the available operators inside the “Help” tab of the Calculator Action:

e) To check if your mathematical expression is valid (e.g. to check if you used the required variable type: numerical or string), you can click the ? (F7) button: Anatella will use the “Current Value” of each variable to evaluate your mathematical expression and, if there are no errors, it will give the result here: Here is another example:

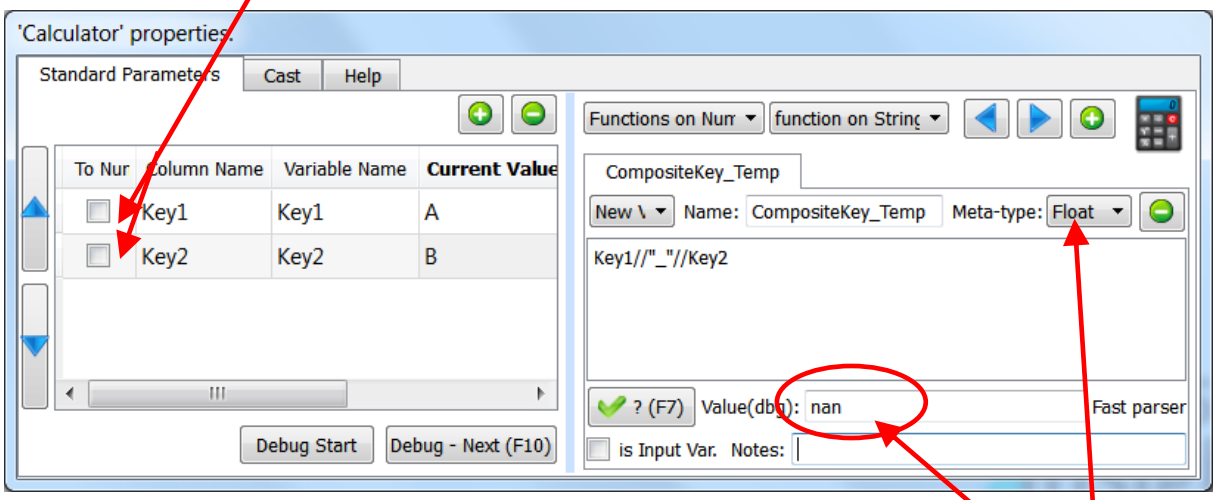


When you click the ? (F7) button, you get:

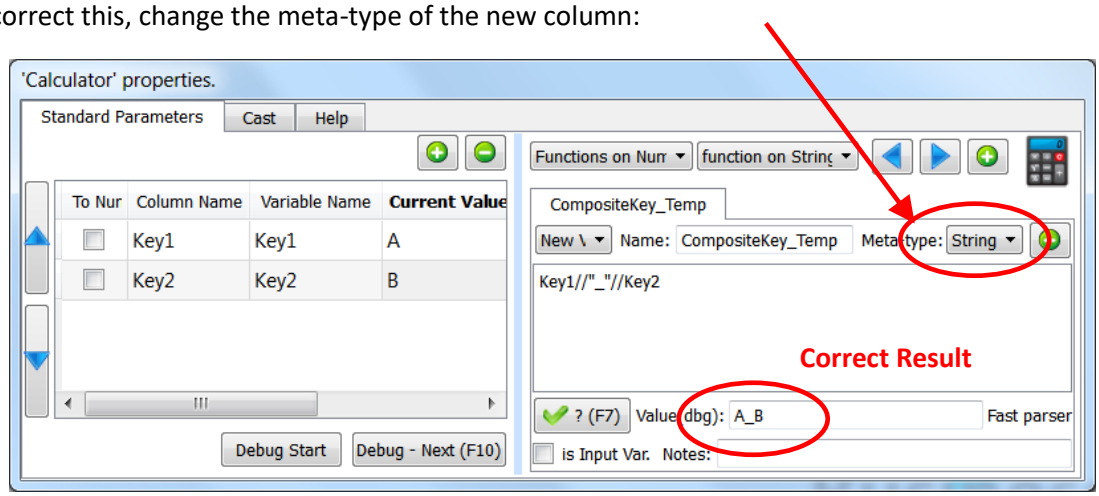


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The above error message tells you that your variable types are wrong: You defined your 2 variables “Key1” and “Key2” as “integers” (i.e. type ‘i’) but you used them with the string-concatenation-operator (the “//”) that requires the type ‘s’ (i.e. the string type). The solution is simple: click here:

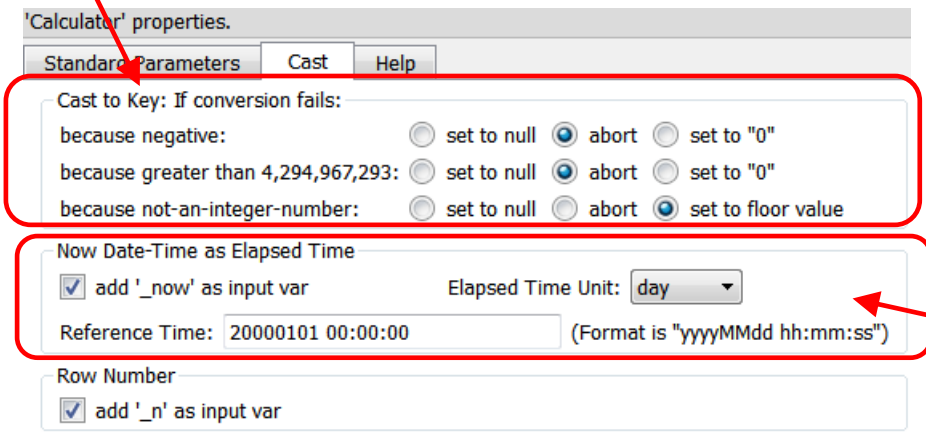


When you click the ? (F7) button again, you get no errors but the result is strange:
(We were expecting “A_B”, as result, but we got “nan” instead). How did we obtain “nan”?
The answer is simple: Anatella evaluates your expression and correctly obtains “A_B” but, after that, it attempts to store the result “A_B” into a column of the floating-point type:
The result of the conversion of “A_B” to a floating-point number is “nan” (i.e. “**N**ot-**A**-**N**umber).
To correct this, change the meta-type of the new column:



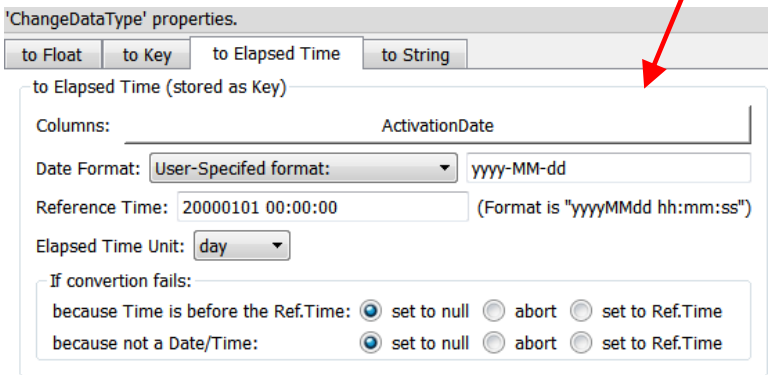
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

f) Once Anatella has finished evaluating your mathematical expression, it stores the result into the (new) column inside the output table of the Calculator Action. The Meta-type of the (new) column must be compatible with the result's type (otherwise you get strange results such as the "nan" result of the example on the previous page). In particular, the conversion of the result to the "Key" Meta-type might fail for different reasons. Use these settings: to parameterize how Anatella handles the conversion to the "Key" Meta-type:

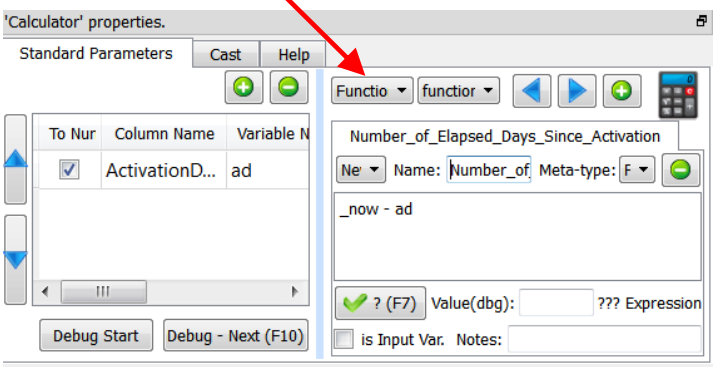


5.5.6.1. Manipulating Dates-stored-as-Key.

You can use the Calculator Action to make different computations on Dates&Times in a very efficient way: For example: This will compute the "*Number of Elapsed Days Since Activation*", based on the column "*ActivationDate*" that is converted as a **Date-stored-as-Key**:



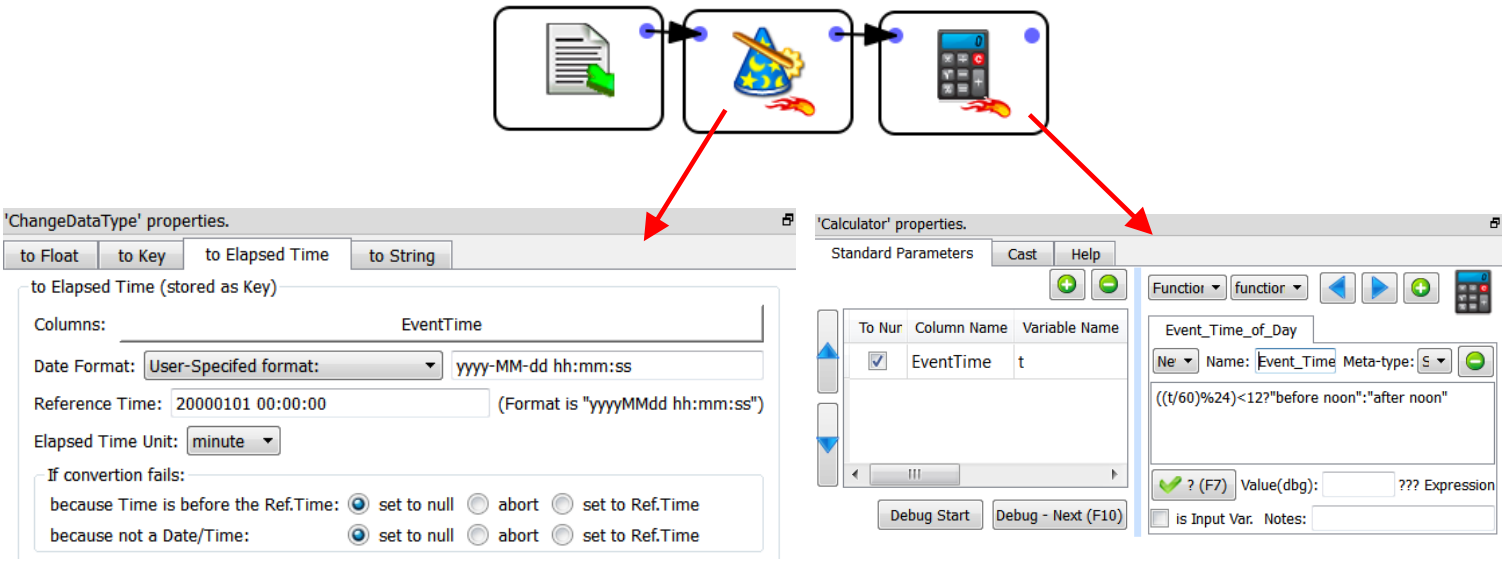
Convert the column "ActivationDate" to the Date-stored-as-Key data-type (We parameterized the conversion so that it contains the number of days since the 1st January 2000).



Compute "*Number of Elapsed Days Since Activation*" (We parameterized the variable "_now" so that it's the number of days since the 1st January 2000).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is another example: This will compute the "Event Time Of Day" (i.e. "before noon" or "after noon"), based on the column "Event Time" that is converted as a **Date-stored-as-Key**:



Convert the column "EventTime" to the Date-stored-as-Key data-type (We parameterized the conversion so that it contains the number of minutes since the 1st January 2010).

Compute "*Event Time of Day*":

- **The division by 60 is used to convert from minutes to hours.**
- **The modulo ("%") operator is used to get the "hour in the day".**
- **The ternary operator ("x?y:z") test the "hour in the day" and outputs the corresponding string.**

The modulo operator can be used to know the "hour in the day", the "minute in the day", the "day in the week", etc.

5.5.6.2. Handling NULL Values

Most programming languages (such as Java, C/C++, Delphi, etc.) are using a standard 2-way boolean logic:

	a	b	a_or_b (a||b)	a_and_b (a&&b)	a_equal_b (a==b)	a_notEqual_b (a!=b)	a_<_b	a_>_b	not(a)
1	0	0	0	0	1	0	0	0	1
2	0	1	1	0	0	1	1	0	1
3	1	0	1	0	0	1	0	1	0
4	1	1	1	1	1	0	0	0	0

Note: "0" means FALSE; "1" means TRUE.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To handle null values, Anatella uses the same three-way logic as the standard SQL:

	a	b	a_or_b (a||b)	a_and_b (a&&b)	a_equal_b (a==b)	a_notEqual_b (a!=b)	a_<_b	a_>_b	not(a)
1	0	0	0	0	1	0	0	0	1
2	0	1	1	0	0	1	1	0	1
3	0	NULL	NULL	0	NULL	NULL	NULL	NULL	1
4	1	0	1	0	0	1	0	1	0
5	1	1	1	1	1	0	0	0	0
6	1	NULL	1	NULL	NULL	NULL	NULL	NULL	0
7	NULL	0	NULL	0	NULL	NULL	NULL	NULL	NULL
8	NULL	1	1	NULL	NULL	NULL	NULL	NULL	NULL
9	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

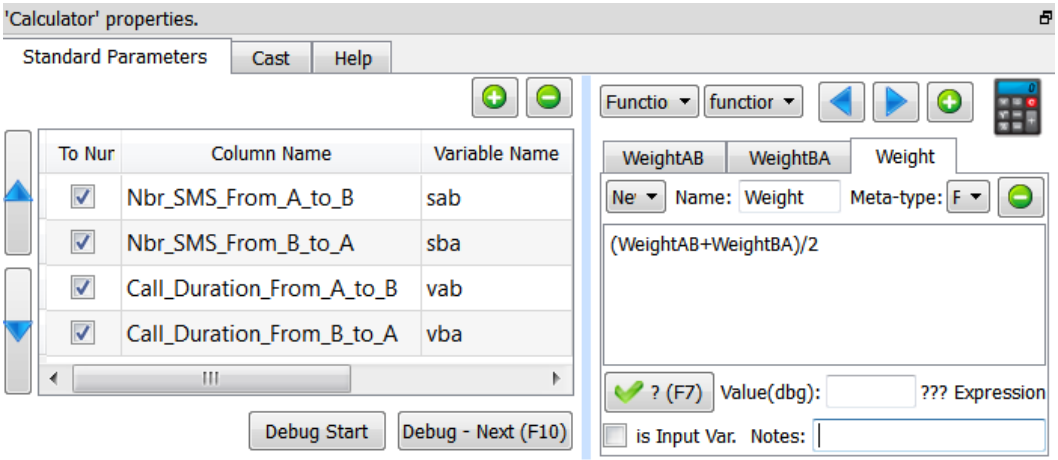
Note: "0" means FALSE; "1" means TRUE; "NULL" is the null value.

This means that the expression (*a==_null)* will always return null as result. As a consequence, the expression (*a==_null?"foo":"bar"*) will always return "bar". The right way to test if a variable is null is the following: (*isNull(a):"foo":"bar"*) or even better (because it's faster): (*nvl2(a,"foo","bar")*).

The word "_null" inside an expression cannot be used inside a boolean operation but it's perfectly valid to use it as "return" value. For example: (*a<b:_null:1*) or (*a<b:_nullS:"1"*) are perfectly valid expressions (use "_nullS" when the return value is of the string/unknown type and use "_null" when the return value is of the Float or Key type).

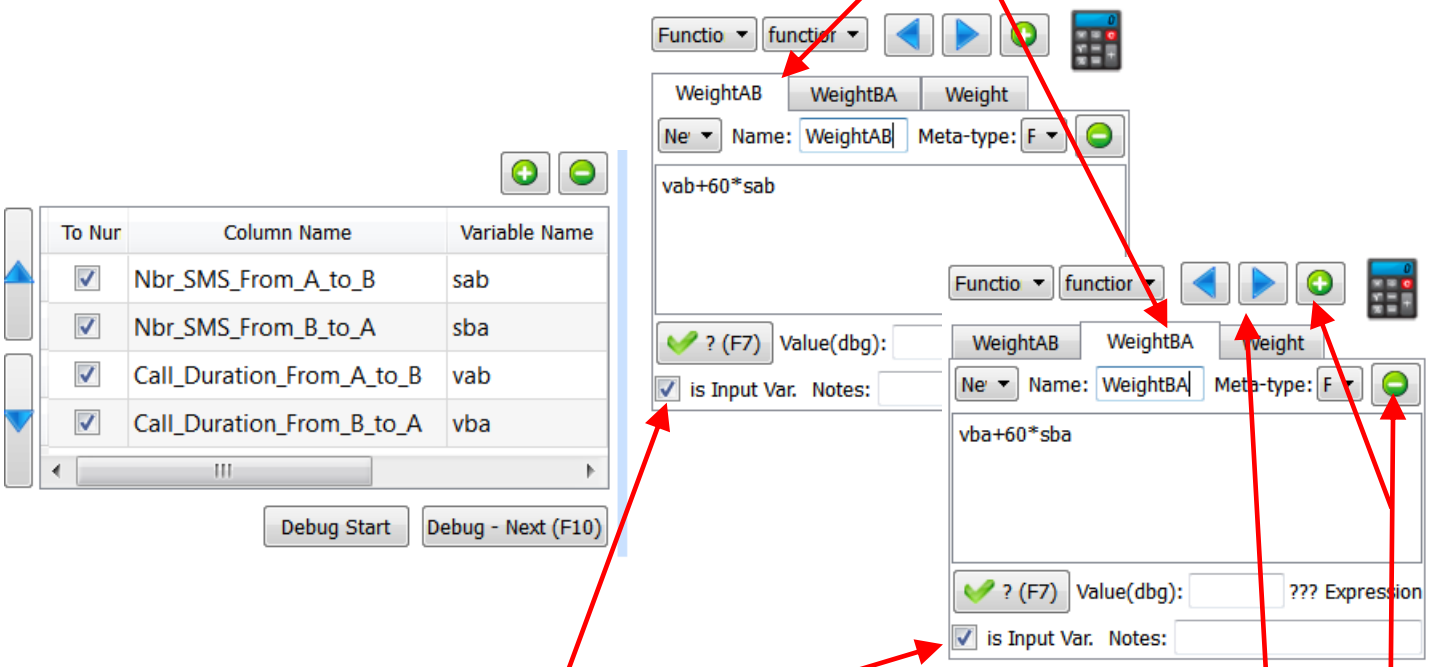
5.5.6.4. Computing many columns

You can compute many different (new) columns inside the same Calculator Action. For example, here is an example that computes the column "weight" that is the "*weight of the connection between 2 individuals (A and B) inside a telecommunication network*".



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

As you can see in the screenshot above, the column weight is the arithmetic mean of the 2 variables "Weight**AB**" and "Weight**BA**". These variables are defined on the other tabs inside the Calculator Action:



To be able to use the variables "Weight**AB**" and "Weight**BA**" inside the expression of the "Weight", we must enable these 2 checkboxes:

More precisely, if you want to use a column X inside the expression of the column Y, you need to:

a) Enable the "*is Input Var*" checkbox inside the tab of the column X.
b) Move the tab of the column X on the **left-side** of the tab of the column Y. (because the different expressions are evaluated in the order from the left to the right).

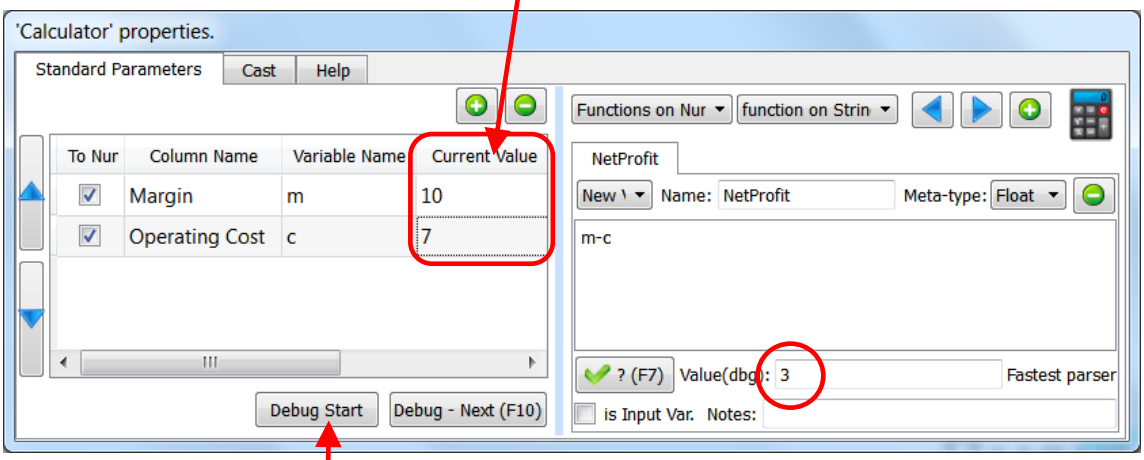
Use the 2 arrows to re-order the tabs.

Use the and the buttons to add/remove tabs.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.6.5. Debugging complex expressions

As explained previously, you can click the ? (F7) button to test your expression using the different values stored inside the "Current Values" column:



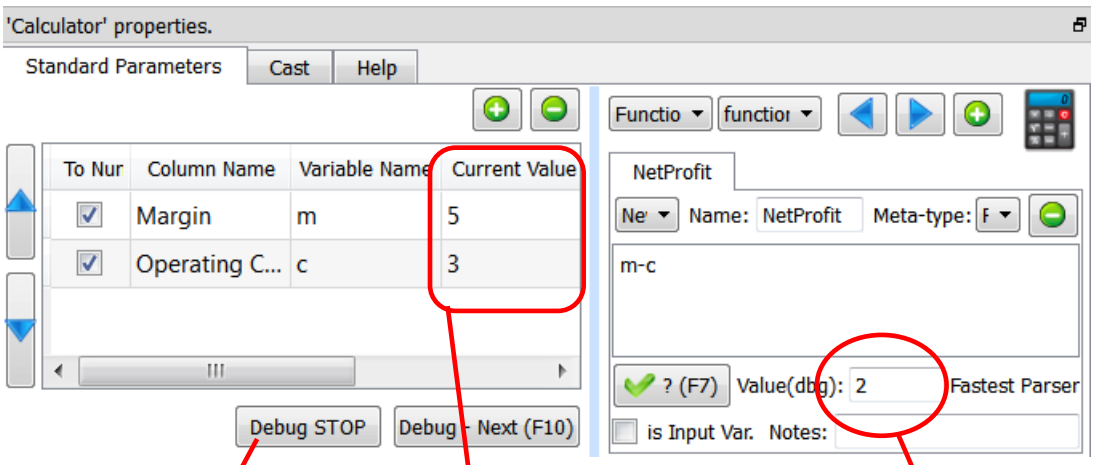
When you click the "Debug Start" button:

a) Anatella starts reading the input table of the Calculator Action row-by-row.
(To go to the next row, click the "Debug – Next(F10)" button).

b) Anatella fills-in the "Current Values" column with the values found on the "current" input row.

c) Anatella evaluates all the expressions (one expression per tab) of the Calculator Action and displays all the results in the different "Value(dbg)" fields (as usual).

For example:



Click here to stop the debugging session.

These values are coming from the input table of the Calculator Action. Each time you click the "Debug-Next" button, these values are changed to the values captured on the "next" row.

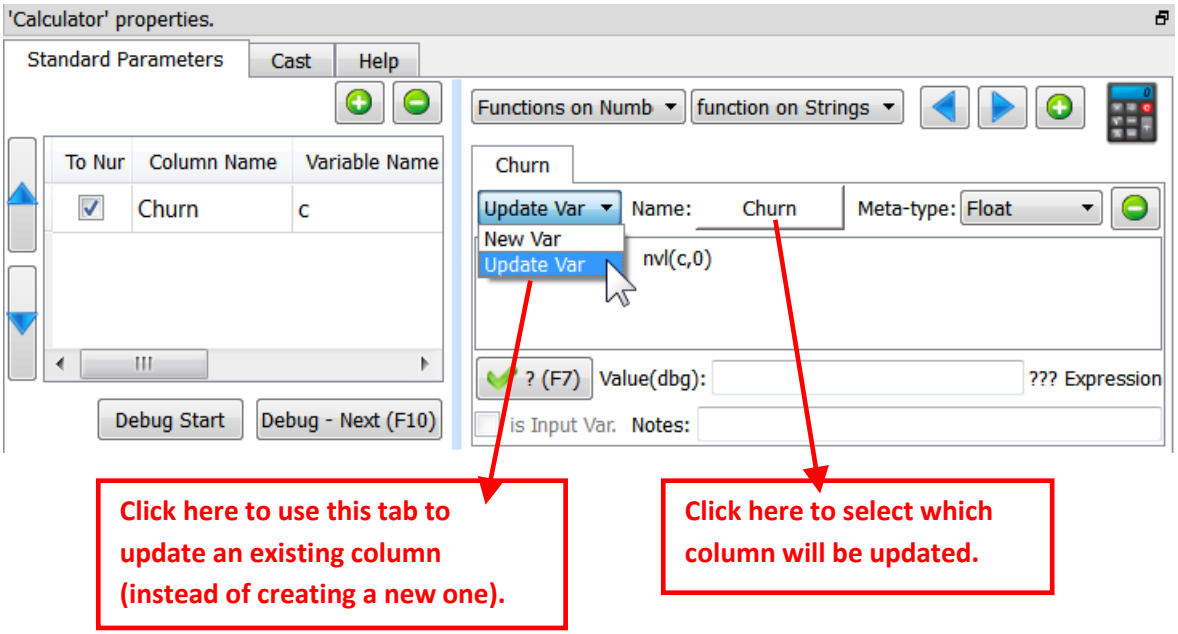
This is updated each time you click the "Debug-Next" button.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

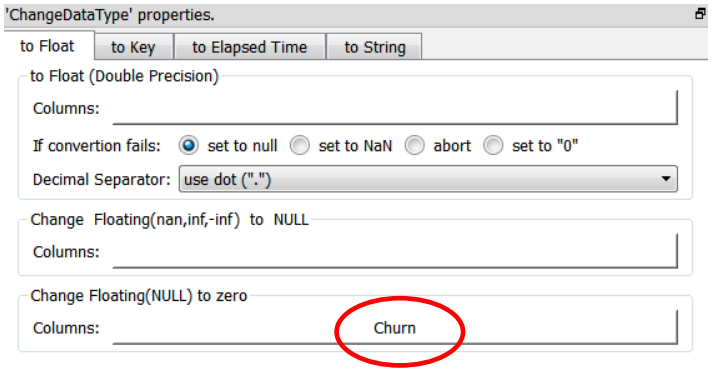
5.5.6.6. Updating columns

By default, each tab of the Calculator Action is defining a new column inside the output table of the Action. You can also use a tab to UPDATE an existing column (instead of creating a new one): The following example updates the column "Churn":

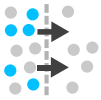
a) It replaces all the NULL values with the number "0" (using the "*nvl()*" function).
b) It changes the Meta-Type of the "Churn" column to the Float Meta-Type.



Note that we could have also used the ChangeDataType Action with the following parameters (and this second technique is faster):

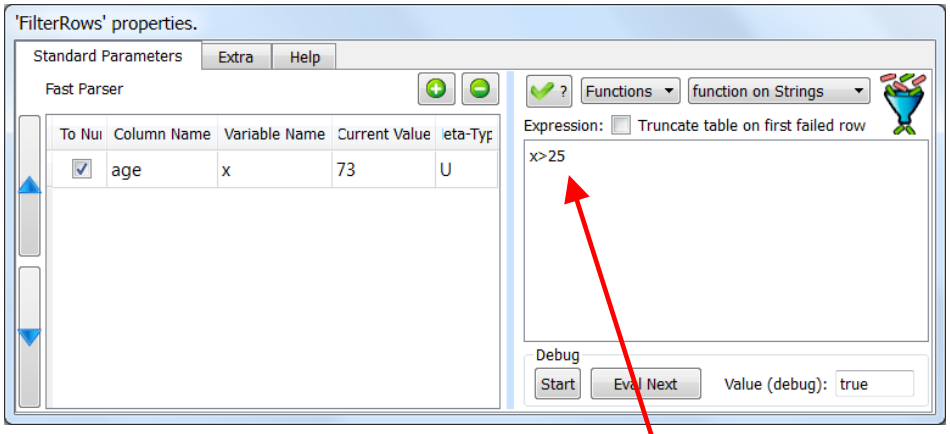


5.5.7. Filter Rows (High-Speed action)

Icon: 

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Property window:



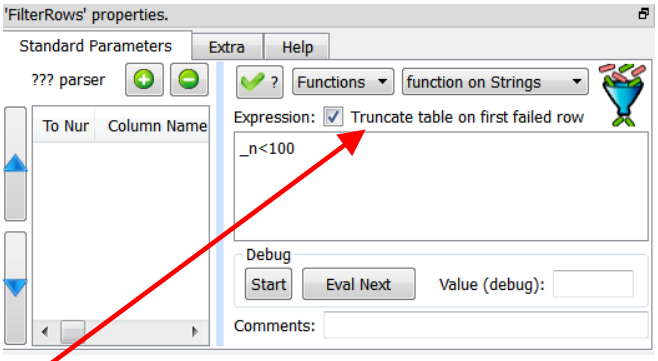
Short description:
Filters rows.

Long Description:
This operator filters the row of the input table.
The first output pin contains all the rows that match the given test.
The second output pin contains the non-matching rows.

For example: If we want to select all the people with an age above 25, we'll have:

The expressions given inside FilterRows Action follows the same syntax as the expressions from the Calculator Action: Please refer to the previous section about the Calculator Action for more information about the usage of the The FilterRows Action.

One very common usage of the FilterRows Action is to use it to create a sample. Samples are very useful to reduce the computing-time when developing a new script: We develop new data-transformation-graphs on small samples and, once they are working ok, we run the data-transformation-graphs on the whole dataset. For example: If we want to select the first 100 rows of the input table:



(The variable "_n" is the "row number"). Please note that we checked the option "*Truncate table on first failed row*": When this option is checked, Anatella will stop reading the input table at the first row where the expression value is false. When this option is active, the 2 expressions "*_n<100*" and "*(_n<100)||(_n>200)*" will both select only the first 100 rows of the input table. The main objective of this option is to "give a hint" to Anatella in order to reduce the computing time (we can safely check this option when evaluating the expression "*_n< 100*", because it's totally useless to read more than the first 100 rows).

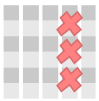
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When the "*Truncate table on first failed row*" option is enabled the table on the second output pin of the rowFilter Action is always empty.

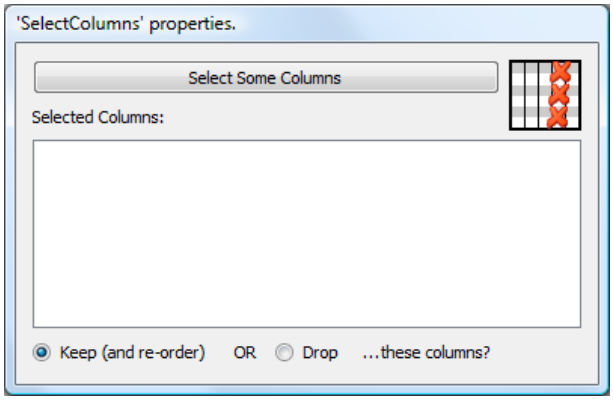
When the expression is "***_n< [number]***", then Anatella automatically checks/enables for you the "*Truncate table on first failed row*" option (because 99% of the time, this is what you really want, even if you forgot to check it).

This also means that, when the expression is "***_n< [number]***", you'll always obtain an empty table table on the second output pin of the rowFilter Action. If you want a non-empty table on the second pin, re-write you expression in the following way: "***[number]>_n***".

5.5.8. Column filter (High-Speed action)

Icon: 

Property window:

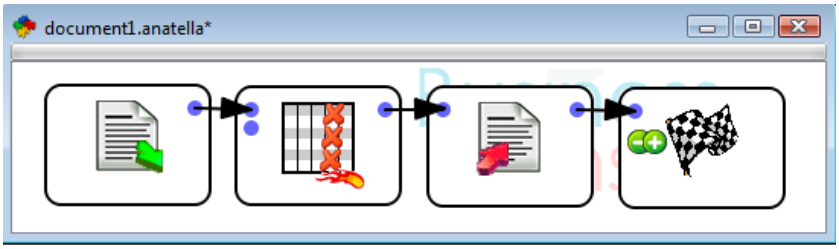


Short description:
Filters out some columns of the input table.

Long Description:
Filters out some columns of the input table.

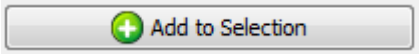
You can also use this Action to re-order the column of your table. For example, let's say that we want to "move" the column "revenu" as the first column on the left of the Table. We will have the following Anatella-Graph:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

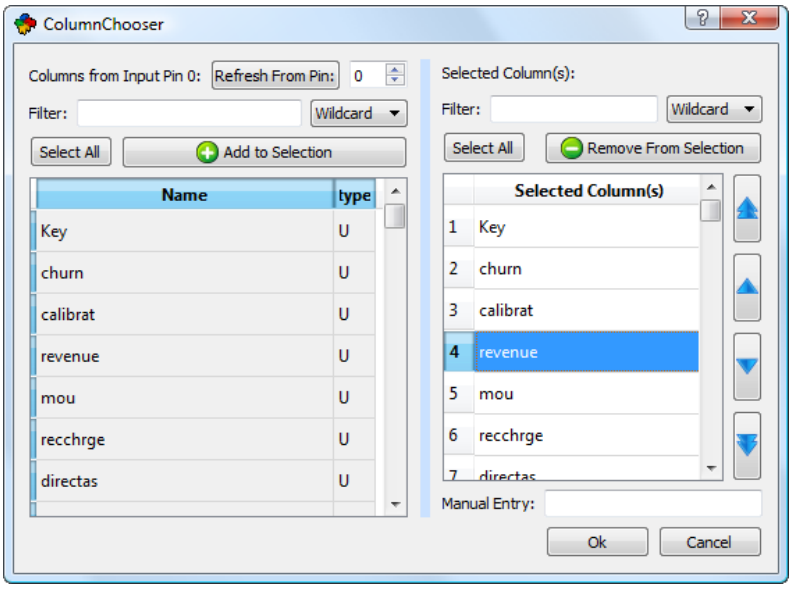


Double-click the "Column Filter " Action to open its properties window. Click the "Select Some Columns" button: The standard "Column Chooser" window opens (see section 5.1.4. about the "Column Chooser" window). In this window, Click the

button, click on the

button in the LEFT pane, click the 

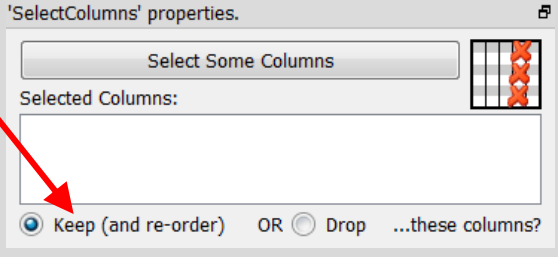
button. In the RIGHT pane, click on the "revenue" column name. You should now have:



Click on the button to "move" the "revenue" column as the first column in the list. You can re-order the columns of the output table in anyway you want, using the buttons.

About re-ordering the columns

You must be in "Keep" mode to be able to use the column re-ordering functionalities of this action.



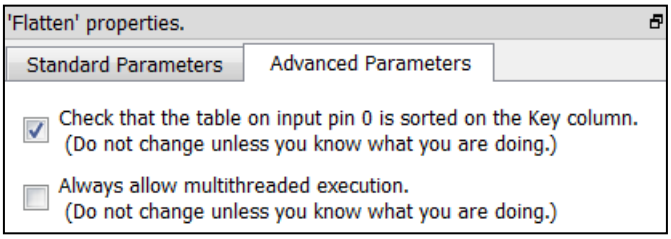
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

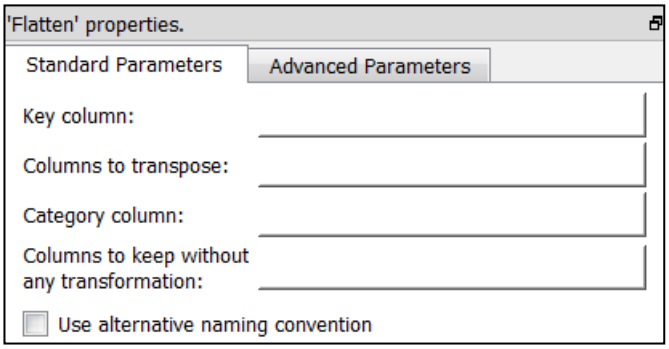
Optionally, you can connect the second input pin of the "Column Filter Action" to a table that contains column's names (one name per row, only the first column is used). These column's names will be added to the already selected columns.

5.5.9. Flatten (High-Speed action)

Icon:

Property window:





Short description:
Flatten the input table.

Long Description:
One row of the output table regroups several rows of the input table.
This makes the output table "flatter" than the "input table" (thus the name: "flatten").

For example: We want to transform this table (where each row is a day):

Week	DayOfWeek	TotalSales	Count
1	Monday	100	10
1	Tuesday	101	11
1	Wednesday	102	12
1	Thursday	103	13
1	Friday	104	14
1	Sunday	106	16
2	Monday	107	17
2	Tuesday	108	18
2	Wednesday	109	19
2	Thursday	110	20
2	Friday	111	21
2	Saturday	112	22
2	Sunday	113	23
3	Monday	114	24
3	Tuesday	115	25
3	Wednesday	116	26
3	Thursday	117	27
3	Friday	118	28
3	Saturday	119	29
3	Sunday	120	30

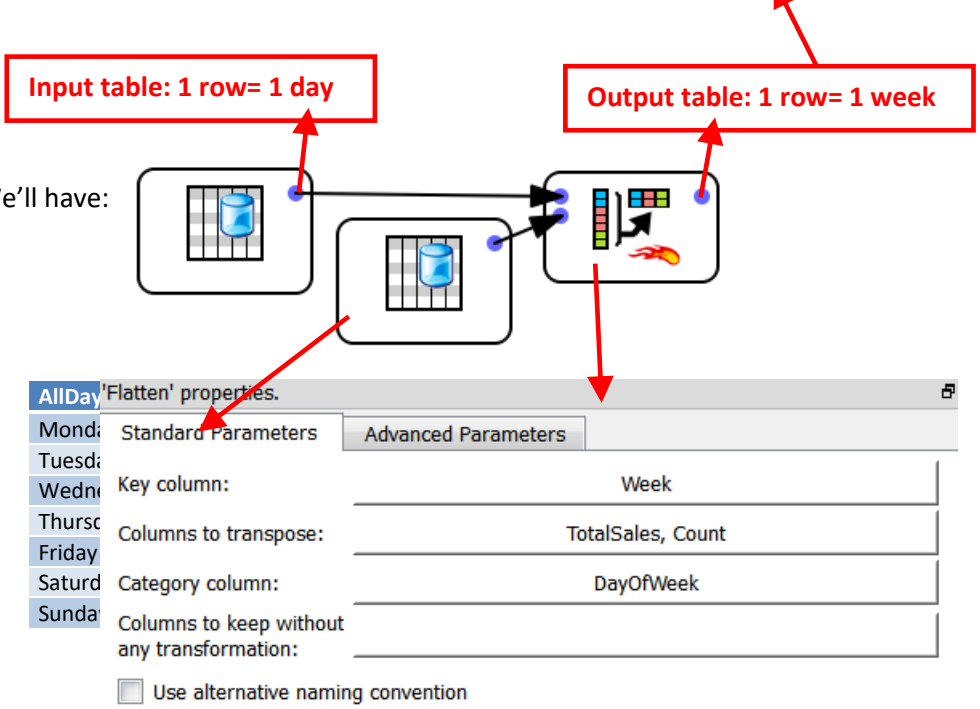
The "Saturday" row is missing in the input table. The output table contains a NULL for the corresponding cells.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

… into this one (where each row is a week):

Week	Monday_ TotalSales	Monday_ Count	Tuesday_ TotalSales	Tuesday_ Count	Wednesday_ TotalSales	Wednesday_ Count	Thursday_ TotalSales	Thursday_ Count	Friday_ TotalSales	Friday_ Count	Saturday_ TotalSales	Saturday_ Count	Sunday_ TotalSales	Sunday_ Count
1	100	10	101	11	102	12	103	13	104	14			106	16
2	107	17	108	18	109	19	110	20	111	21	112	22	113	23
3	114	24	115	25	116	26	117	27	118	28	119	29	120	30

We'll have:

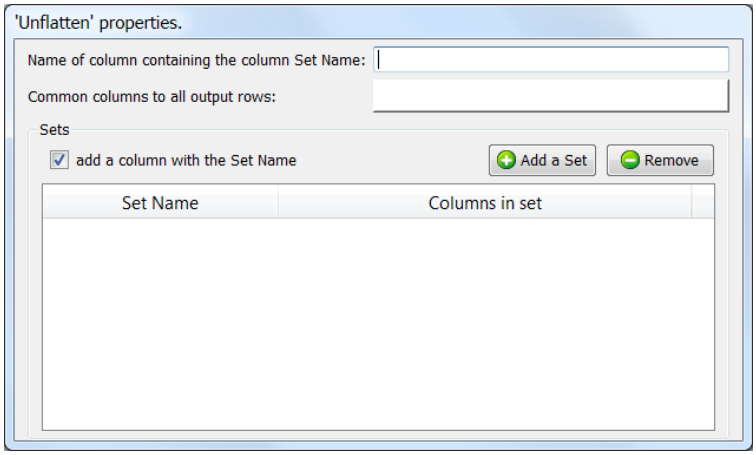


You can run the Flatten Action inside a N-Way multithreaded Section if the "partitioning" parameter of the Multithread Action is equal to the "Key Column" of the the Flatten Action: see section 5.3.2.5. for more information about this subject.

5.5.10. Unflatten (High-Speed action)

Icon:

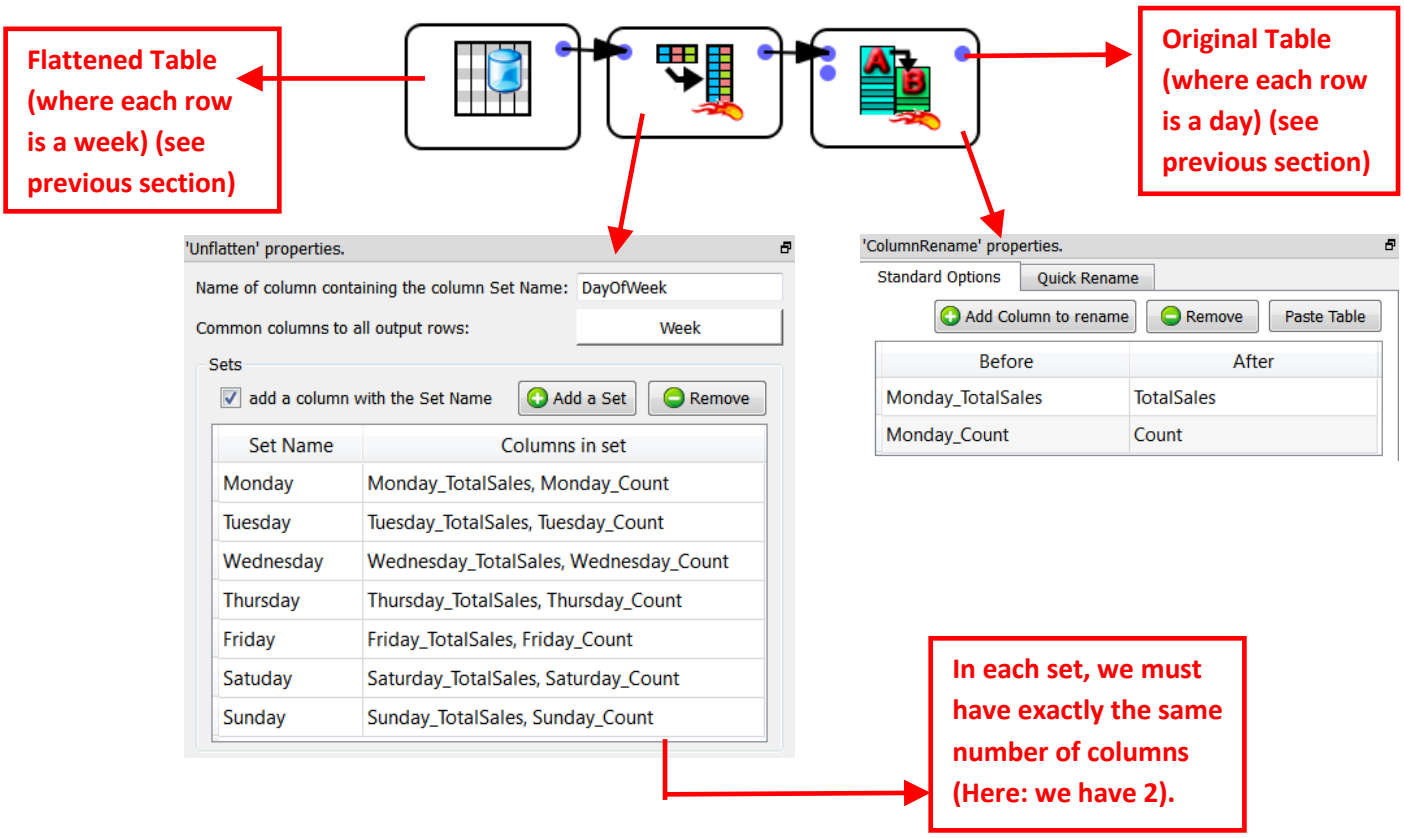
Property window:



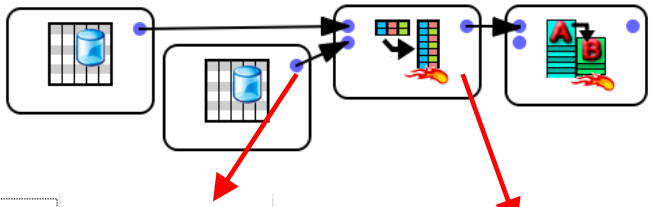
Short description:
Flatten the input table.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

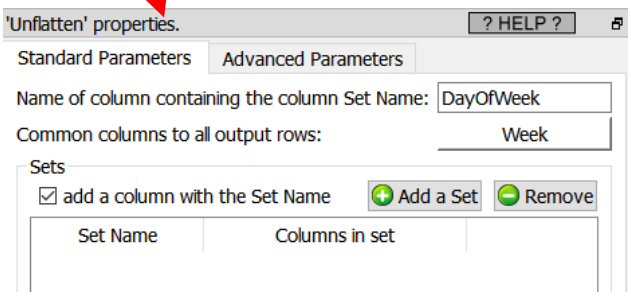
Long Description:
One row of the input table is used to create several rows of the output table.
For example: We want to revert the "flatten" Action described in the previous section. We'll have:



The second input pin allows you to use a dynamic "Sets" table. For example, the above graph is equivalent to:



Monday	Monday_TotalSales
Monday	Monday_Count
Tuesday	Tuesday_TotalSales
Tuesday	Tuesday_Count
Wednesday	Wednesday_TotalSales
Wednesday	Wednesday_Count
Thursday	Thursday_TotalSales
Thursday	Thursday_Count
Friday	Friday_TotalSales
Friday	Friday_Count
Satuday	Saturday_TotalSales
Satuday	Saturday_Count
Sunday	Sunday_TotalSales
Sunday	Sunday_Count

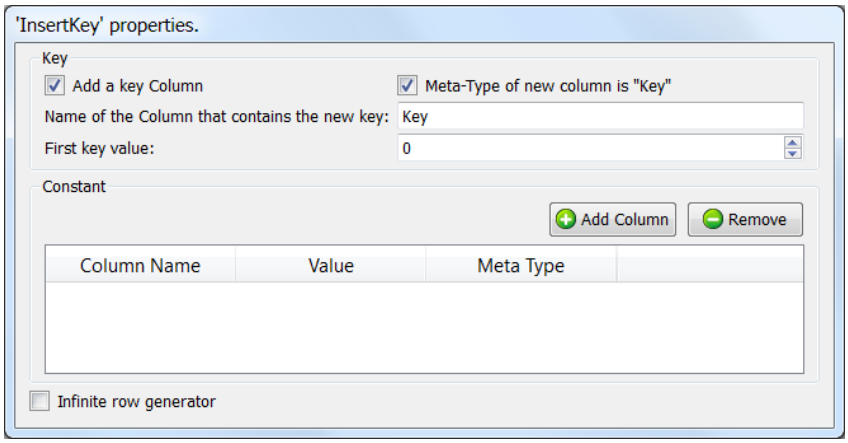


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.11. Insert Key (High-Speed action)

Icon:

Property window:



Short description:
Adds a key column and/or a constant column.

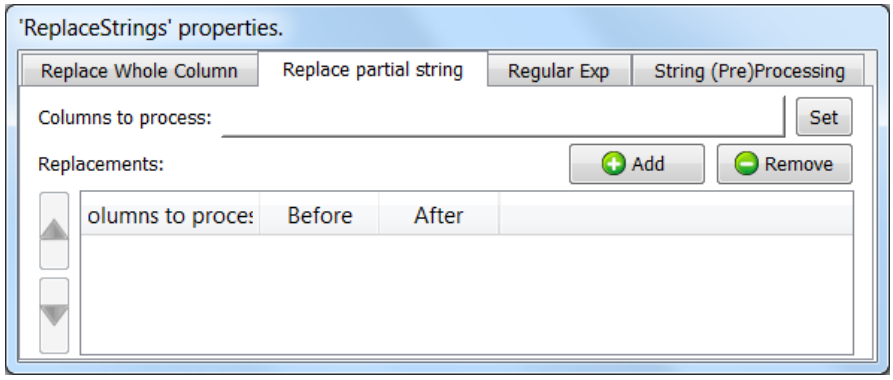
Long Description:
Adds one new column that contains a sequence of increasing number (i.e. a "Key" Column).
Adds some new columns that each contain a constant value (i.e. a "Constant" Column)

To create the "Key" column, you could also use the Calculator Action (use the variable "_n") but this one is faster. To create the "Constant" columns, you could also use the Calculator Action but this one is faster.

5.5.12. Replace String (High-Speed action)

Icon:

Property window:



Short description:
Replace a given string by another into some selected columns.

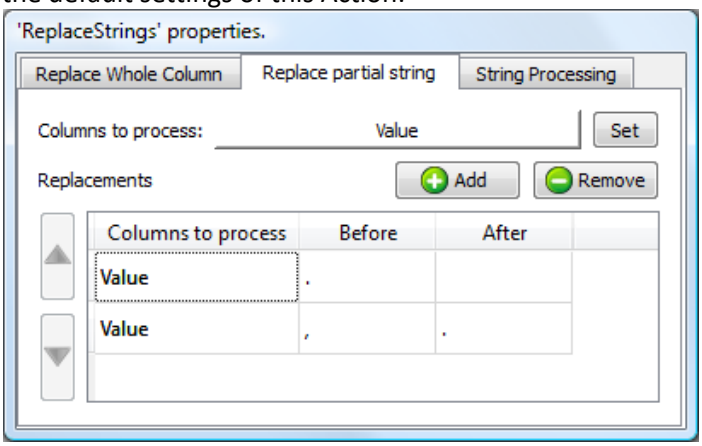
Long Description:
This operator searches for some given strings into some selected columns. If there is a "match", then the "found" string is replaced by another one (specified inside the "After" field).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

There are three different ways to compute the “matches”:

a) String-Replace Partial String.

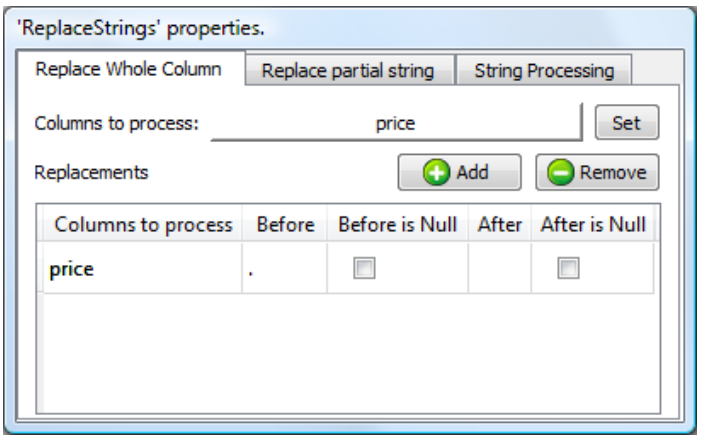
For example, the default settings of this Action:



… will process the column “Value” and transform the string “1,000.9” into this string: “1000.9” (i.e. it transforms a number from the “French notation” to the more common“English notation” used everywhere into Anatella). This is not a very good example because the changeDataType Action is usually the best way to perform such transformation.

b) String-Replace Whole Column.

For example, these settings:



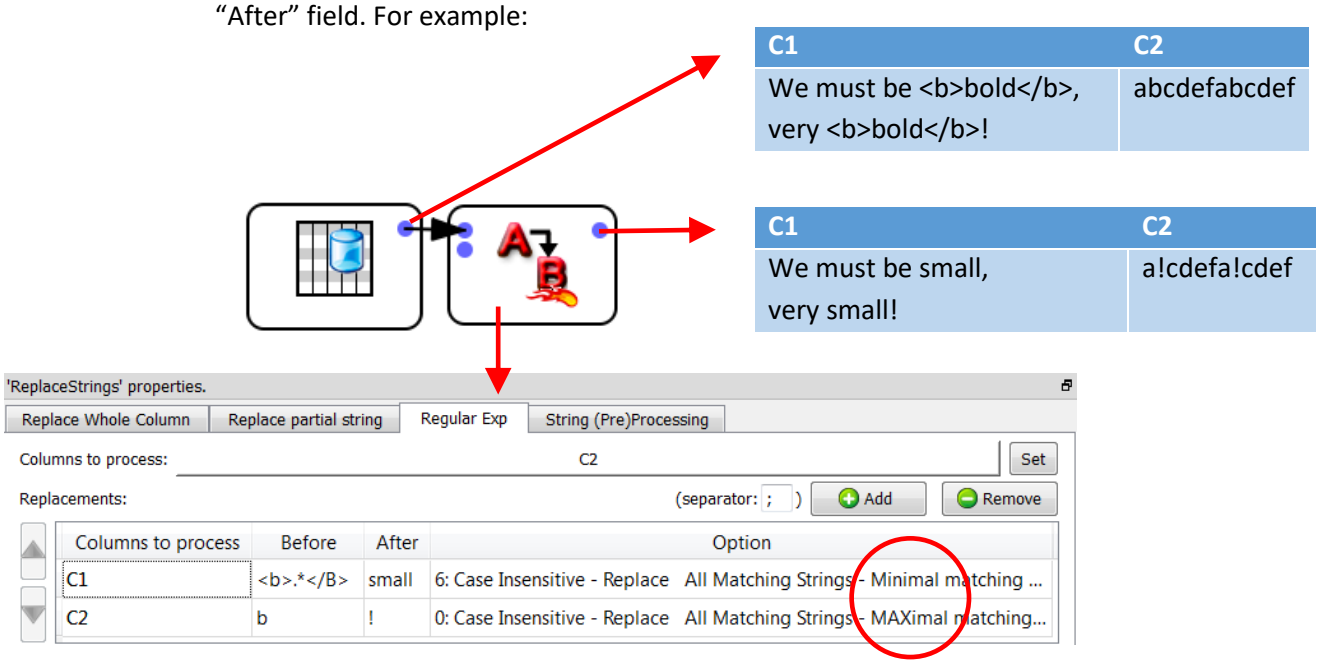
… will transform a column containing the “.” string (that represents the “missing value” inside the SAS system) into a column containing a string of zero-length (which usually represents the “missing” value in many systems). Note that the string “1,000.9” will NOT be changed (it won’t be replaced by “1,1009”) because there is a “match” ONLY IF the whole column matches the given string.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

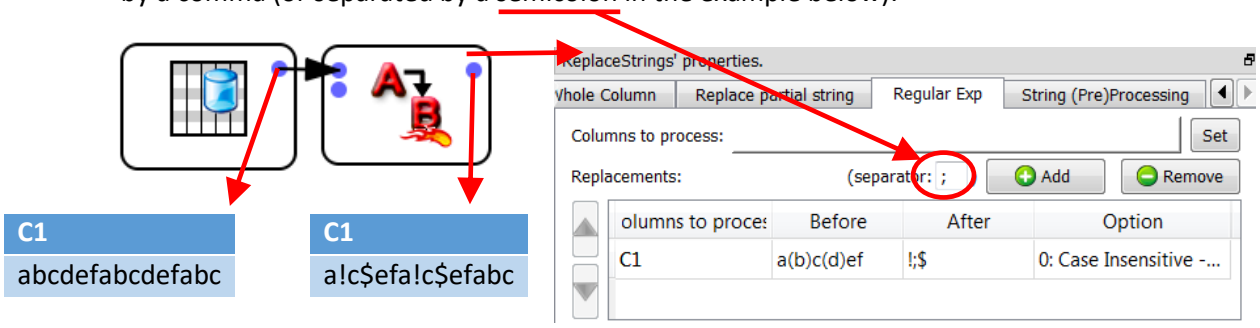
c) String-Replace based on Regular expression.

There are 2 operating modes when working with regular expressions:

I. You regular expressions do not contain any capturing parenthesis.
Anatella replace the whole string matching your regular expression with the (unique) "After" field. For example:



II. You regular expressions contain capturing parenthesis.
You must give a different "After" string for each different capturing parenthesis. By default, the many different "After" strings are written in the "After" field, separated by a comma (or separated by a semicolon in the example below):



The many "Regular expression" options (24 in total) are all the different combinations of the following basic options:

I. Case Sensitive/Case insensitive (self-explanatory)
II. Replace All Matching Strings / Replace First Matching String (self-explanatory)
III. Maximal Matching / Minimal Matching.
Let's assume that we are searching for the regular expression "*.**" inside the string "*We must be bold, very bold!*". Each match must be replaced with "*small*". We obtain:
a. For Minimal Matching: "We must be small, very small!"
b. For Maximal Matching: "We must be small!"
This option has no effect on how the capturing parenthesis are working.

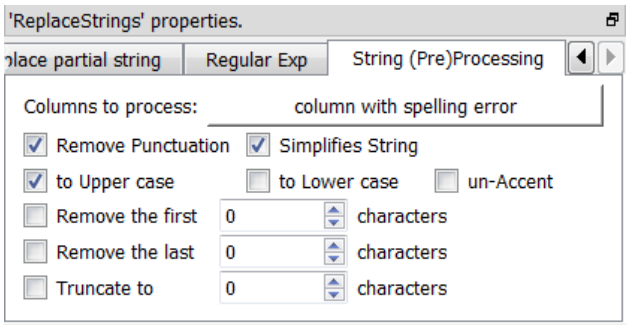
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

IV. Perl Like / Perl Like with greedy captures / Reg.Exp. as defined by W3C.
Regular expression syntax is slightly different inside Perl than inside a web page. With this option, you can:
   a) select the desired syntax (Perl or Web).
   b) select how the capturing parenthesis are working: For historical reasons, quantifiers (e.g. *****) that apply to capturing parentheses are more "greedy" than other quantifiers. For example, in the Regular Expression "**ba*(a*)b",** we want to replace all capturing parenthesis with the "**!**" character. The Regular Expression "**a*(a*)"** will, of course, match "**baaab**" but we obtain, as output:
      - without greedy capture (i.e. perl-like): "**baaab**"
      - with greedy capture: "**b!b**"

The optional table on the second input pin of this Action is also used to do "whole column" content replacement. It contains 3 columns:
1. The name of the column to process
2. The string to search for (before)
3. The replacement string (after)

Usually, the second input pin is attached to an "Inline-Table" that contains a list of replacements to make to correct spelling mistakes.

The last tab of the property window of this parameter describes some string processing that you can activate on specific columns. These processings are performed BEFORE any attempt to replace anything. In particular, when you are using the "String Replace" action with the "Correct Spelling" Action of Anatella, it's strongly suggested to always activate the following string-processing:



These String-Processings are performed in a top-to-bottom order (referring to the order of the controls inside the property window of the "String Replace" action). Here is a small explanation of these String-Processings:

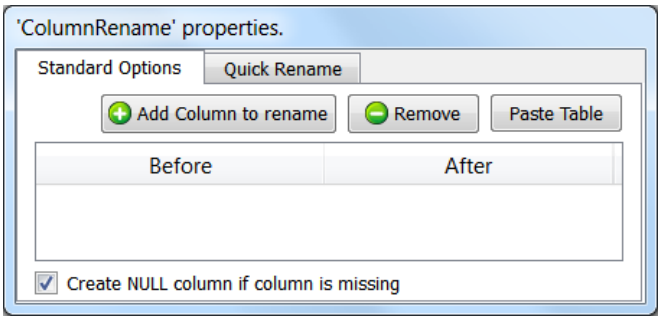
a) Remove Punctuation: replace the character dot, comma, exclamation mark, quotation mark, semi-column, double-dot with the blank (' ') character.
b) Simplifies String: remove any un-necessary blank character. For example the string " AB C D " simplifies to "AB C D".
c) To Upper Case: self-explanatory
d) To Lower Case: self-explanatory
e) Un-Accent Case: self-explanatory
f) Remove the first 'n' characters: self-explanatory
g) Remove the last 'n' characters: self-explanatory
h) Truncate to 'n' characters: self-explanatory

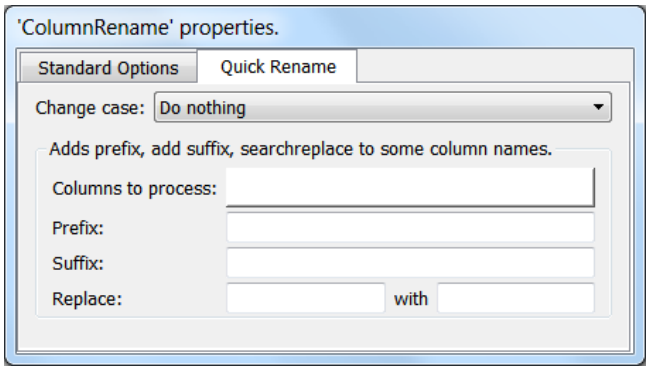
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.5.13. Column Rename (High-Speed action)

Icon:

Property window:





Short description:
Rename some columns.

Long Description:
This operator renames some columns.

Optionally, you can connect on the second pin of the "Rename Column Action" a table that contains 2 columns:

1. First column: the name of the column that will be re-named (the "before" column).
2. Second column: the new name after re-naming (the "after" column).

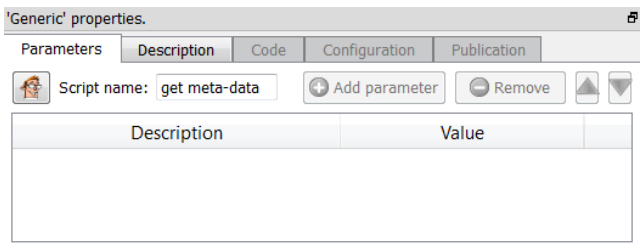
One easy way to create this "renaming" table is to modify slightly the ouput table of the Get Meta-Data Action (see next section).

The "change case" option is particularly useful when you have a datasource that is case insensitive (such as SAS datasets or SQL tables). Let's assume that you developed a Data-Transformation-Graph that reads a text file and process it. Each week, you get a new text file and you run your Anatella-Graph. Let's also assume that the column names inside this text file are "A" and "B". It can happen that, on a given week, the column names inside the text file change suddenly to "a" and "b" (because the tool that generates the text file is case insensitive and thus the case does not matter for him. A good example is SAS or any RDBMS). This has no consequence on case insensitive systems but, since Anatella is case-sensitive, it can' find anymore the required columns (i.e. the columns "A" and "B") and it fails. To avoid such failure, you can use the "*Change case*" option: Select in the combo-box: "*All Columns to Uppercase*".

5.5.14. Get Meta-Data

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Get the Columns Meta-Data.

Long Description:

For example:

	A	B	C
1	a	3.14	42

	Before	After	Type
1	A	A	U
2	B	B	F
3	C	C	K

The output-table contains 3 columns: The first and second column contains the column names of the input table (to make it easy to use the Get Meta-Data Action together with the Column Rename Action). The third column contains the data-type:

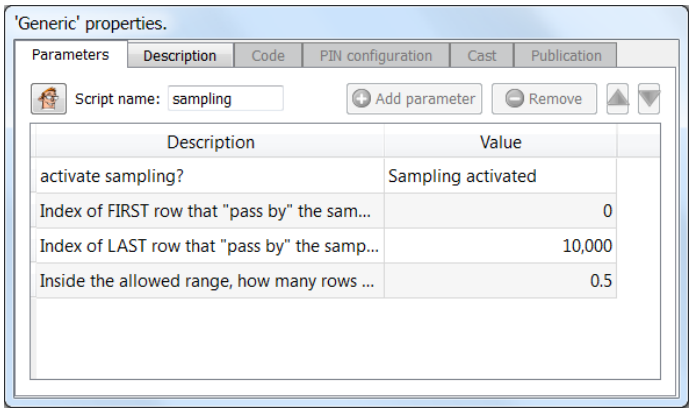
a) F: Floating-point Data type.
b) K: Key Data Type.
c) U: Unknown/String Data Type.

See section 5.1.2. to know more about Data Types.

5.5.15. Sampling

Icon:

Property window:



Short description:
Create a Sample.

Long Description:

With the default parameters, the Sampling Action outputs about 50% (0.5) of the first 10.000 rows of the input table. The 5.000 rows (out of the 10.000 rows) are selected randomly and change at each run.

This action is useful during development of a new data-transformation graph.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Sometime, the running-time when you click an output pin of an Action can be quite long (especially when it involves running a

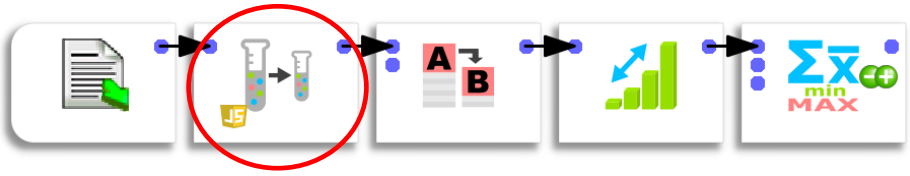
Sort Action that can potentially run for several hours). To get quicker results, you can:

1. Create a Hard Drive Cache **after** the a

Sort Action (it will prevent running the "sort" all the time: That's good): To do so, click the output pin of the

Sort Action and wait a little.

2. Work on a sample for the time required to develop the graph. For example, insert a Sampling Action here:



With the default settings, the sampling action returns about 5000 rows randomly selected amongst the first 10000 row of the input table (These 5000 rows are changing at each run because of the random component). Now, when you click the output pin of the aggregate action, Anatella instantaneously displays the results (because sorting a 5000 row table is almost instantaneous).

Once the design of your new graph is complete, remove the Sampling Action completely: Leaving it "in-place" costs a large amount of CPU time because it's written in Javascript and it's thus not as fast as a standard C++ Action.

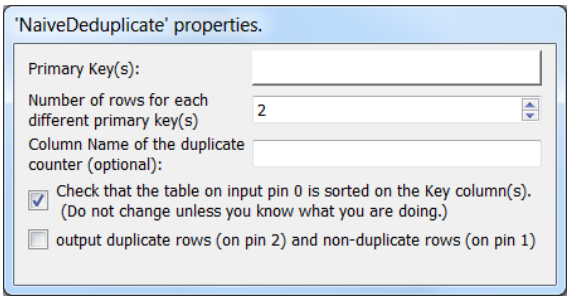
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.6. TA – Cleaning (TA=Transformations Actions)

5.6.1. Naïve Deduplicate (High-Speed action)

Icon:

Property window:



Short description:
Remove Duplicate rows

Long Description:
This action remove duplicates rows.
Two rows are considered as "duplicate" when all their primary key(s) match.
The input table of this action must be sorted on its primary key(s).

Only the primary key(s) are checked to search for duplicates, it means that two rows that have :

- different values on NON-primary columns.
- the same primary key(s).

… will still be “de-duplicated” (to only keep one row).

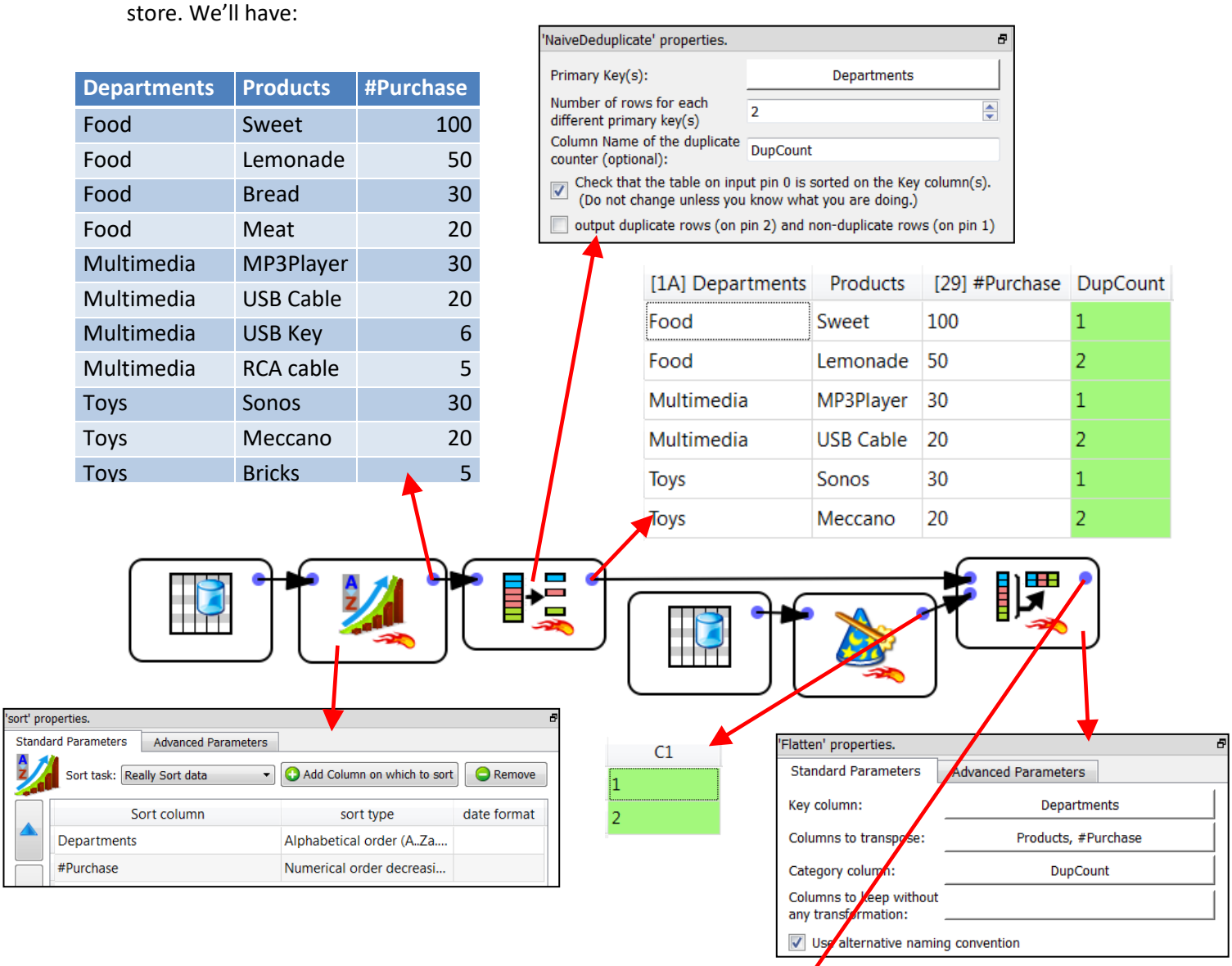
Pre-requisite
The input table of this action must be sorted on its primary key(s).

The NaïveDeDuplicate Action is very often combined with the Flatten Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example, let's say that we want to know the 2 most purchased products in each department of the store. We'll have:

Departments	Products	#Purchase
Food	Sweet	100
Food	Lemonade	50
Food	Bread	30
Food	Meat	20
Multimedia	MP3Player	30
Multimedia	USB Cable	20
Multimedia	USB Key	6
Multimedia	RCA cable	5
Toys	Sonos	30
Toys	Meccano	20
Toys	Bricks	5



As result, we'll get:

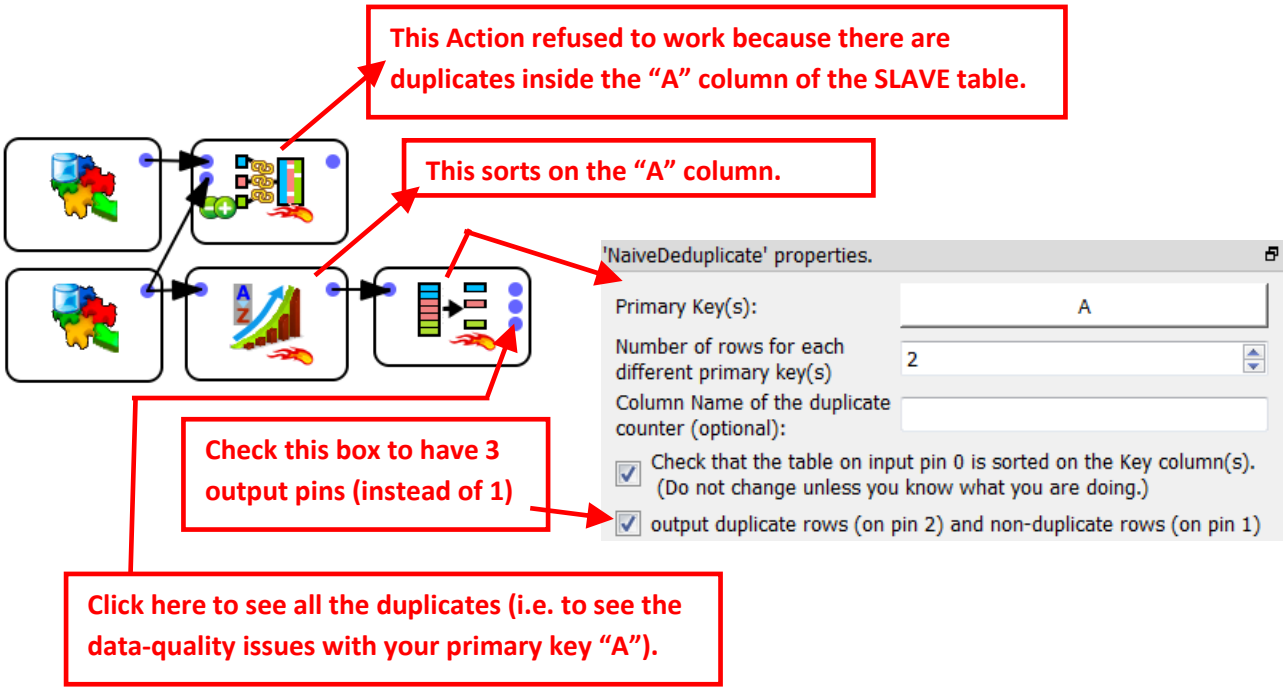
[1A] Departments	Products_1	#Purchase_1	Products_2	#Purchase_2
Food	Sweet	100	Lemonade	50
Multimedia	MP3Player	30	USB Cable	20
Toys	Sonos	30	Meccano	20

E.g. For the "Food" d 00 sales) and "Lemonade" (with 50 sales).

The NaïveDeDuplicate Action is also very often used to investigate data quality issues initially detected with the MultiJoin Action or the FilterOnKey Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

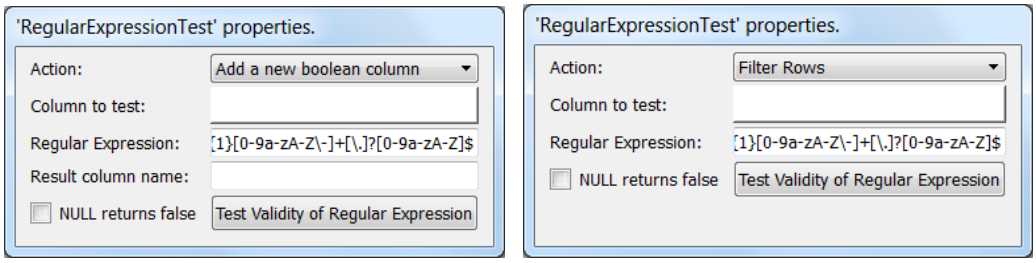
These 2 actions requires no duplicates values inside the "key" column of the SLAVE tables. When a key-duplication is detected, you can see the duplicates in the following way:



5.6.2. Regular Expression Test (High-Speed action)

Icon:

Property window:



Short description:
Check columns using a regular expression.

Long Description:
This action checks the given columns to see if they match a specific pattern that is given as a standard regular expression.

The default test for the Regular Expression Action checks if the selected column contains a valid e-mail address. This test uses a regular expression to check of the columns contains a well-formed e-mail address:

```
^[0-9a-zA-Z]+@[0-9a-zA-Z\-]+[\.]{1}[0-9a-zA-Z\-]+[\.]?[0-9a-zA-Z]+$
```

This demonstrates that you can easily use the powerful and very versatile "*Regular Expression syntax*" everywhere inside your Anatella graphs.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

There exist many books and references about regular expression. Here are some of them:

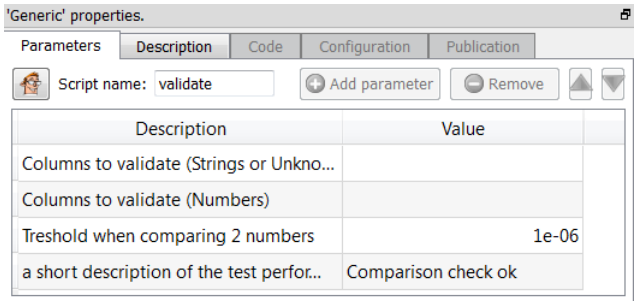
- Regular Expressions Cookbook by Jan Goyvaerts and Steven Levithan
- Teach Yourself Regular Expressions in 10 Minutes by Ben Forta
- Mastering Regular Expressions by Jeffrey Friedl

The easiest way to use the Regular Expression Action is to use some pre-defined regular expression. Regular expression can be used for a wide variety of usage. On the internet, you'll find large libraries of regular expressions: for example: This website: http://regexlib.com contains thousands of "pre-made" regular expressions that are classified in a wide variety of topics: email, url, numbers, strings, dates, times, address/phone, markup/code, etc.

5.6.3. Table validate

Icon:

Property window:



Short description:
Validate/Compares two tables

Long Description:
This operator compares the 2 tables given on the 2 input pins.
Anatella checks if the 2 tables are identical.
Only the selected columns will be compared.

The output of this operator is one unique row that contains:

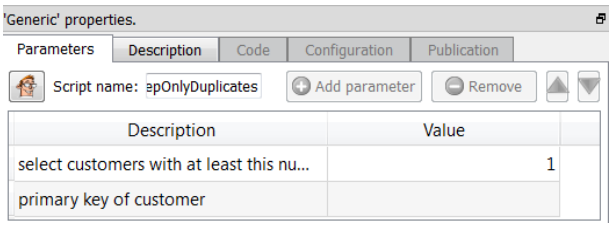
1. In the first column: the user-supplied "short description" of the performed test.
2. In the second column:
    - If the comparison is successful (i.e. the selected columns of the 2 tables matches), then you obtain, the string "OK".
    - If the comparison failed, then you obtain a short description of the "mismatch".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.6.4. Keep Only Duplicates

Icon:

Property window:



Short description:
Keep only duplicate rows.

Long Description:
Let's assume that:

- We have a Transaction-table that is ready to be transformed into a "Global-Customer-Level" table.
- we only want to analyze customers that have made 2 or more transactions.

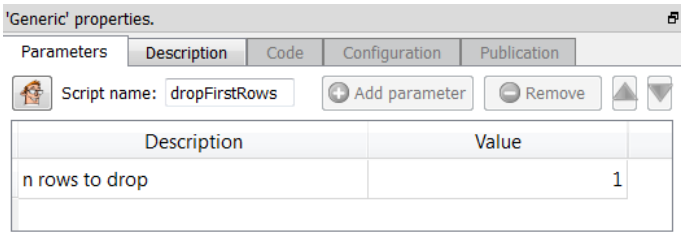
In other word, all the rows of the Transaction-table that are linked to a customer that made only one transaction must be removed from the Transaction-table. This Action allows you to do precisely that.

Pre-requisite
The input table of this action must be sorted on its primary key(s).

5.6.5. Drop First Rows

Icon:

Property window:



Short description:
Remove the first n rows

Long Description:
Self-Explanatory

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.6.6. Pattern Extractor

Icon:

Property window:

'Generic' properties.

Parameters | Description | Code | Configuration | Publication

Script name: patternExtractor | Add parameter | Remove

Description	Value
what to do?	Extract all patterns
column to analyze	
filtering pattern	

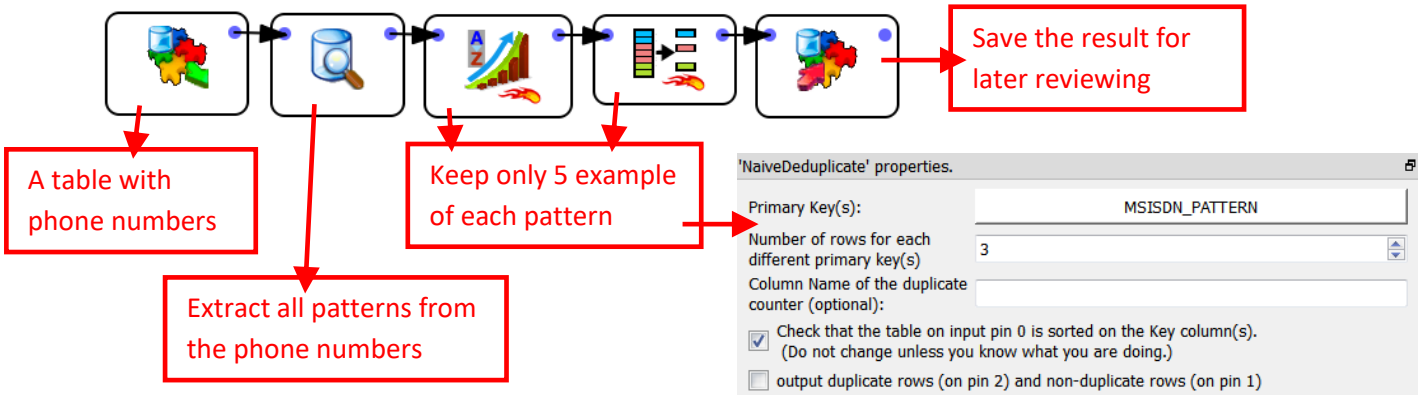
Short description:
Extract patterns out of phone numbers.

Long Description:
This analyzes a column containing phone numbers. It has 2 operating mode. The first operating mode is typically used to find some examples of different patterns in the phone numbers.

The rules to create the patterns are:

- All the digits have been replaced with “n”.
- All the letters have been replaced with “A”.
- All the punctuations are kept “in place”.

You typically use the first operating mode this way:



For example: Input table → Output table

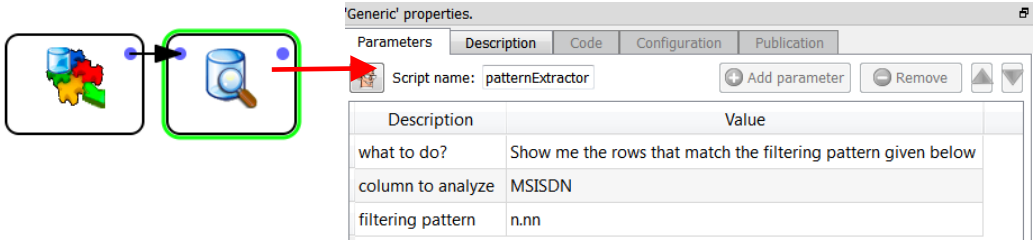
MSISDN
123
4HELLO
6.89
456
567
789
890
98
876

MSISDN	MSISDN_PATTERN
6.89	n.nn
4HELLO	nAAAAA
98	nn
123	nnn
456	nnn
789	nnn

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

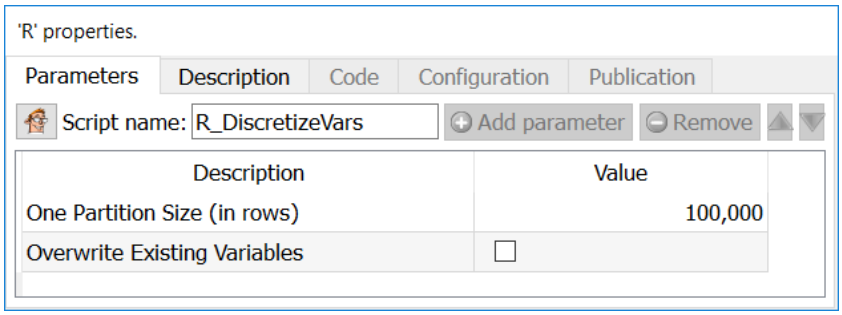
The output table contains 3 examples of phone number for each different pattern. Three examples is enough to understand if a pattern is representing some valid entries or not. Thereafter, you can take action to correct & clean the data.

Let's now assume that you are interested in a specific pattern "n.nn" and you want to have all examples following this specific pattern: You'll use:



5.6.7. Discretize Variables

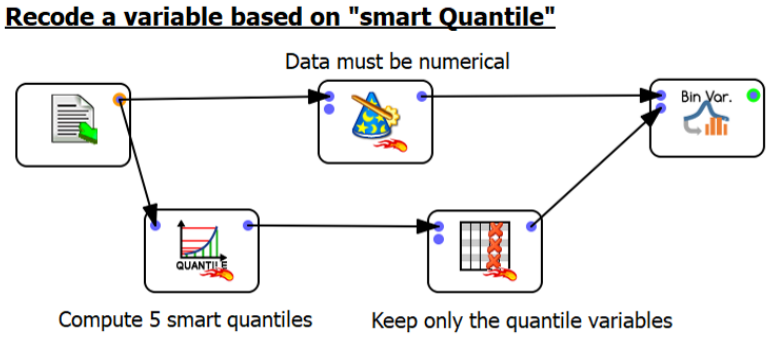
Icon:



Property window:

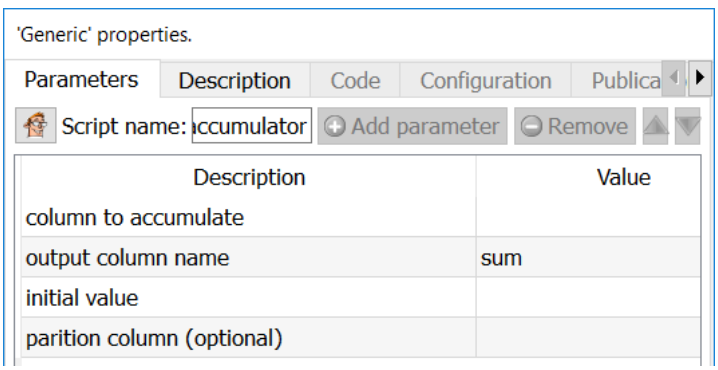
Short description :
Discretize a continuous variable

Long description :
This node allows to easily recode variables based on a quantile. It expects two inputs: a complete table with the (numerical and continuous) variables to recode, and the calculated "bins" or quantiles. As illustrated below, you only need the recoded variables, and the node will automatically recode the variables with a matching name.



5.6.8. Accumulator

Icon:

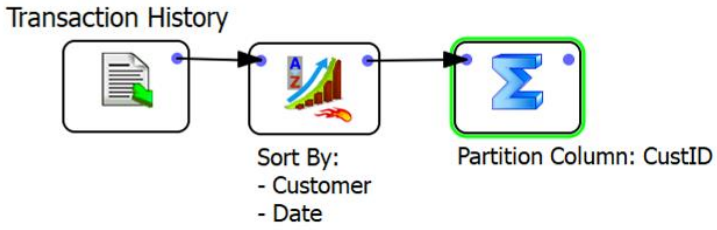


Property window:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Compute the accumulated sum by Row

Long description:
This node allows you to sum rows, optionally by partition. For example, in a transaction table it will compute the total sale per row, and reset every time there is a change of customer (if the partition column is selected)

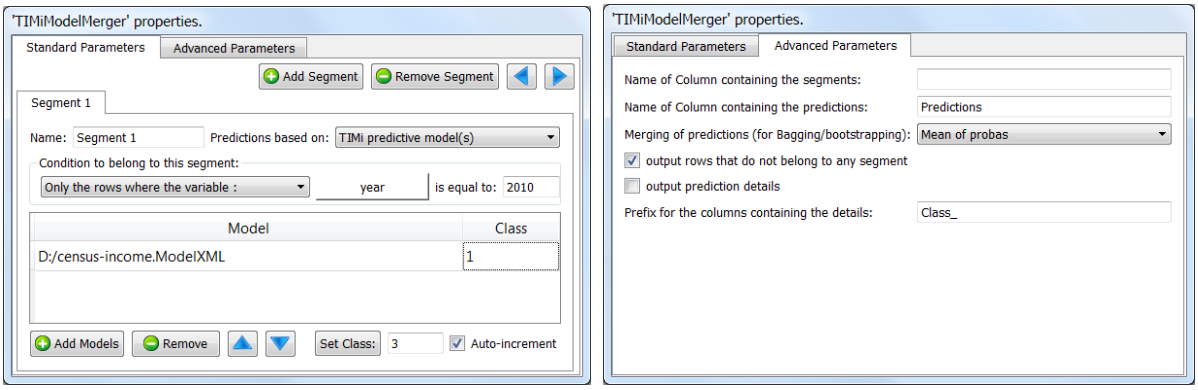


5.7. TA – Data Mining – Modeling (TA=Transformations Actions)

5.7.1. TIMi Use Models (High-Speed action)

Icon:

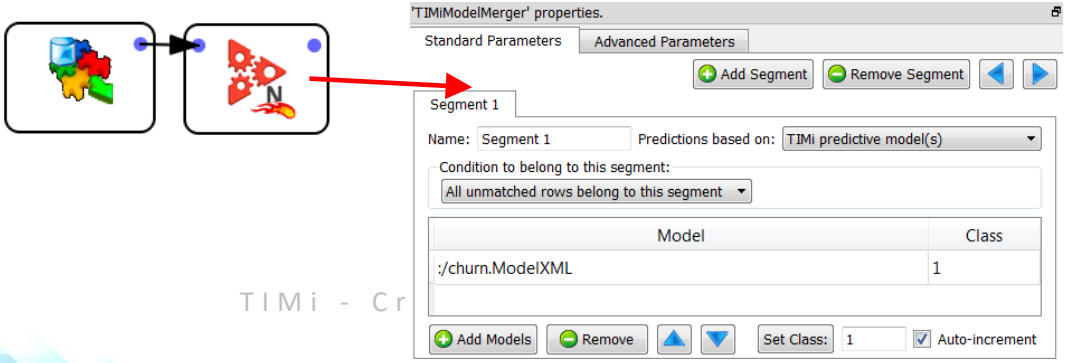
Property window:



Short description:
Apply many TIMi predictive models on each row of the input table.

Long Description:
This operator applies many TIMi predictive models on each row of the input table.

Let's assume that we developed a simple churn model (i.e. we have a Binary Classification problem or, in other words, a scoring/ranking). We want to use this churn model inside Anatella. We'll have:



Let's now assume that you are developing a "next-to-buy" solution: i.e. You must guess which product out of these 5 products: TEDDY BEAR, LIGHT SABER, IPOD, FLOWER, BEER your customer will buy next. In the scientific world, this is known as a Multi-Class prediction system (in opposition to the classical Binary-Class prediction system: Did the customer churned or not? Yes/No).

To do multi-class prediction with TIMi, you must first transform the multi-class prediction problem into a series of binary classification tasks. These binary classifiers can be of one of the two types:

- 1 vs all others
- 1 vs 1

To know more about this subject, see section 6 of the "TIMi Advanced Guide".

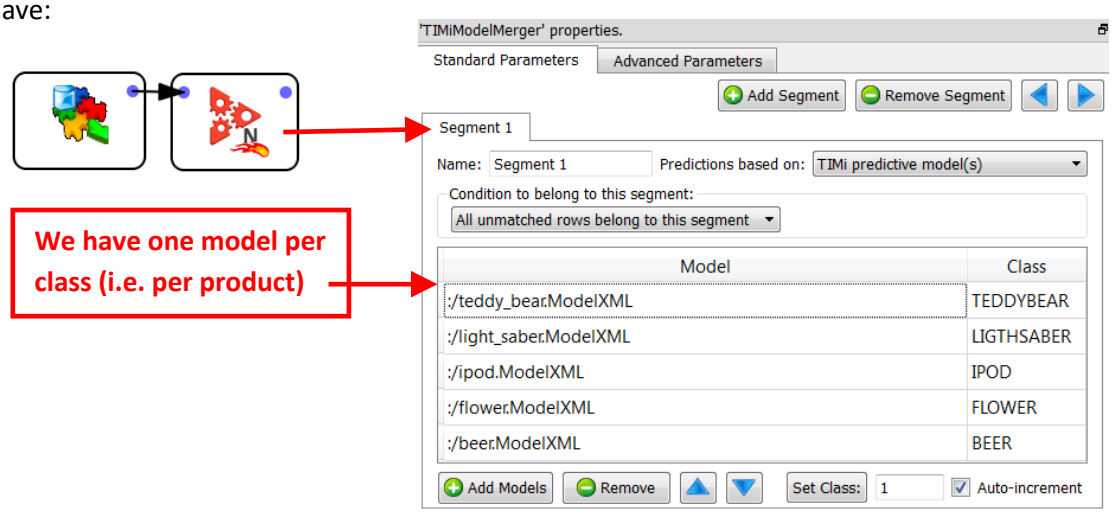
Let's first investigate the simplest case: "1 vs all others". In such a case, to make the prediction you need only 5 different binary predictive models (one for each product). You will apply these 5 models on each of the customers to obtain 5 purchase-probabilities (these 5 probabilities are different for all customers: This is true "one-to-one" marketing). The product with the highest purchase-probability will be the one that you will recommend as the next-to-buy. If you need to give a suggestion about 2 different products, you'll take the 2 products with the 2 highest probabilities.

It can happen that the quantity of each different product is limited. In such a case, you cannot simply suggest the products with the highest purchase-probabilities (and totally disregard the limitation on the quantity of each product). When such a constraint (on the quantity of each product) exists, the assignment of a specific product to a specific customer must be computed using a more complex mechanism: see section 5.16.1 about the "assignmentSolver".

Let's return to a very simple case: More precisely:

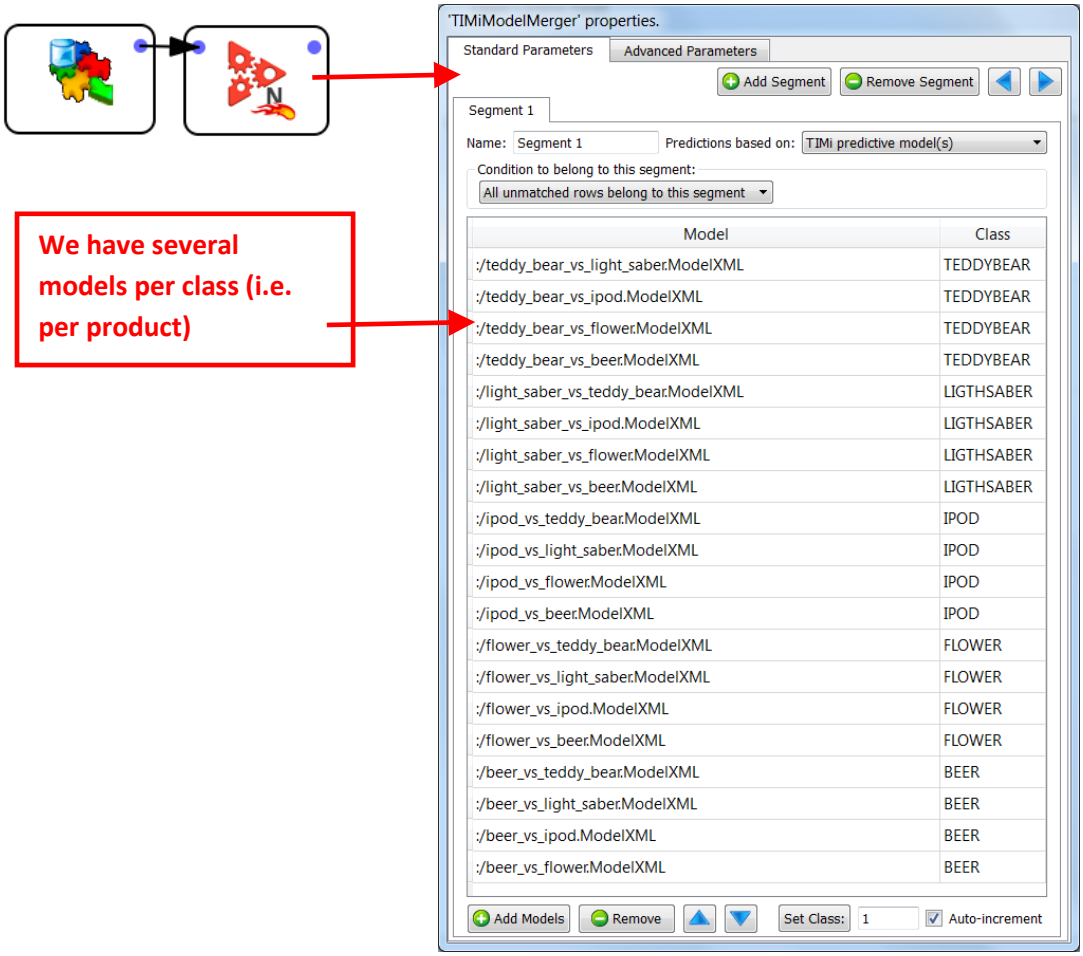
- There are no constraints on the quantity of each of the products.
- We are using the "*1 vs all others*" type of predictive models.

We'll have:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

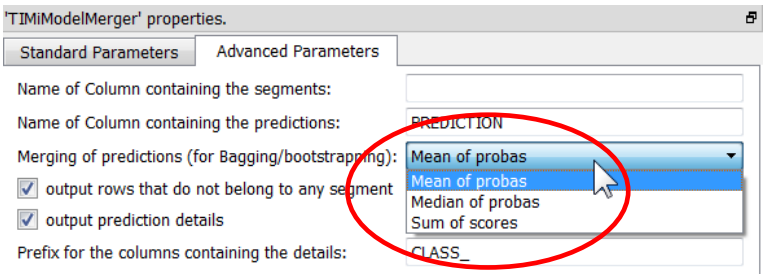
For the “1 vs 1” type of predictive models, we’ll have:



By default, when there are several models “voting” for the same class, the final (purchase) probability is the mean of the individual probabilities of each model. For example: The probability of buying a “teddy bear” is the mean of the 4 probabilities given by the 4 models:

- :/teddy_bear_vs_light_saber.ModelXML
- :/teddy_bear_vs_ipod.ModelXML
- :/teddy_bear_vs_flower.ModelXML
- :/teddy_bear_vs_beer.ModelXML

The “mean” operator is the default operator that is used to aggregate the individual probabilities of each predictive model to obtain the “final” probability for each different class (LIGHTSABER, IPOD, FLOWER, BEER). You can also use a different operator:



Even for simple Binary Classifier, it’s sometime interesting to use several predictive models (instead of just one). Depending on the way these models are created, this approach is either named “bagging”, “boosting” or “ensemble learning”. Why use several models instead of one? Because it usually delivers higher predictive accuracy.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

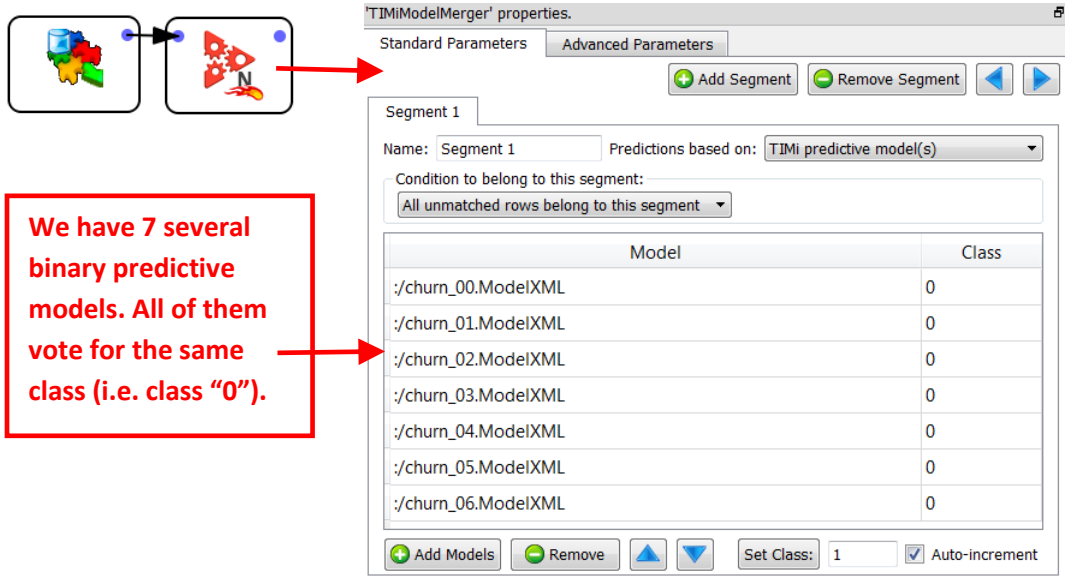
Why do we get higher accuracy when using many models?

For example: Let's assume that we have 10 predictive models. The first 4 predictive models P1, P2, P3 & P4 always recognize and classify correctly the class C1 (they are "specialized" for C1). The other predictive models are specialized in other classes. Let's now assume that you want to classify a new example (i.e. a new row) and that this example belongs to the class C1. The 4 predictive models P1, P2, P3 &P4 will correctly answer C1. The other predictive models will give random answers that will form some "uniformly distributed noise" during the vote. The result of the vote will thus be C1. We just realized an error de-correlation.

How to generate the 10 different predictive models (BAGFS) ?

We don't have to know what's the specialty of each of the predictive models. We only have to build predictive models that have various behaviors. We will build the 10 different predictive models using 10 different learning datasets. How do we generate these 10 learning datasets? We will use only a small part (called a bootstrap) of the "full" learning dataset (this technique is called BAGGING). Each bootstrap (there are 10 of them) is built using random examples (i.e. random rows) taken from the "full" learning dataset (random selection with duplication **allowed**). We will also typically use for each bootstrap a different subset of all the features/columns available (this technique is called FEATURE SELECTION). The selection of the features/columns included inside each bootstrap is random (random selection with duplication **forbidden**). The combination of BAGGING and FEATURE SELECTION is named BAGFS.

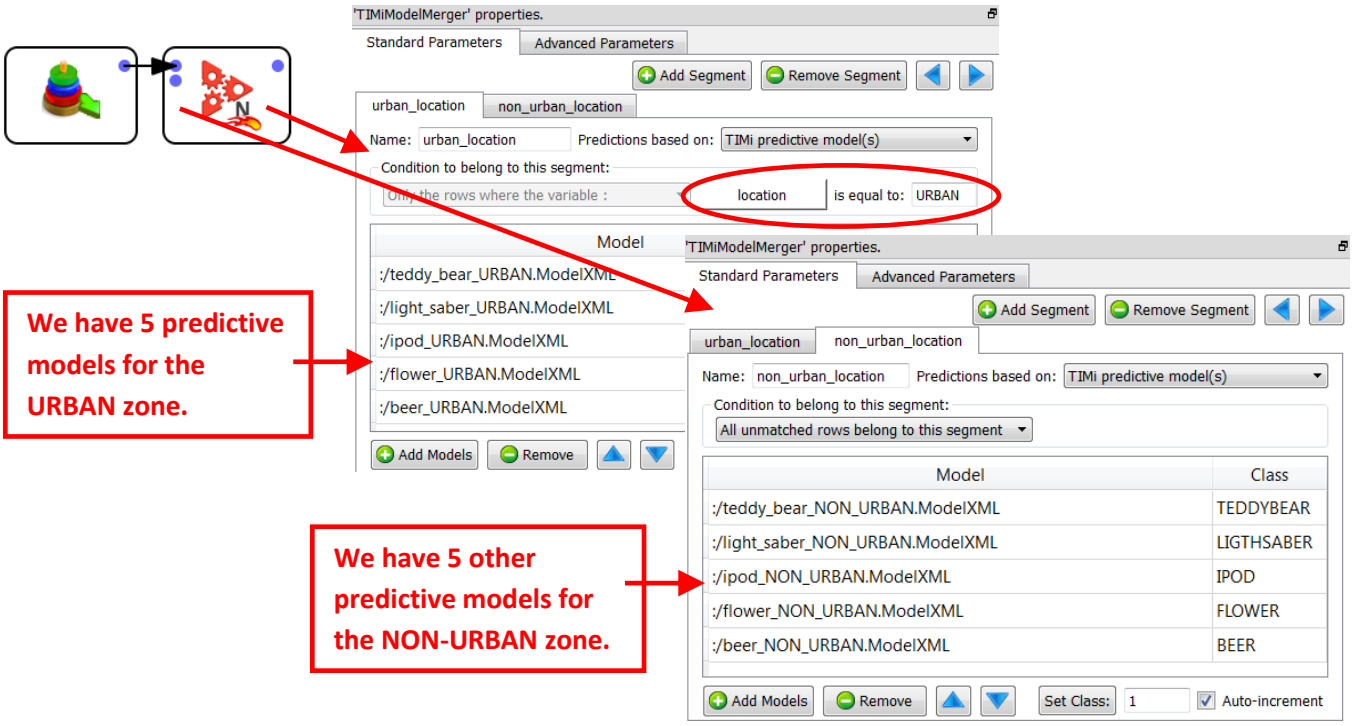
To do "bagging", "boosting" or "ensemble learning" with Anatella, use the following settings:



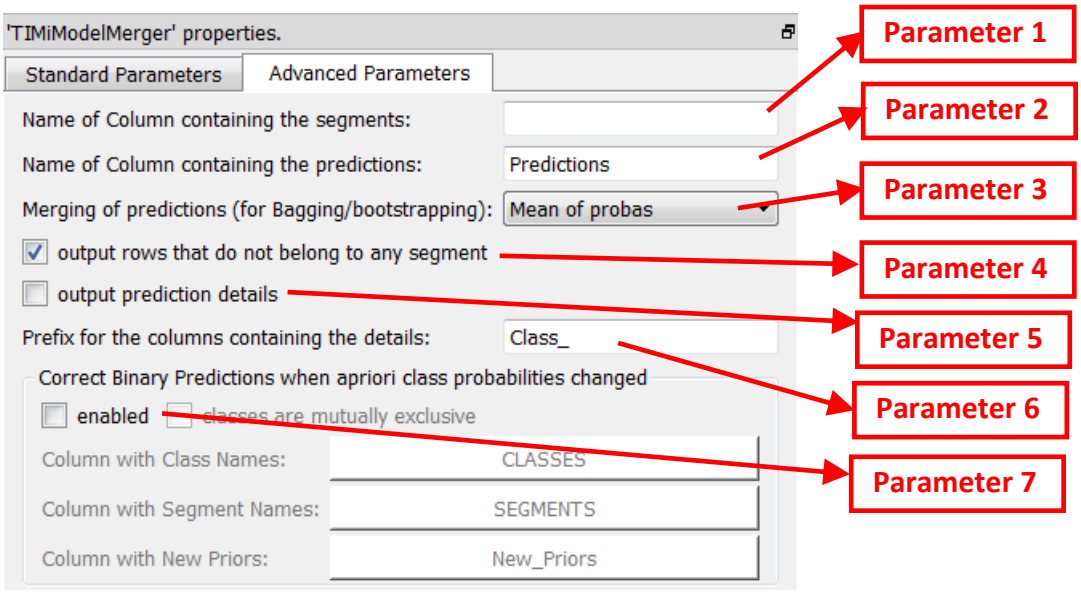
Once again, the final churn probability is the mean (or the median, depending on the settings) of the individual probabilities of each of the 7 models.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Finally, it's quite common to create different predictive models for the different segments of your population. For example: it can happen that the purchasing behavior of your customers is totally different depending if the customer resides in an urban zone or not. In such a case, we'll have 2 sets of models ("urban" and "non-urban"). For example:



The advanced parameter tab contains some more parameters:



Parameter 1: If this parameter is non-empty, a column containing the segment name (e.g. "urban_location" or "non_urban_location") is added to the result table.

Parameter 2: Name of the column containing the final prediction:

- For a binary classification problem, this column contains the 'final" probability.
- For a continuous prediction problem, this column contains the 'final" prediction.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- For a Multi-Class classification problem, this column contains the most-likely class (e.g. “TEDDYBEAR”, “LIGTHSABER”, “IPOD”, “FLOWER”, “BEER”).

Parameter 3: The operator user to aggregate into one “final” number the individual probabilities given by each predictive model (to obtain the final probability to belong to a specific class).

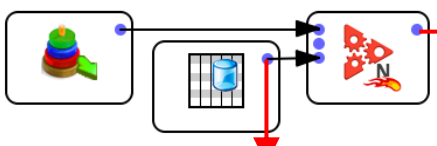
Parameter 4: Self-explanatory

Parameter 5: When the parameter “output prediction details” is checked, we’ll obtain:

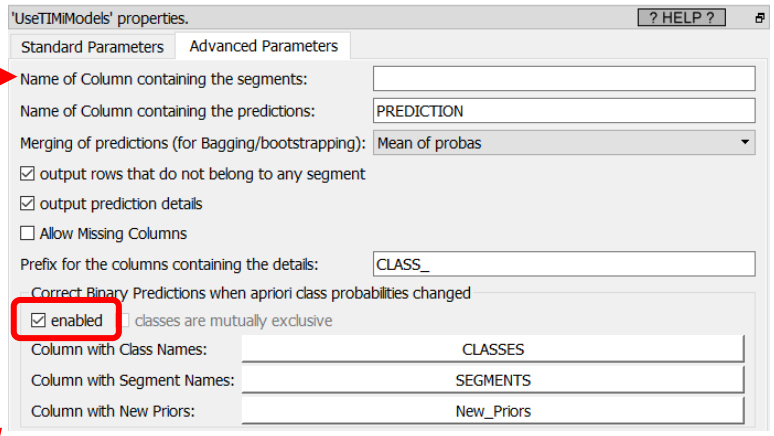
- For a binary or continuous predictive model: Each of the individual probabilities given by each of the predictive models.
- For a multi-class predictive model: The different probabilities to belong to each of the different classes. When using the “1 vs 1” type of predictive models or when using “bagging”, “boosting” or “ensemble learning”, these probabilities are already “aggregated” values of different individual probabilities.

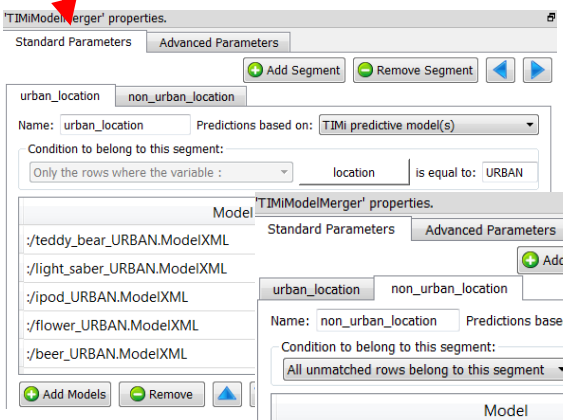
Parameter 6: Prefix of the name of the columns containing the “details”. You can change the prefix to avoid any column name collision.

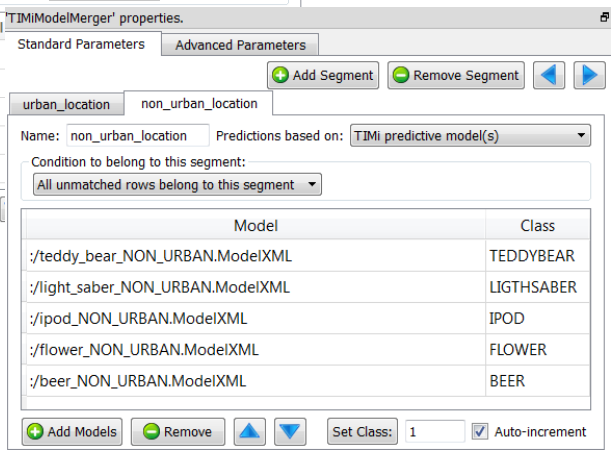
Parameter 7: When this option is activated, Anatella computes some corrections on the probabilities computed by the different predictive models to account for the fact that the density of (binary) targets inside the training set(s) is different from the density of targets inside the apply/scoring set. This correction is named, in technical terms, “apriori correction”. To use this option, you must give on the second input pin of this Action, a table that contains the expected density of targets inside the apply/scoring dataset (i.e. the new apriori probabilities) for each different classes&segments. Here is an example:



SEGMENTS	CLASSES	New_Priors
urban_location	TEDDYBEAR	0.2
urban_location	LIGTHSABER	0.2
urban_location	IPOD	0.2
urban_location	FLOWER	0.2
urban_location	BEER	0.2
non_urban_location	TEDDYBEAR	0.2
non_urban_location	LIGTHSABER	0.2
non_urban_location	IPOD	0.2
non_urban_location	FLOWER	0.2
non_urban_location	BEER	0.2







Same parameters as for the previous example:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

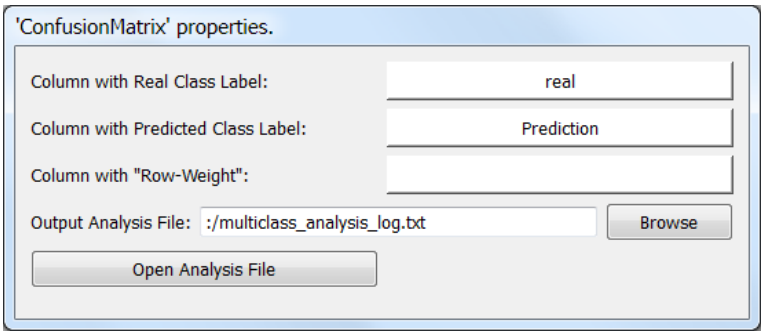
In the above example we corrected the apriori probabilities so that all items (teddy bear, light saber, ipod, flower, beer) have equal apriori's (i.e. 20%) whatever the percentage of targets that was actually observed in the learning dataset(s) used to create the various predictive models.

For even faster deployment of your scoring/models, the TIMiUseModels Action can run inside a N-Way multithreaded section (see section 5.3.2. about multithreading).

5.7.2. Confusion Matrix (High-Speed action)

Icon:

Property window:



Short description:
Analyze the performances of multi-class predictive models.

Long Description:
Binary predictive models or continous predictive models are typically built with TIMi and TIMi automatically delivers a set of graphical report that allows you to easily estimate the quality/accuracy of your models. These reports contain AUC (for binary classification) or R^2 (for continuous predicition) that allows you to easily select the best models.

To estimate the quality of your Multi-class predictive models (which are typically built using the TIMiModelMerger Action), you need to "manually" use the ConfusionMatrix Action.

The ConfusionMatrix Action produces an analysis report containing:

- The accuracy of your Multi-class predictive model.
- The accuracy of the "constant" Multi-class predictive model (i.e. the accuracy of the predictive model that always gives as answer the majority class). This is useful because it gives you "baseline" on which to compare your predictive model. The question is: "*Is your predictive model really better than the simple "constant" predictive model?*".
- The confusion matrix of your Multi-class predictive model.
  Inside the report, this matrix has 2 representations:
  - Each cell contains an absolute value of the number of examples inside your scoring dataset.
  - Each cell contains a percentage compared to the total number of examples inside your scoring dataset.
- The kappa statistic of your Multi-class predictive model. The Kappa statistics is the de-facto standard when estimating the accuracy of a multi-class classifier. For more information about the Kappa statistics: http://en.wikipedia.org/wiki/Cohen%27s_kappa

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The “percentage of correct classification” and the “correlation coefficient” are inappropriate statistics when measuring the agreement between the predicted class and the real observed class. The correct statistic is Kappa because it corrects the proportion of agreement due to chance. Here is a table that helps you interpret the strength of the agreement:

Value of κ (kappa)	Strength of agreement
≤0.20	Poor
0.21 - 0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Good
0.81-1.00	Very good

Here is an example of report:

```
Confusion Matrix Report v1.01
=============================

Number of rows =3204
Number of rows with invalid weight (ignored) =0

Predicted Class Labels =
   Class 1='METRO'
   Class 2='SPORT'
   Class 3='FIN'
   Class 4='ENTERTAINMENT'
   Class 5='FOREIGN'
   Class 6='NATIONAL'

Real Class Labels=
   Class A='Metro'
   Class B='Sports'
   Class C='Financial'
   Class D='Entertainment'
   Class E='Foreign'
   Class F='National'
```

```
TIMi Classifier
---------------
  Kappa=77.4762 %
  Percentage Of Correct classification=82.3346 %
  Confusion Matrix:
             1       2       3       4       5       6
      A    877      17      26      10      13       0
      B     56     671       2       3       6       0
      C    104       2     431       1      15       2
      D     55       8       4     260       0      27
      E     75       1       2       1     232      30
      F     81       7      11       6       1     167

  Confusion Matrix (percentage):
             1       2       3       4       5       6
      A  27.37    0.53    0.81    0.31    0.41    0.00
      B   1.75   20.94    0.06    0.09    0.19    0.00
      C   3.25    0.06   13.45    0.03    0.47    0.06
      D   1.72    0.25    0.12    8.11    0.00    0.84
      E   2.34    0.03    0.06    0.03    7.24    0.94
      F   2.53    0.22    0.34    0.19    0.03    5.21

"Always vote for Majority Class" Classifier
-------------------------------------------
  Kappa=0 %
  Percentage Of Correct classification=29.432 %
  Confusion Matrix:
             1       2       3       4       5       6
      A    943       0       0       0       0       0
      B    738       0       0       0       0       0
      C    555       0       0       0       0       0
      D    354       0       0       0       0       0
      E    341       0       0       0       0       0
      F    273       0       0       0       0       0

  Confusion Matrix (percentage):
             1       2       3       4       5       6
      A  29.43    0.00    0.00    0.00    0.00    0.00
      B  23.03    0.00    0.00    0.00    0.00    0.00
```

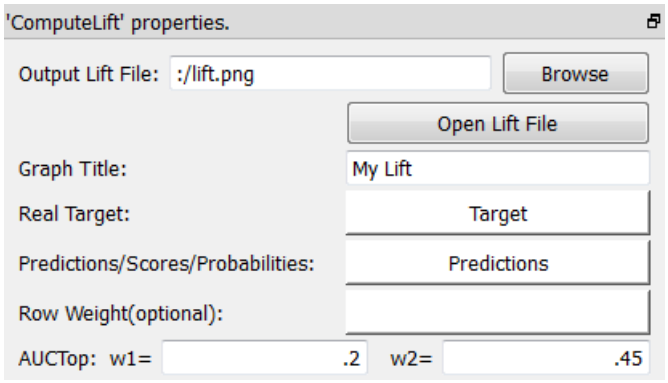

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Using the ConfusionMatrix Action, you can quickly find which one of your many different Multi-class predictive models is the best (i.e. which one has the highest accuracy).

5.7.3. Lift Curves (High-Speed action)

Icon:

Property window:



Short description:
Compute a Lift Curve

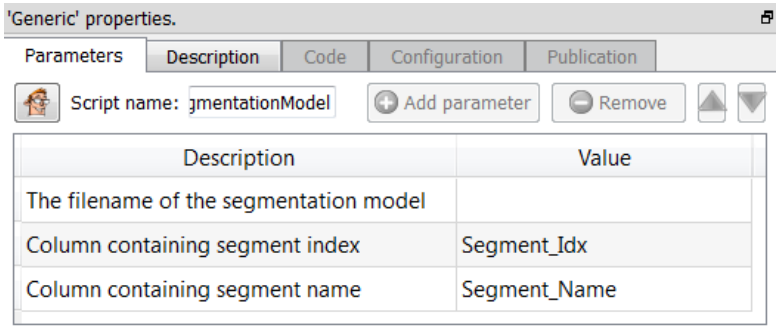
Long Description:
Create a PNG file that contains:

- The lift curve.
- The AUC (Area Under the Lift Curve)
- The AUCtop (Area Under the top of the Lift Curve) (the top is defined with the prarameters w1 and w2)
- The total number of rows & the number of targets. These can be non-integer numbers if the "Row Weight" option is used)

5.7.4. Segmentation Model

Icon:

Property window:



Short description:
Apply one Stardust Segmentation model.

Long Description:
Self-Explanatory.

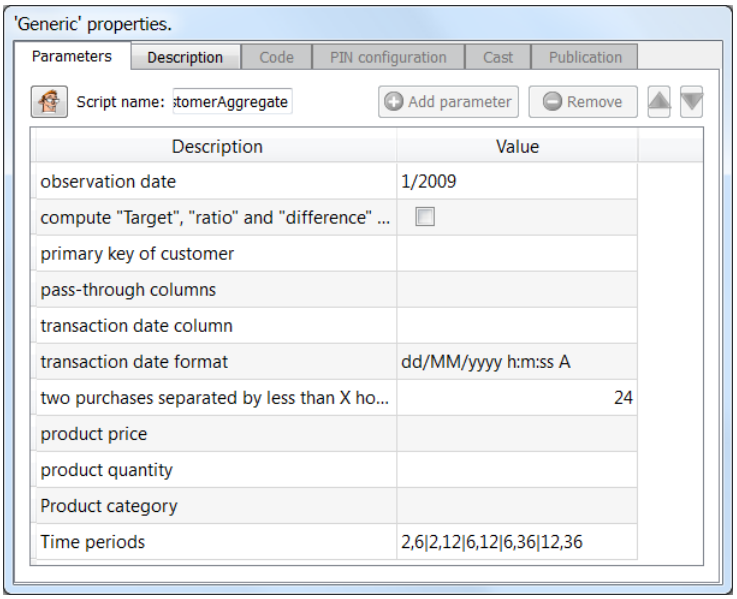
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For even faster deployment of your multivariate segmentation, the segmentationModel Action can run inside a N-Way multithreaded section (see section 5.3.2. about multithreading).

5.7.5. Transaction-level to Customer-level Aggregation

Icon:

Property window:



Short description:
Transaction-level to customer-level aggregation. This Action transforms the Transaction-Table (containing all the transactions made by the customers - one transaction per row) into a Customer-Table (each row of the Customer-Table consolidates all the information pertaining to one customer).

Long Description:

Pre-requisite
The input table of this action must contain all the transactions sorted first by customers, and secondly by time.

In most commercial systems, you have inside the operational system the following 3 tables:

1. The Transaction or Event-Table
   The Transaction-Table contains all the transactions that have occured. Each row of the "Transaction-Table" represents a transaction. For each transaction, you usually know, at least, the following (these are the columns of the transaction-table):
   - ID_Customer (the primary key of the customer)
   - ID_Product (the primary key of the product that has been purchased)
   - Date_of_transaction

   See the next pages for an example of Transaction Table.

2. The Product-Table

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Usually, all the available products are placed in different categories. These categories are given inside the "Product-table". Each row of the "Product-table" contains many informations on a different products and, in particular, their category.

3. The Customer-Table
   The "Customer-Table" contains many information about a customer: its age, postal address (this is required for the delivery of the purchased items), sex,etc. Usually, the "Customer-table" does not contain any information about the products that a specific customer has bought.

The objective of this script is to aggregate all the information contained into these 3 tables into one (possibly very large) Global-Customer-Level table.

This Global-table can thereafter be used to:

1. to create a good segmentation of your customer base
2. to create (with TIM!) powerful predictive models for cross-selling purposes.

Each row of the final Global-Table represents one customer. The columns of the Global-Table contains all the informations that we know about the customers. In particular, there are some columns very important columns that describe the "purchase habit" of the customers. For example: "*How many purchases are there in a specific category over the last x months?*"

To summarize: This script allows:

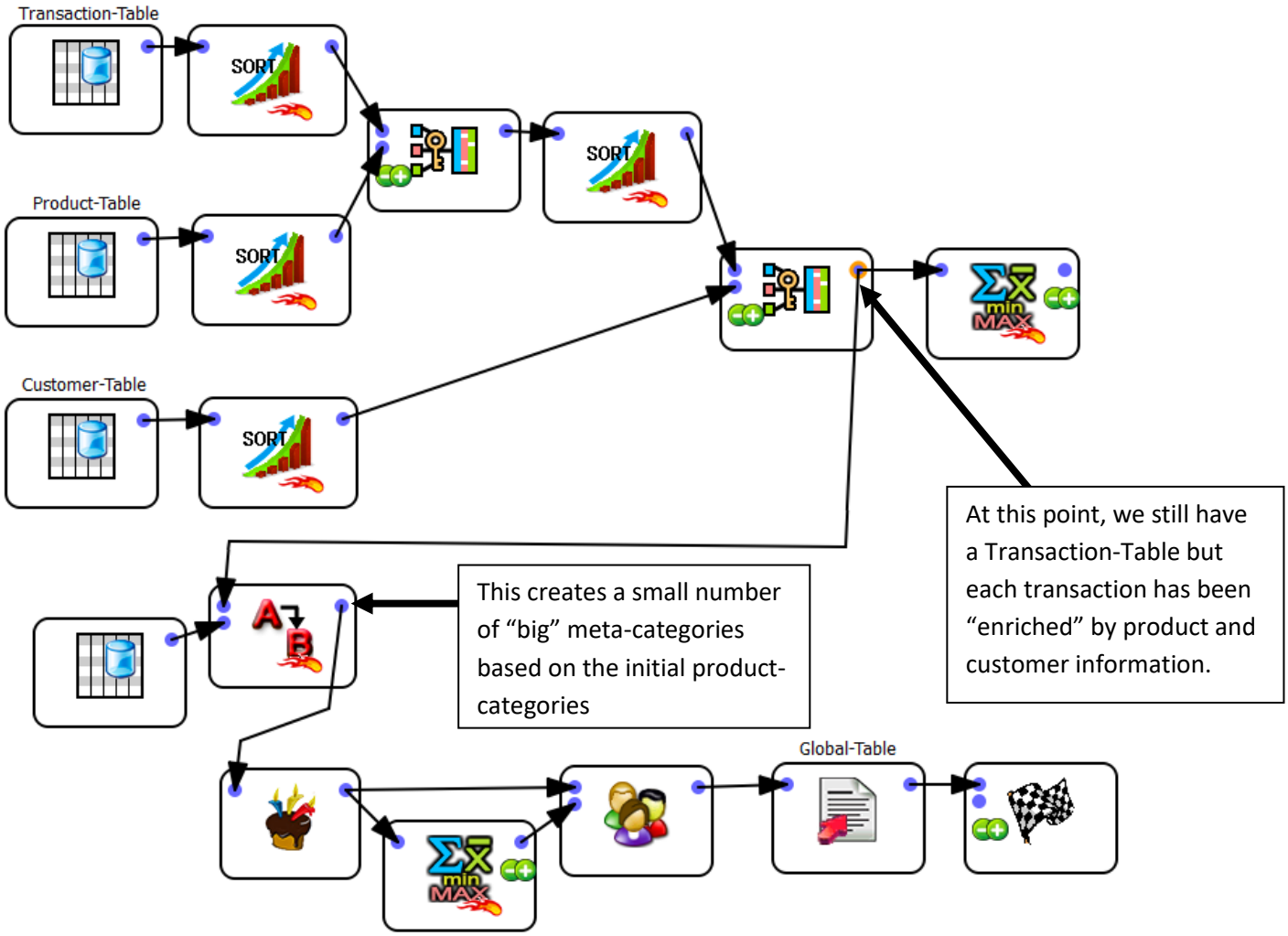
- to extract the "purchase habit" of each customer out of the Transaction-Table
- to put the result into a "Customer-Level" Table.

Basically, it transforms a "*Transaction-level*" table into a "*Customer-level*" table.
As you can see, Anatella allows you to easily program these types of transformation. Other ETL tools are not as flexible since they do not offer any transformation-operators that allow you to perform easily such (very common) data-transformation. It's usually always an incredibly painful nightmare to try to do the same job into any other ETL tool. But fortunately, now, Anatella is there to save the day!

Let's take a first explanatory example.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is a typical Anatella-Graph using the “ Action”:



An example Customer-Table looks like this:

DataTable

	CustomerID	Title	Date_of_Birth	Marital_Status	City	Zip Code
1	2	Mr	05-01-1984	SINGLE	Brussel	1000
2	1	Ms	04-11-1990	SINGLE	Brussels	1050
3	5	Ms	22-11-1992	MARRIED	Bruxell	1000
4	3	Ms	31-12-1974	SINGLE	Brussels	1050

Copy Filter Rows Filter Columns Define Filters

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

An example Transaction-Table looks like this:

DataTable

	CustomerID	ProductID	Price	Transaction_Date	Product_Quantity
1	5	12	144	1/02/2006 18:13	1
2	5	11	-144	1/02/2006 18:13	1
3	5	11	144	1/02/2006 18:13	1
4	2	100	175	1/02/2006 18:21	5
5	5	19	298	1/02/2006 18:57	1
6	2	22	359	1/02/2006 19:58	1
7	5	23	593	1/03/2006 21:00	1
8	5	24	316	1/03/2006 21:00	1
9	5	45	149	1/05/2006 16:32	1
10	2	101	369	1/05/2006 16:33	1
11	5	101	69	1/05/2006 17:52	1
12	5	12	120	1/06/2006 18:55	1
13	5	102	79	1/06/2006 19:22	1
14	2	23	776	1/07/2006 20:15	1
15	5	108	426	1/07/2006 20:16	1
16	1	200	154	1/03/2006 20:17	8
17	5	104	522	1/07/2006 20:18	1
18	5	40	79	1/07/2006 20:18	1
19	3	11	149	1/07/2006 20:18	1
20	3	12	379	1/07/2006 20:21	1

Copy | Filter Rows | Filter Columns | Define Filters

An example Product-Table looks like this:

DataTable

	ProductID	Title	Category
1	11	Elvis Presley: Jailhouse Rock	Rock'N'Roll
2	12	Bill Haley: Rock Around the clock	Rock'N'Roll
3	19	Jerry Lee Lewis: Great ball of Fire	Rock'N'Roll
4	22	Little Richard: Tutti Frutti	Rock'N'Roll
5	23	Chuck Berry: Roll over Beethoven	Rock'N'Roll
6	24	Carl Perkins: Blue Suede Shoes	Rock'N'Roll
7	40	Chubby Checker: The twist	Rock'N'Roll
8	45	Big Joe Turner: Shake, Rattle & Roll	Rock'N'Roll
9	100	Aqua: Barbie Girl	Pop -Dance
10	101	Lau Bega: Mambo N°5	Pop -Mambo
11	102	Wynonna: Burning love	Pop - Movie
12	104	Vaya Con Dios: Nah Neh Nah	Pop - Gipsy
13	108	James Brown: I feel good	Pop - Blues
14	200	Patrick Suskin: Parfume	Audio Book

Copy | Filter Rows | Filter Columns | Define Filters

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is a view of the "Enriched" Transaction-Table:

					Product information.		Customer information.				
CustomerID	ProductID	Price	Transaction Date	Product Quantity	Product.Title	Product Category	Customer Title	Customer: Date_of_Birth	Customer Marital_Status	Customer City	Customer Zip Code
1	200	154	1/03/2006	8	Patrick Suskin: Parfume	Audio Book	Ms	4/11/1990	SINGLE	Brussels	1050
2	22	359	1/02/2006	1	Little Richard: Tutti Frutti	Rock'N'Roll	Mr	5/01/1984	SINGLE	Brussel	1000
2	101	369	1/05/2006	1	Lau Bega: Mambo N°5	Pop -Mambo	Mr	5/01/1984	SINGLE	Brussel	1000
2	100	175	1/02/2006	5	Aqua: Barbie Girl	Pop -Dance	Mr	5/01/1984	SINGLE	Brussel	1000
2	23	776	1/07/2006	1	Chuck Berry: Roll over Beethoven	Rock'N'Roll	Mr	5/01/1984	SINGLE	Brussel	1000
3	12	379	1/07/2006	1	Bill Haley: Rock Around the clock	Rock'N'Roll	Ms	31/12/1974	SINGLE	Brussels	1050
3	11	149	1/07/2006	1	Elvis Presley: Jailhouse Rock	Rock'N'Roll	Ms	31/12/1974	SINGLE	Brussels	1050
5	12	144	1/02/2006	1	Bill Haley: Rock Around the clock	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	108	426	1/07/2006	1	James Brown: I feel good	Pop - Blues	Ms	22/11/1992	MARRIED	Bruxell	1000
5	104	522	1/07/2006	1	Vaya Con Dios: Nah Neh Nah	Pop - Gipsy	Ms	22/11/1992	MARRIED	Bruxell	1000
5	102	79	1/06/2006	1	Wynonna: Burning love	Pop - Movie	Ms	22/11/1992	MARRIED	Bruxell	1000
5	101	69	1/05/2006	1	Lau Bega: Mambo N°5	Pop -Mambo	Ms	22/11/1992	MARRIED	Bruxell	1000
5	45	149	1/05/2006	1	Big Joe Turner: Shake, Rattle & Roll	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	19	298	1/02/2006	1	Jerry Lee Lewis: Great ball of Fire	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	40	79	1/07/2006	1	Chubby Checker: The twist	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	24	316	1/03/2006	1	Carl Perkins: Blue Suede Shoes	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	11	144	1/02/2006	1	Elvis Presley: Jailhouse Rock	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	23	593	1/03/2006	1	Chuck Berry: Roll over Beethoven	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	12	120	1/06/2006	1	Bill Haley: Rock Around the clock	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000
5	11	-144	1/02/2006	1	Elvis Presley: Jailhouse Rock	Rock'N'Roll	Ms	22/11/1992	MARRIED	Bruxell	1000

The next "Replace String" Action creates a small number of "big" meta-categories based on the initial product-categories. The category-regroupments are defined inside the "Inline-Table" connect on pin 2 of the "Replace String" Action. This "Inline-Table" is:

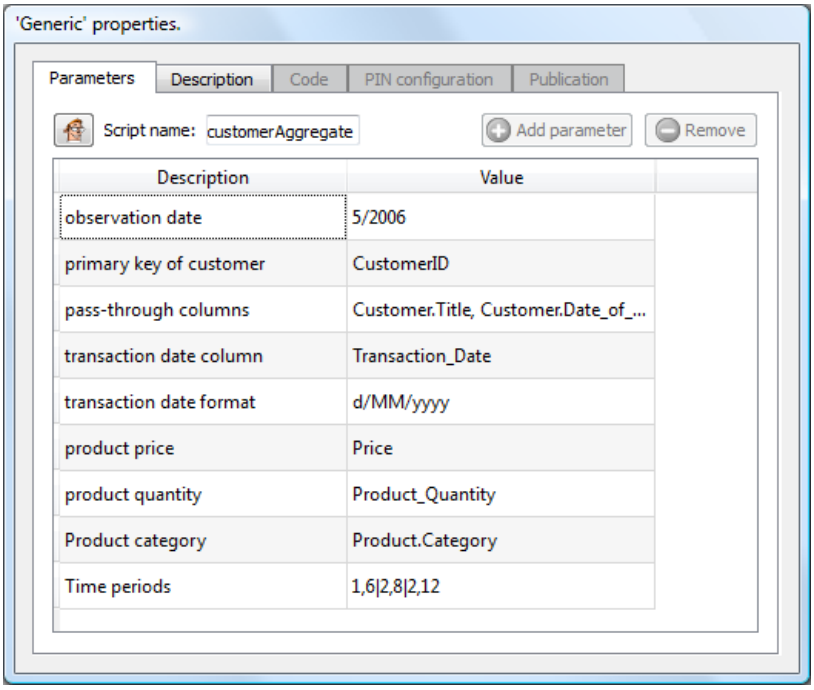
DataTable

	Column Name	before	after	count
1	Product.Category	Audio Book	Audio Book	1
2	Product.Category	Pop - Blues	Pop	1
3	Product.Category	Pop - Gipsy	Pop	1
4	Product.Category	Pop - Movie	Pop	1
5	Product.Category	Pop -Dance	Pop	1
6	Product.Category	Pop -Mambo	Pop	2
7	Product.Category	Rock'N'Roll	Rock'N'Roll	13

Copy | Filter Rows | Filter Columns | Define Filters

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Thereafter the " Action" transforms the "Enriched Transaction-table" into a "Customer-Level" table directly usable for predictive & segmentation analytics. The parameters of the " Action" are:



Anatella will generate as output a "Customer-Level" table containing a set of new columns. These columns are:

A. <u>RFM columns:</u>

Let's set the observation date as "o".

For each "product_category", Anatella will create these variables:

- Recency:
  - number of months since last purchase
- Frequency: For each time period (x,y):
  - number of transactions in [x o]
  - (average number of transactions in [x o]) - (average number of transactions in [y x])
  - (average number of transactions in [x o]) / (1+(average number of transactions in [y x]))
  - number of purchased items in [x o]
  - (average number of purchased items in [x o]) - (average number of purchased items in [y x])
  - (average number of purchased items in [x o]) / (1+(average number of purchased items in [y x]))
- Monetary: For each time period (x,y):
  - sum of purchase values in [x o]
  - (average purchase values for transactions in [x o]) - (average purchase values for transactions in [y x])
  - (average purchase values for transactions in [x o]) / (1+(average purchase values for transactions in [y x]))

The same columns are also computed ignoring the "product_category".

B. <u>Target Columns:</u>

The "Target Columns" that are generated by Anatella are:

- For each "product_category":
  - Is there a purchase after the observation date? yes/no
  - Number of transactions after the observation date?
  - Sum of purchases values after the observation date?

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The same targets are also computed ignoring the “product_category”.

Inside our example, the observation date "o" is “5/2006” and the Time-Periods (x,y) are:

{ (-1,-6) ; (-2,-8) ; (-2,-12) }

This means that the following columns are created as output of the “ Action”:

CustomerID
Customer.Title
Customer.Marital_Status
Customer.City
Customer.Zip Code
Customer.Age
Recency last purchase
nItem in [-1 observ.]
ntransactions in [-1 observ.]
sumTransactionsValues in [-1 observ.]
(Avg nItems in [-1 observ.]) - (Avg nItems in [-6 -1])
(Avg nItems in [-1 observ.]) / (1+Avg nItems in [-6 -1])
(Avg nTransactions in [-1 observ.]) - (Avg nTransactions in [-6 -1])
(Avg nTransactions in [-1 observ.]) / (1+Avg nTransactions in [-6 -1])
(Avg transactionsValue in [-1 observ.]) - (Avg transactionsValue in [-6 -1])
(Avg transactionsValue in [-1 observ.]) / (1+Avg transactionsValue in [-6 -1])
nItem in [-2 observ.]
ntransactions in [-2 observ.]
sumTransactionsValues in [-2 observ.]
(Avg nItems in [-2 observ.]) - (Avg nItems in [-8 -2])
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-8 -2])
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-8 -2])
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-8 -2])
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-8 -2])
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-8 -2])
(Avg nItems in [-2 observ.]) - (Avg nItems in [-12 -2])
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-12 -2])
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-12 -2])
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-12 -2])
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-12 -2])
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-12 -2])
BTarget: Is Lapsed customer(churner)
BTarget: Is Recurrent customer
CTarget: number of purchased items after obs.date
CTarget: sum of transactions values
Recency last purchase in category Audio Book
nItem in [-1 observ.] in category Audio Book
ntransactions in [-1 observ.] in category Audio Book

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

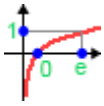
sumTransactionsValues in [-1 observ.] in category Audio Book
(Avg nItems in [-1 observ.]) - (Avg nItems in [-6 -1]) in category Audio Book
(Avg nItems in [-1 observ.]) / (1+Avg nItems in [-6 -1]) in category Audio Book
(Avg nTransactions in [-1 observ.]) - (Avg nTransactions in [-6 -1]) in category Audio Book
(Avg nTransactions in [-1 observ.]) / (1+Avg nTransactions in [-6 -1]) in category Audio Book
(Avg transactionsValue in [-1 observ.]) - (Avg transactionsValue in [-6 -1]) in category Audio Book
(Avg transactionsValue in [-1 observ.]) / (1+Avg transactionsValue in [-6 -1]) in category Audio Book
nItem in [-2 observ.] in category Audio Book
ntransactions in [-2 observ.] in category Audio Book
sumTransactionsValues in [-2 observ.] in category Audio Book
(Avg nItems in [-2 observ.]) - (Avg nItems in [-8 -2]) in category Audio Book
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-8 -2]) in category Audio Book
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-8 -2]) in category Audio Book
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-8 -2]) in category Audio Book
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-8 -2]) in category Audio Book
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-8 -2]) in category Audio Book
(Avg nItems in [-2 observ.]) - (Avg nItems in [-12 -2]) in category Audio Book
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-12 -2]) in category Audio Book
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-12 -2]) in category Audio Book
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-12 -2]) in category Audio Book
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-12 -2]) in category Audio Book
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-12 -2]) in category Audio Book
BTarget: Is Lapsed customer(churner) in category Audio Book
BTarget: Is Recurrent customer in category Audio Book
CTarget: number of purchased items after obs.date in category Audio Book
CTarget: sum of transactions values in category Audio Book
Recency last purchase in category Pop
nItem in [-1 observ.] in category Pop
ntransactions in [-1 observ.] in category Pop
sumTransactionsValues in [-1 observ.] in category Pop
(Avg nItems in [-1 observ.]) - (Avg nItems in [-6 -1]) in category Pop
(Avg nItems in [-1 observ.]) / (1+Avg nItems in [-6 -1]) in category Pop
(Avg nTransactions in [-1 observ.]) - (Avg nTransactions in [-6 -1]) in category Pop
(Avg nTransactions in [-1 observ.]) / (1+Avg nTransactions in [-6 -1]) in category Pop
(Avg transactionsValue in [-1 observ.]) - (Avg transactionsValue in [-6 -1]) in category Pop
(Avg transactionsValue in [-1 observ.]) / (1+Avg transactionsValue in [-6 -1]) in category Pop
nItem in [-2 observ.] in category Pop
ntransactions in [-2 observ.] in category Pop
sumTransactionsValues in [-2 observ.] in category Pop
(Avg nItems in [-2 observ.]) - (Avg nItems in [-8 -2]) in category Pop
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-8 -2]) in category Pop
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-8 -2]) in category Pop
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-8 -2]) in category Pop
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-8 -2]) in category Pop

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

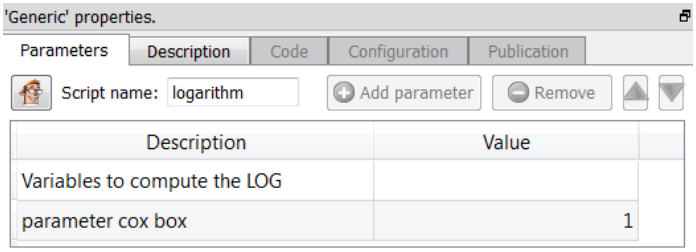
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-8 -2]) in category Pop
(Avg nItems in [-2 observ.]) - (Avg nItems in [-12 -2]) in category Pop
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-12 -2]) in category Pop
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-12 -2]) in category Pop
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-12 -2]) in category Pop
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-12 -2]) in category Pop
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-12 -2]) in category Pop
BTarget: Is Lapsed customer(churner) in category Pop
BTarget: Is Recurrent customer in category Pop
CTarget: number of purchased items after obs.date in category Pop
CTarget: sum of transactions values in category Pop
Recency last purchase in category Rock'N'Roll
nItem in [-1 observ.] in category Rock'N'Roll
ntransactions in [-1 observ.] in category Rock'N'Roll
sumTransactionsValues in [-1 observ.] in category Rock'N'Roll
(Avg nItems in [-1 observ.]) - (Avg nItems in [-6 -1]) in category Rock'N'Roll
(Avg nItems in [-1 observ.]) / (1+Avg nItems in [-6 -1]) in category Rock'N'Roll
(Avg nTransactions in [-1 observ.]) - (Avg nTransactions in [-6 -1]) in category Rock'N'Roll
(Avg nTransactions in [-1 observ.]) / (1+Avg nTransactions in [-6 -1]) in category Rock'N'Roll
(Avg transactionsValue in [-1 observ.]) - (Avg transactionsValue in [-6 -1]) in category Rock'N'Roll
(Avg transactionsValue in [-1 observ.]) / (1+Avg transactionsValue in [-6 -1]) in category Rock'N'Roll
nItem in [-2 observ.] in category Rock'N'Roll
ntransactions in [-2 observ.] in category Rock'N'Roll
sumTransactionsValues in [-2 observ.] in category Rock'N'Roll
(Avg nItems in [-2 observ.]) - (Avg nItems in [-8 -2]) in category Rock'N'Roll
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-8 -2]) in category Rock'N'Roll
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-8 -2]) in category Rock'N'Roll
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-8 -2]) in category Rock'N'Roll
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-8 -2]) in category Rock'N'Roll
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-8 -2]) in category Rock'N'Roll
(Avg nItems in [-2 observ.]) - (Avg nItems in [-12 -2]) in category Rock'N'Roll
(Avg nItems in [-2 observ.]) / (1+Avg nItems in [-12 -2]) in category Rock'N'Roll
(Avg nTransactions in [-2 observ.]) - (Avg nTransactions in [-12 -2]) in category Rock'N'Roll
(Avg nTransactions in [-2 observ.]) / (1+Avg nTransactions in [-12 -2]) in category Rock'N'Roll
(Avg transactionsValue in [-2 observ.]) - (Avg transactionsValue in [-12 -2]) in category Rock'N'Roll
(Avg transactionsValue in [-2 observ.]) / (1+Avg transactionsValue in [-12 -2]) in category Rock'N'Roll
BTarget: Is Lapsed customer(churner) in category Rock'N'Roll
BTarget: Is Recurrent customer in category Rock'N'Roll
CTarget: number of purchased items after obs.date in category Rock'N'Roll
CTarget: sum of transactions values in category Rock'N'Roll

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.7.6. Logarithm

Icon: 

Property window:



Short description:
Compute the logarithm of many different Columns.

Long Description:
The logarithm of the selected columns is computed in the following way:

Out= ((C+In)>0 ? log(C+In) : -1)

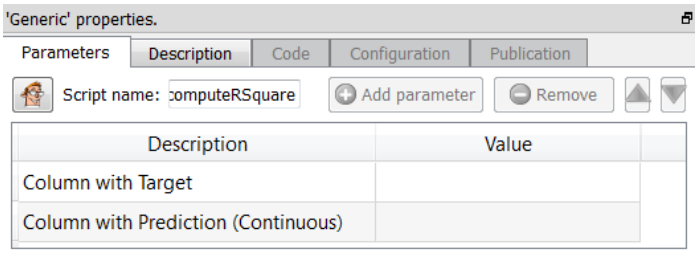
…where: “In” is the input number
“Out” is the result
“C” is the parameter cox box.
“x?y:z” is the classical ternary operator.

Transforming monetary values using a log function can sometime improve the accuracy of the continuous predictive models using these monetary values.

5.7.7. Compute RSquare

Icon:

Property window:



Short description:
Analyze the performances of a continuous predictive model.

Long Description:
This computes:

- the R-square "goodness of fit" statistics …
- the MAE (mean absolute error) …
- the RMSE (root mean squared error) …

…between 2 columns (the "real" target and the "predicted" target).

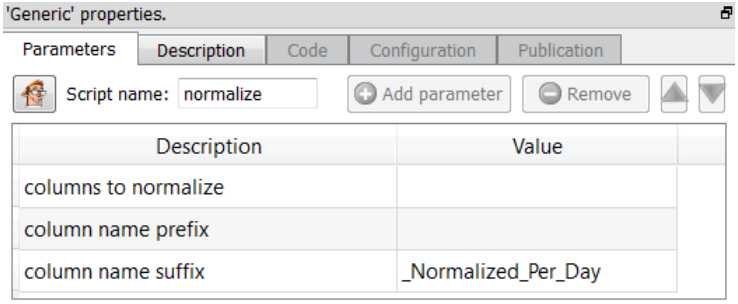
This also computes the mean and the standard deviation of the target column.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.7.8. Normalize

Icon:

Property window:

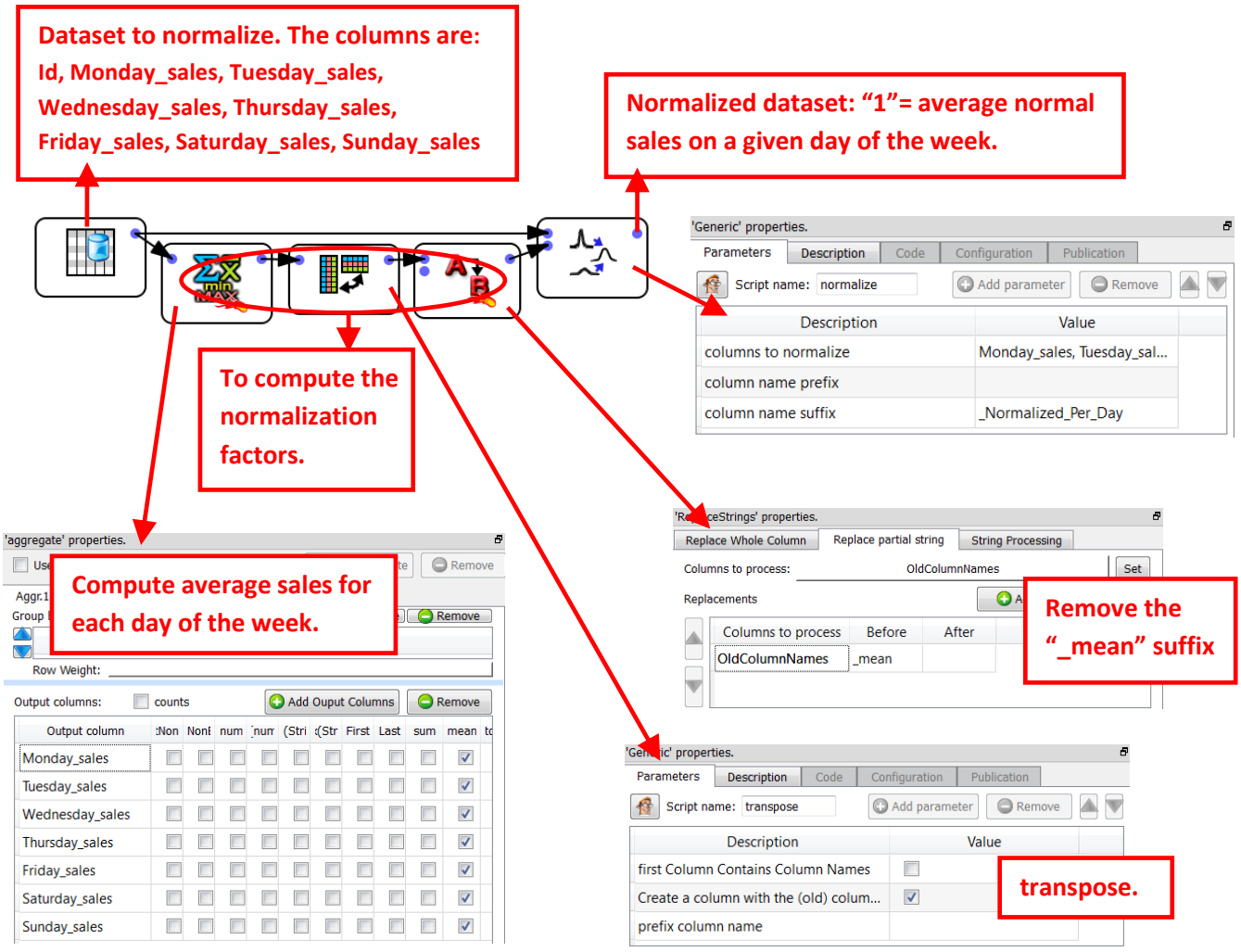


Short description:

Apply a normalization factor to some columns based on their name.

Long Description:

Let's assume that you have a table where each row represents a week and the columns contain the sales for each different day of the week. We want to normalize this dataset so that the value "1" represents the average normal sales on a given day of the week. Such kind of data transformation is useful when you want to create a predictive model that takes into account the "seasonality effects" (more precisely here: The seasonality effect of the day of the week). We'll have:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The pin 0 of the Normalize Action is connected to the dataset to normalize.

The pin 1 of the Normalize Action is connected to a table with two columns. The first column contains the name of the columns to normalize. The second column contains the normalization factors. For example, in the above Anatella graph, we have:

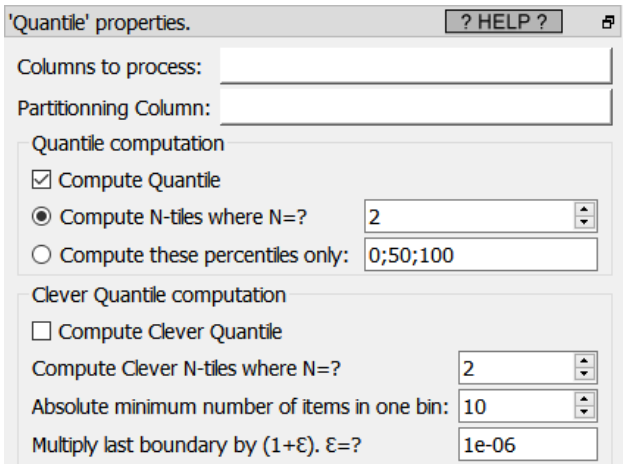
The set of columns given here must match the set of column given as the parameter named "columns to normalize" of the Normalize Action.

	OldColumnNames	C1
1	Monday_sales	39.75
2	Tuesday_sales	40
3	Wednesday_sales	39.75
4	Thursday_sales	39.25
5	Friday_sales	39.75
6	Saturday_sales	79.75
7	Sunday_sales	79.5

5.7.9. Quantile (High-Speed action)

Icon:

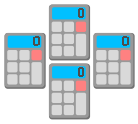
Property window:



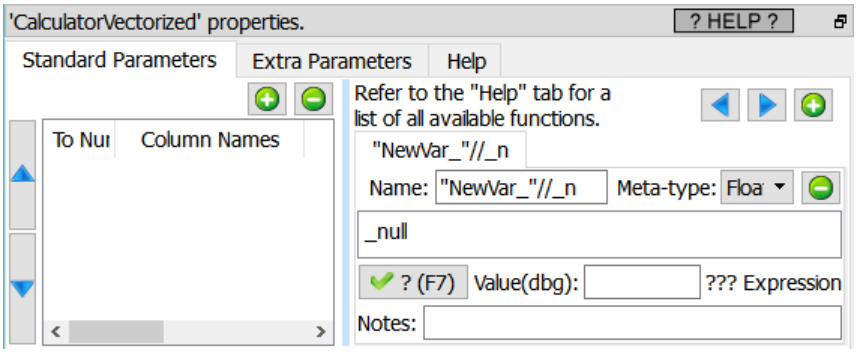
Short description:
Compute Quantiles and Clever Quantiles

Long Description:
Compute Quantiles and Clever Quantiles

5.7.10. Vectorized Calculator (High-Speed action)

Icon: 

Property window:



Short description:
Compute many new columns easily.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:

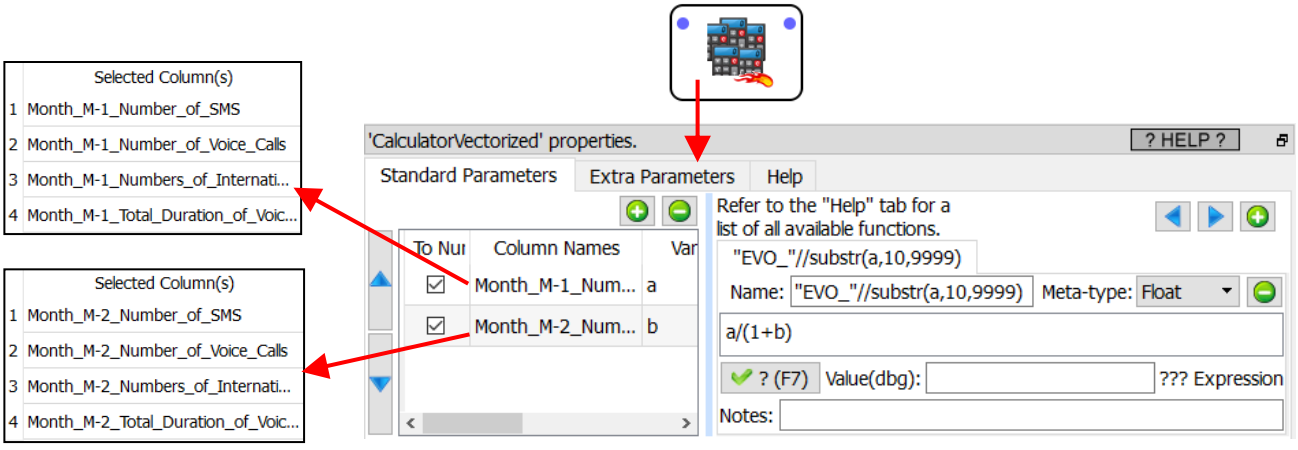
The first step before creating a predictive model is to build a learning dataset that contains as many "good" features as possible (this is named "feature engineering"). For example, in telecom, if you want to predict if a subscriber will renew its subscription the next month (i.e. if you want to do a "churn" model), it's interesting to look at the consumptions patterns of the previous months (i.e. to look at the Month "M-1", "M-2" and also maybe "M-3"). This means that you'll have a learning dataset that will include features such as:

- Month_M-1_Number_of_Voice_Calls
- Month_M-2_Number_of_Voice_Calls
- Month_M-1_Number_of_SMS
- Month_M-2_Number_of_SMS
- Month_M-1_Total_Duration_of_Voice_Calls
- Month_M-2_Total_Duration_of_Voice_Calls
- Month_M-1_Numbers_of_International_Calls
- Month_M-2_Numbers_of_International_Calls

The above 8 features are interesting in themselves but there exist some even better, more meaningful features: These features are named the "Evolution" features and represents the evolution over time of the "original, raw" features. Here are 4 "Evolution" features that are based on the list of 8 features shown here above:

- EVO_Number_of_Voice_Calls = $$\frac{\text{Month\_M-1\_Number\_of\_Voice\_Calls}}{1+\text{Month\_M-2\_Number\_of\_Voice\_Calls}}$$
- EVO_Number_of_SMS = $$\frac{\text{Month\_M-1\_Number\_of\_SMS}}{1+\text{Month\_M-2\_Number\_of\_SMS}}$$
- EVO_Total_Duration_of_Voice_Calls = $$\frac{\text{Month\_M-1\_Total\_Duration\_of\_Voice\_Calls}}{1+\text{Month\_M-2\_Total\_Duration\_of\_Voice\_Calls}}$$
- EVO_Numbers_of_International_Calls = $$\frac{\text{Month\_M-1\_Numbers\_of\_International\_Calls}}{1+\text{Month\_M-2\_Numbers\_of\_International\_Calls}}$$

One of the (many) objective of the Vectorized Calculator Action is to allow you to quickly create thousands of "Evolution" features. For example, to create the 4 "EVO_*" features shown here above, we'll have:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

About the above example:

- The equation that is used to compute the "Evolution" variables is "`a/(1+b)`" but many other variations exist: For example: "`a-b`", "`b?a/b:0`", "`2*a/(b+c)`", etc.
- We created 4 new "Evolution" variables/columns. In most "real-life" examples, the Vectorized Calculator Action is rather used to create between 100 and 1000 new variables/columns.

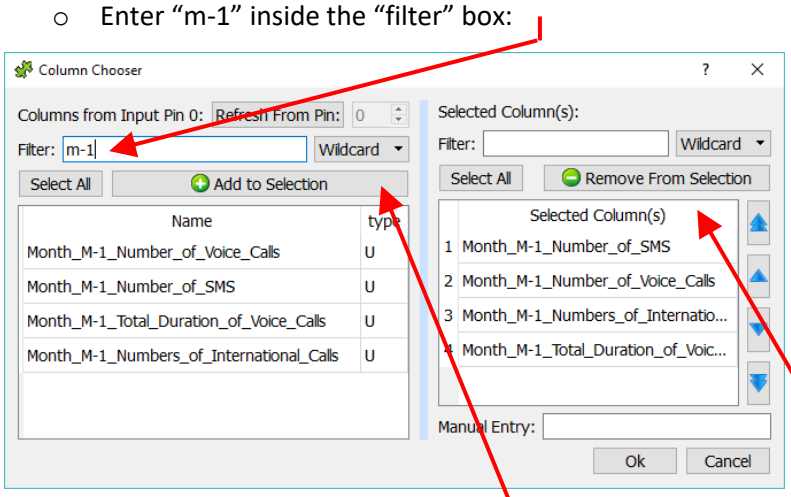
The different variables declared in the left panel of the Vectorized Calculator Action (in the above example: the variables "a" and "b") can either be:

- A list of several columns. All the list that are declared must have the same size. In the example above, the list size is "4", and there are two lists ("a" and "b"):

	Selected Column(s)
1	Month_M-1_Number_of_SMS
2	Month_M-1_Number_of_Voice_Calls
3	Month_M-1_Numbers_of_Internati...
4	Month_M-1_Total_Duration_of_Voic...

	Selected Column(s)
1	Month_M-2_Number_of_SMS
2	Month_M-2_Number_of_Voice_Calls
3	Month_M-2_Numbers_of_Internati...
4	Month_M-2_Total_Duration_of_Voic...

  The procedure to create each of these 2 list is quite simple too: For example, to create the "a" list:
  - Open the "column chooser" Window
  - Enter "m-1" inside the "filter" box:



  - Click the "Add to selection" button
  - Sort the selected columns by clicking on the header of the list, here:
    This last "sort" step is required to ensure the consistency between all the different lists (between the "a" and "b" list), to be sure that the variables/columns are all in the same order in all the lists.

- One unique column.

Here is second example (demonstrating the "one unique column" option):

We start with a table that contains the sales of the 12 months of the year (in absolute value):

Sales_January	Sales_February	Sales_March	Sales_April	Sales_May	Sales_June	Sales_July	Sales_August	Sales_September	Sales_October	Sales_November	Sales_December
20	40	80	30	70	90	80	70	20	30	20	10

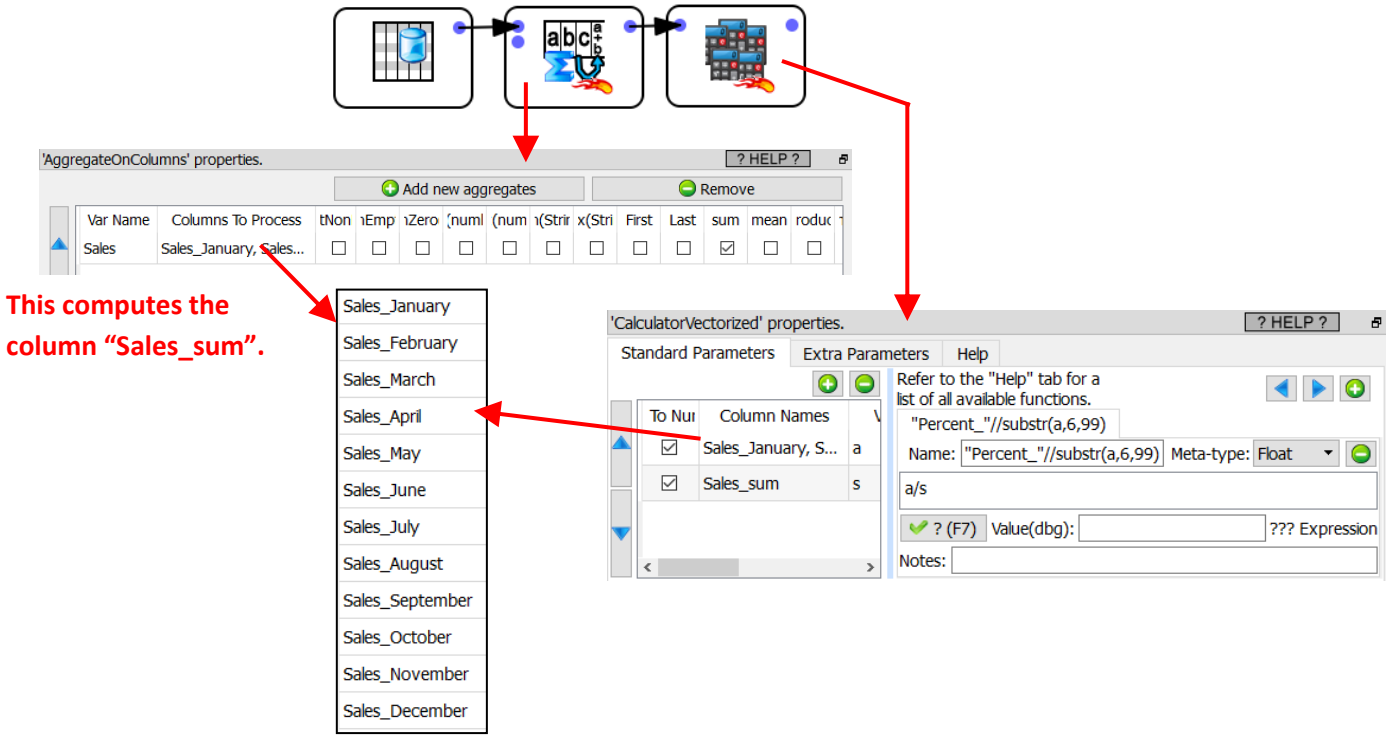
…and we want to convert all these "absolute values" to some percentage.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To arrive to the solution to this small problem, we need two steps:

- Compute the sum of all the sales for the whole year (using the AggregateOnColumns Action)
- Compute the actual percentages (using the Vectorized Calculator Action)

More precisely, we have:



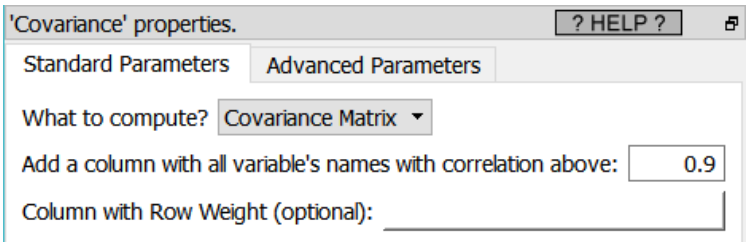
In the above example, the Vectorized Calculator Action adds these columns to the final, result table:

Percent_January	Percent_February	Percent_March	Percent_April	Percent_May	Percent_June	Percent_July	Percent_August	Percent_September	Percent_October	Percent_November	Percent_December
3.57143	7.14286	14.2857	5.35714	12.5	16.0714	14.2857	12.5	3.57143	5.35714	3.57143	1.78571

5.7.11. Covariance (High-Speed action)

Icon:

Property window:

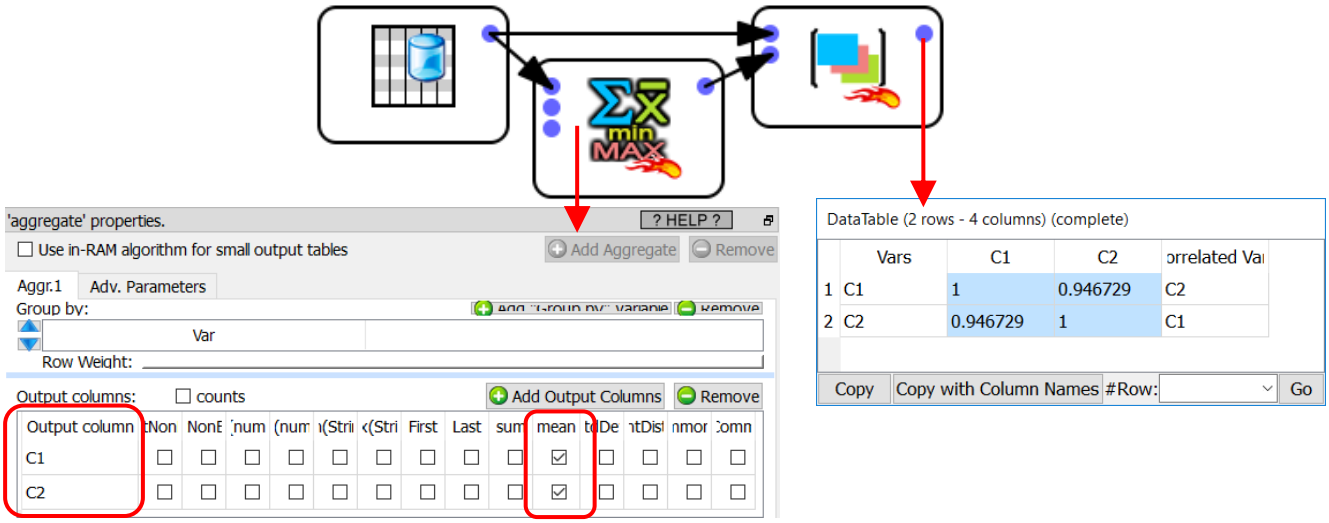


Short description:
Compute a correlation or a covariance matrix.

Long Description:
Compute a correlation or a covariance matrix. This action has two input pins: The first input pin is the Main Data Table. The second input pin defines the columns that are included inside the correlation/covariance matrix: i.e. it contains the mean of the variables that must be included inside the correlation/covariance matrix.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The data preparation consists of a simple aggregate: Just compute the mean of the variables that you want to include inside the correlation matrix:



5.8. TA – Graph Mining (TA=Transformations Actions)

The "Graph Mining Module" included inside Anatella perform **S**ocial **N**etwork **A**nalysis (S.N.A.).

SNA techniques can be applied:

- …in the "telecommunication world", where each arc is a communications between two individuals (i.e. between two phone numbers).
- …in the "banking world" to analyse graphs where each node represents a bank account and where:
  - …each arc is a money transfer between two bank accounts.
  - …an arc between the nodes *i* and *j* represents the fact that the nodes *i* and *j* are:
    - …spending their money in the same shops (on the same hours of the day).
    - …going to the same locations (on the same hours of the day).
- …to "text mining" where each node represents a word and where the weight $W_{i,j}$ of the arc between the (nodes/words) *i* and *j* is the number of times that the (nodes/words) *i* and *j* are cited in the same document. The detected communities are the different "topics" of the text corpus.
- …in any field of commerce or science where a graph exists.

Using SNA techniques, you can extract valuable metrics out of your graphs.
These SNA metrics can be used for:

- Advanced Segmentation of your customer base
- Improved targeting
  - mainly for churn prediction models.
  - …also for cross-sell and up-sell models.
- Detection of Influencers for:
  - viral marketing,
  - refer-a-friend campaigns,
  - new customer acquisition
- Optimization of the service level depending on the value of the subscriber (i.e. more valuable&important subscribers receive better services).
- Next generation recommendation engine.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

All modern SNA algorithms are very efficient and can process extremely large social networks (composed of several dozen millions of nodes) in less than an hour. Thus, during a typical SNA analysis session, most of the processing time is this spent in the computation of the arc weights (and not in the actual SNA algorithms).

We strongly advise you to use an efficient data manipulation tool to compute all the arc weights. Indeed, when performing SNA analysis, you cannot use any sampling technique to reduce the computation time (because sampling will remove some arcs and it will completely destroy the original nature of the graph). Since sampling is forbidden, you need an efficient (i.e. fast) tool to be able to manipulate your graph in a reasonable amount of time. This is why we strongly encourage you to use Anatella to prepare your graphs, otherwise you might end-up "blocked" by the long CPU-time required to create your graphs.

To use efficiently the computed SNA metrics, we suggest that you use the Award-winning predictive engine "TIMi Modeler" included inside the "TIMi Suite".

In the "Telecom World", we created a solution named "LinkAlytics". "LinkAlytics" makes extensive use of the SNA modules included inside Anatella. "LinkAlytics" is a set of Anatella-data-transformation-graphs that uses SNA algorithms (i.e. graph mining) to create very accurate predictive models for churn, cross-sell & up-sell problems for Telecoms. "LinkAlytics" starts from raw (binary) CDR logs, create a 360° customer view (including around one thousand of SNA-based variables) and finally create many predictive models. Everything is automated and runs in a few hours every day.

The Anatella-Graph-Mining module is composed of:

- All the Community Detection Algorithms: included in the Action:
- The Advanced Social Leader Detection Algorithm: included in the Action:
- The Louvain Social Leader Detection Algorithm: included in the Action:
- The Core Number of each node: included in the Action:
- All the algorithms to detect the Business Leaders: included in the Action:
- An algorithm to compute the significance of each arc: included in the Action:
- An algorithm that simulates the propagation of a disease inside a network:

There are, basically, 6 metrics that can be extracted:

- Social Communities
- Social Leaders ("In-Betweeness" centrality, Louvain Leaders, Core Numbers).
- Influencers.
- Business Leader.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Arc significance.
- Disease propagation.
- How to code you own SNA algorithms in C++.

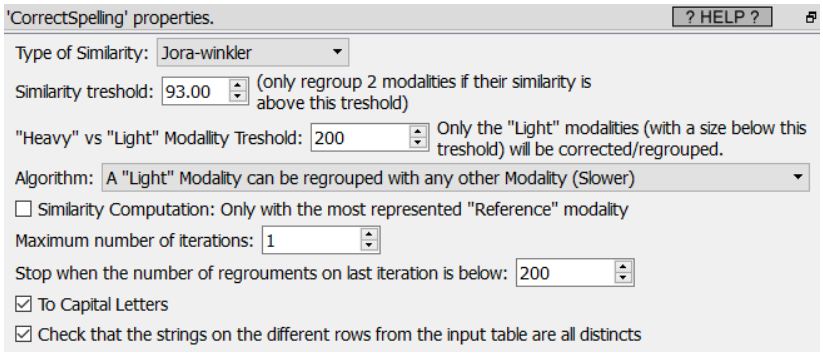
The graph mining module inside Anatella has its own, separate documentation: See the document named “Anatella_GraphMining.pdf”.

5.9. TA – Text Mining (TA=Transformations Actions)

5.9.1. Correct Spelling (High-Speed action)

Icon:

Property window:



Short description:
Correct Spelling-Mistakes in text fields.

Long Description:
Anatella include an operator that checks & corrects the spelling mistakes in any text field. For example, let’s assume that your database contains a field named “City of Birth”. This field will usually contains many different orthography (i.e. spelling error) of the same city. The CorrectSpelling Action will detect and correct these errors automatically. It’s typically used to “clean” the database to get better reports, better predictive models, etc.

For example, the city "RIO DE JANEIRO" can be mis-spelled in a number of different ways (this is a real-world example):

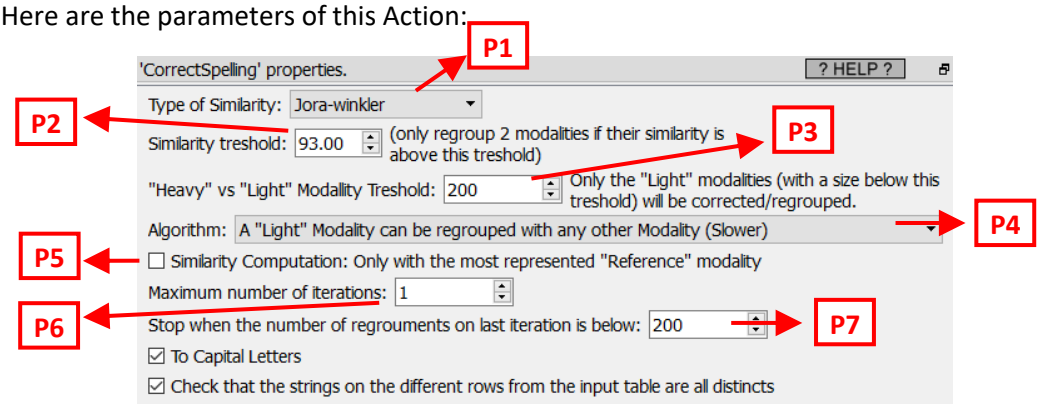
RIO DXE JANEIRO, RIO DE JAEIRO, RIOP DE JANEIRO, RIO NDE JANEIRO, RIO DEJANEIRO, `RIO DE JANEIRO, RIO DE JANIRO, RIO DE JANEI RO, RIO DE JANEIRIO, RI0 DE JANEIRO, RIO DE JNEIRO, RIO DE JANEEIRO, RIO DE JANEIROO, RIO DE JANAEIRO, RIO DE JANEIROR, RIO DE JANEIRO RJ

This action can operate in two different modes:

1. You don’t have any reference table.
2. You have a reference table (For example, you have a table that contains the exact orthography of all the possible city names).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

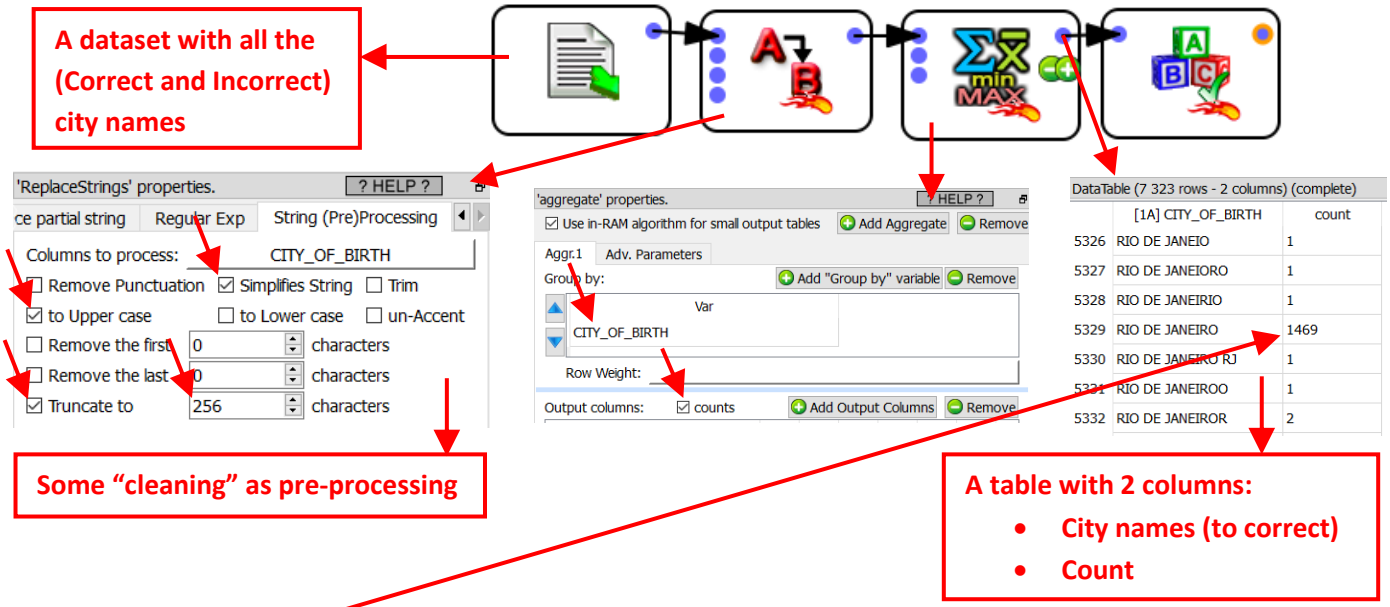
Here are the parameters of this Action:



5.9.1.1. Mode 1: No reference Table

Let's go back the "city name" example: i.e. We have a table with different city names, some of them have some spelling mistakes and we want to correct these mistakes.

We'll typically have such a graph:



Some modalities (i.e. some strings) occurs very often (such as the string "RIO DE JANEIRO" that occurs 1469 times). These very "heavy" strings are assumed to contain the correct spelling of the city name.

Some other modalities practically never appears inside the text corpus (such as the string "RIO DE JANEIORO" that occurs only one time). These very "Light" strings (in opposition to the "heavy" strings) could easily be spelling errors. The threshold between a "Light" and a "Heavy" modality is given in parameter **P3**.

A given string is a spelling error (that must be corrected) if:

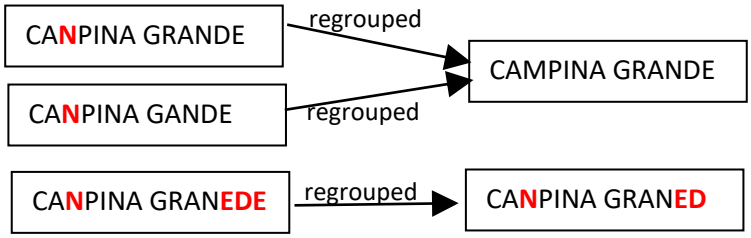
1. It's a "Light" modality.
2. It's "*very similar*" to (i.e. it "looks like") another string X.
   Depending on the parameter **P4**, the other string X can either be:
   - A "heavy" string (This is, by far, the fastest option: Use this option if you have performance issue and the running-time is too long).
   - Any string slightly heavier than the string currently being "corrected".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameters **P1** and **P2** defines if two strings must be declared "*very similar*".

What's happening with the cities that have a small population in real-life? All these cities will only have a small representation inside the database (i.e. they will never be classified as "heavy" modalities). To still be able to correct the spelling mistakes of these "small cities", you must set the parameter **P4** to the value "*regroup with any other modality*" (this is actually the best choice, but it's much slower).

All the strings that are spelling errors are "regrouped" with the string that is the most likely to have the correct spelling (i.e. with the most represented string). So, we can have the following re-groupings:

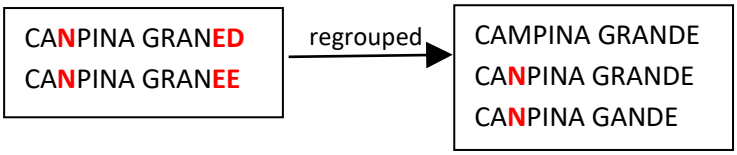


Hereabove, we see that many people make the same spelling mistake: i.e. They replace the "M" letter with the "N" letter inside the word "CAMPINA". Let's now assume that, after the corrections/regrouping illustrated above, we find the following two new "groups":

CAMPINA GRANDE
CANPINA GRANDE
CANPINA GANDE

CANPINA GRANED
CANPINA GRANEDE

We see that, amongst the many people that erroneously write CANPINA (instead of the correct orthography "CAMPINA"), there are some people that don't manage to write correctly the word "GRANDE" neither (i.e. they erroneously write "GRANED" or "GRANEDE" instead). The two groups illustrated above are quite similar because CAMPINA is written with exactly the same spelling mistake ("CANPINA") in both groups. Thus, it would be nice if these 2 groups would new be merged together in a second iteration (because they most likely represent the same city "CAMPINA GRANDE"):



Such that we finally obtain one final group:

CAMPINA GRANDE
CANPINA GRANDE
CANPINA GANDE
CANPINA GRANED
CANPINA GRANEDE

Note that these 5 strings could not be "regouped" together at the first iteration (for example, because "CANPINA GRANED" is "too far away" from "CAMPINA GRANDE") …But the group containing "CANPINA GRANED" is similar enough to the group containing "CAMPINA GRANDE", so that it's still possible to regroup these 2 groups in a later iteration.

We can now give more details about the spelling-correction-algorithm used in Anatella:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The algorithm is:

1. Assign all initial strings into different groups (one string per group).
2. Consider all the "Light Groups" (i.e: the "Light Groups" are defined by "*the sum of the occurrence of all the string's weight inside a "Light Group" is below the threshold given in parameter **P3***") in the order from the "lightest" to the "heaviest": A given group can be merged with another group X if it's "*very similar*" to this other group X.
   The candidate groups X are selected based on the value of parameter **P4**.
   Depending on the parameter **P5**, the similarity between two groups is …:
   - **P5 is not checked:** … the average similarity between all the string-pairs.
     This means that, in the above example, the similarity between our two groups is the average of the similarity between the following string-pairs:

     (CA**N**PINA GRAN**ED** , CAMPINA GRANDE)
     (CA**N**PINA GRAN**ED** , CA**N**PINA GRANDE)
     (CA**N**PINA GRAN**ED** , CA**N**PINA GANDE)
     (CA**N**PINA GRAN**EE** , CAMPINA GRANDE)
     (CA**N**PINA GRAN**EE** , CA**N**PINA GRANDE)
     (CA**N**PINA GRAN**EE** , CA**N**PINA GANDE)

     This is the best option, although it's very slow.

   - **P5 is checked:** … the similarity between the most represented string in each group.
     This means that, in the above example, the similarity between our two groups is the similarity between the following 2 strings:
     (CA**N**PINA GRAN**ED** , CAMPINA GRANDE)
3. Decide if another iteration is required: If yes, go back to step 2.

The parameters **P6** and **P7** are used to decide how many iterations the algorithm does. Most of the time, one iteration is enough. Some time, two iterations can be useful (but the second iteration usually takes a huge amount of time).

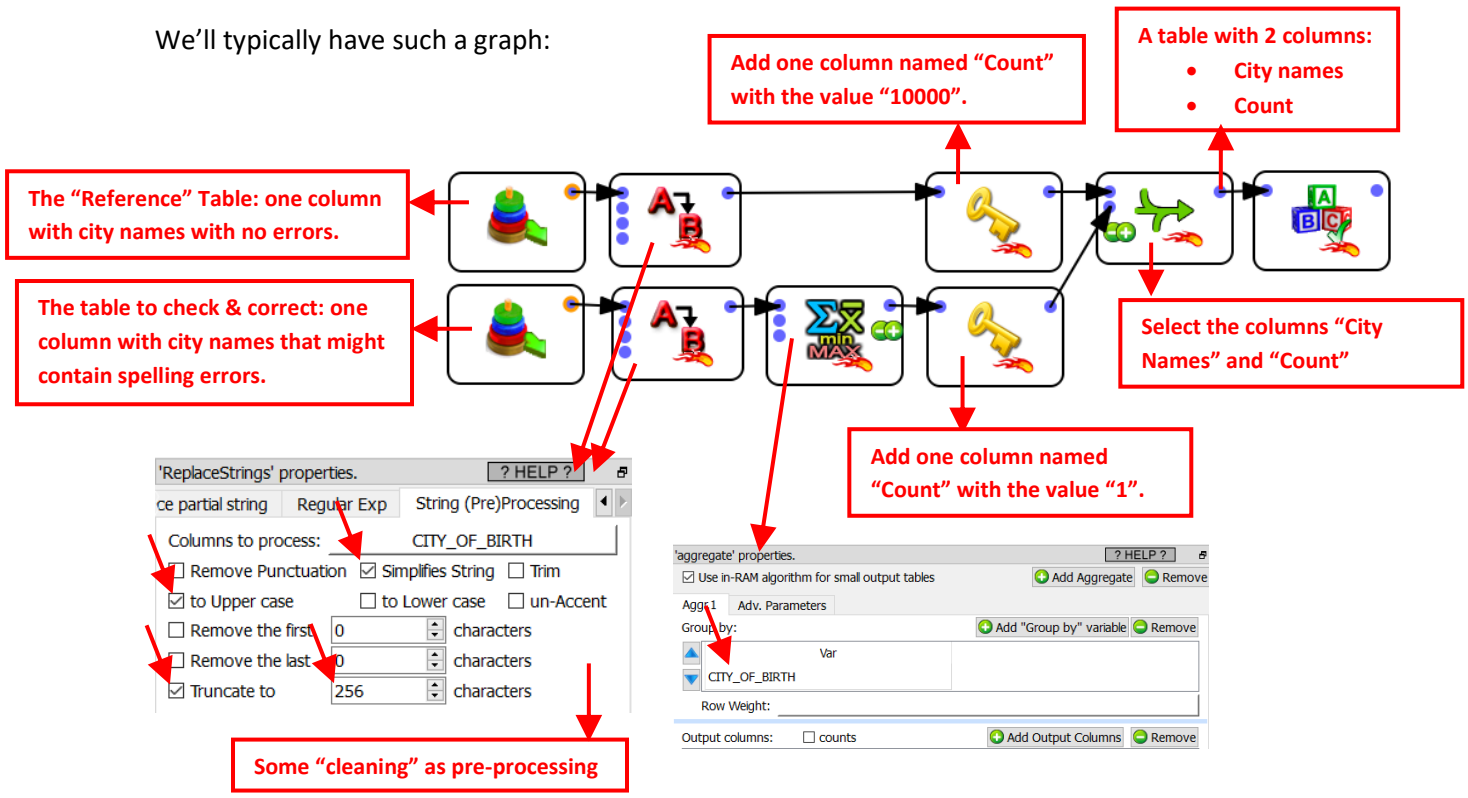
5.9.1.2. Mode 2: We have a reference Table

Let's go back the "city name" example: i.e. We now have two tables that contains city names:

- The first table contains no errors inside the city names: This is the reference table.
- The second table contains city names that might have some spelling mistakes and we want to correct these mistakes.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We'll typically have such a graph:



As input to the CorrectSpelling Action, we have such a table:

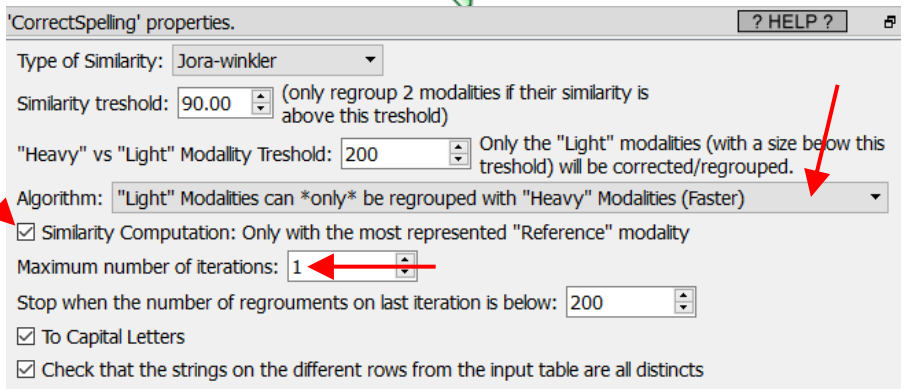
	CITY_OF_BIRTH	[19] count
817	IBIRAPITANGA	10000
818	PARACATU	10000
819	CACOAL	10000
820	FLORANIA	10000
821	CURURUPU	1
822	ALAGOINHA	1
823	SIMOES FILHO	1

From the "Reference" Table

From the table to check & correct

All the cities from the reference table have a "heavy weight" (greater than the parameter **P3**) such that no attempt to "correct" them will be made.

When using a reference table, the parameters from the CorrectSpelling Action are, typically:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When using a reference table, we'll always have:

- Parameter **P4** is "*Only regroup with "Heavy" Modalities*"
- Parameter **P5** is checked
- Parameter **P6** is equal to 1

5.9.1.3. Which "Similarity Measure" should I use?

There are 3 options for the "Similarity Measure":

1. Damereau-Levenstein
2. Jora-Winkler
3. Dice Coefficient

The "Damereau-Levenstein" similarity measure is derived from the "Damereau-Levenstein" distance between two strings. This distance is defined in the following way:

> *The "Damereau-Levenstein" distance between 2 strings is the minimum number of operations needed to transform one string into the other, where an operation is defined as an insertion, deletion, or substitution of a single character, or a transposition of two adjacent characters.*

We derive the "Damereau-Levenstein" similarity from the "Damereau-Levenstein" distance:

$$Similarity_{\text{Damereau-Levenstein}}(s_1, s_2) = 1 - \frac{MaxDist(s_1, s_2) - Distance_{\text{Damereau-Levenstein}}(s_1, s_2)}{MaxDist(s_1, s_2)}$$

$$MaxDist(s_1, s_2) = \min(12, \max(4, \frac{\max(length(s_1), length(s_2))}{2}))$$

Here are some examples:

- The distance between "BRUSELS" (that we'll define as the reference value) and "BRUSSSEL" is 3 because there are three errors:
  - there are three "S" in the middle instead of one (2 insertions= 2 errors)
  - the final "S" is missing.
- The distance between "BRUSELS" (once again, the reference value) and "BRUGELS" is 1 because there is only one error:
  - The "S" letter is replaced by the "G" letter

Let's look at the above two examples: The "Damereau-Levenstein" distance finds that "BRUGELS" is closer to "BRUSELS" than "BRUSSSEL"?? There is something very wrong about that. The example illustrates that the "Damereau-Levenstein" distance…

1. …is worthless to compare strings when the distances between the strings might go above 1.
2. …is ignoring the phonetic content of the characters. If you are interested in a phonetic encoding, please refer to the section 5.9.2.

The Jaro-winkler similarity is an attempt to improve on the idea of the "Damereau-Levenstein" similarity. The Jaro-winkler similarity tries to stay meaningfull in the case where the "Damereau-Levenstein" distance fails: i.e. when the distance between the strings might go above 1. Extensive tests demonstrates that, in practically all real-world studies, the Jaro-winkler similarity outperforms the "Damereau-Levenstein" similarity. The exact definition of the Jaro-winkler similarity is well documented on internet and will not be reproduced here.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The documentation about the "Jaro-winkler similarity" is available on Wikipedia on a page named "Jaro-winkler ***distance***". This is somewhat disturbing because it's actually really a similarity measure (the higher the value, the closer the two strings are) and not a distance measure.

Although the Jaro-winkler similarity is better than the "Damereau-Levenstein" similarity, it's still limited to "small" strings (for example: city names, surnames, etc.) where the number of differences between the compared strings stays relative small because of the small sizes of the strings (After all, the "Damereau-Levenstein" is just an improvement on the "Damereau-Levenstein" similarity and finally suffers from the same defaults). Thus, if you want to compute similarities between very long strings (such as "street names", long "product names", long "SKU names", long "Book titles", etc.), you'd better use the "Dice Coefficient" (the "Dice Coefficient" is also sometime named the "Pair letters similarity").

The "Dice Coefficient" is a similarity metric that rewards both common substrings and a common ordering of those substrings. The definition is:

$$DiceCoeffient(s_1, s_2) = \frac{2 \times (\text{number of common letter-pairs between } s_1 \text{ and } s_2)}{(\text{number of letter-pairs in } s_1) + (\text{number of letter-pairs in } s_2)}$$

The intention is that by considering adjacent characters (i.e. character-pairs), we take into account not only of the common characters, but also of the character ordering in the two strings, since each character pair contains a little information about the ordering.

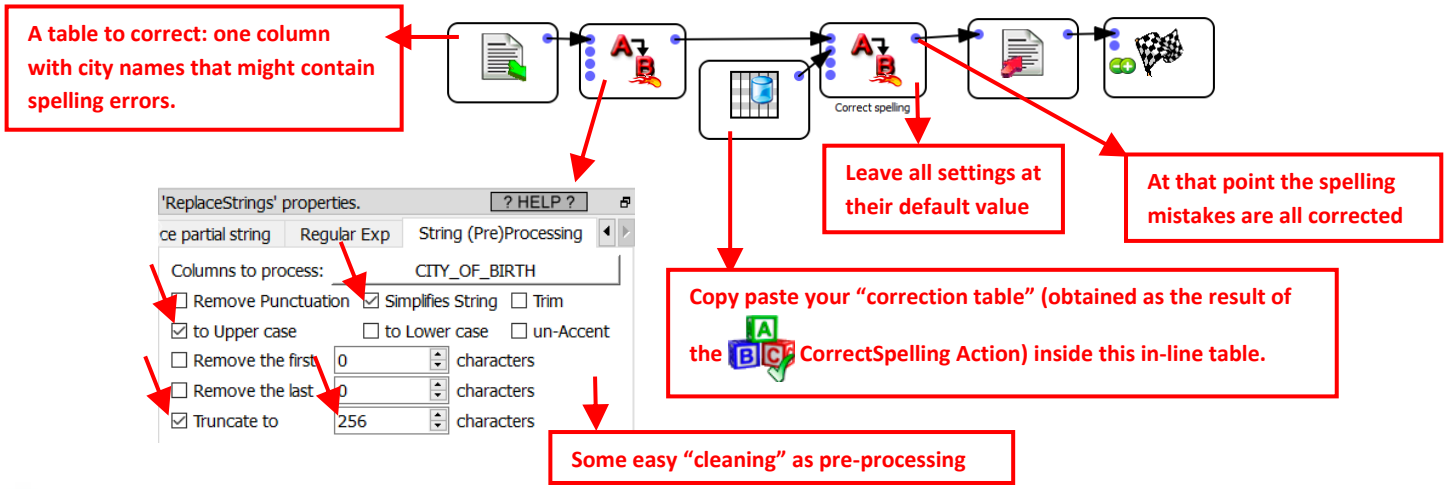
Here is an example:

$$DiceCoeffient("FRANCE","FRENCH") = \frac{2 \times |(FR, NC)|}{|(\text{FR,RA,AN,NC,CE})| + |(FR, RE, EN, NC, CH)|} = \frac{2 \times 2}{5+5} = 0.4$$

Inside the Anatella implementation of the Dice Coefficient, all the letter-pairs that contains a space character are discarded before any computation. This means that the similarity between the strings "vitamin B" and "vitamin C" is 100%. In this case (when you are interested in correcting small differences), you should rather use the "Damereau-Levenstein" or "Jaro Winkler" similarity.

5.9.1.4. How to use the results?

The output of the CorrectSpelling Action is a table that contains all the corrections to apply on the data to "clean it" completely. The easiest way to use this "Correction table" is the following graph:

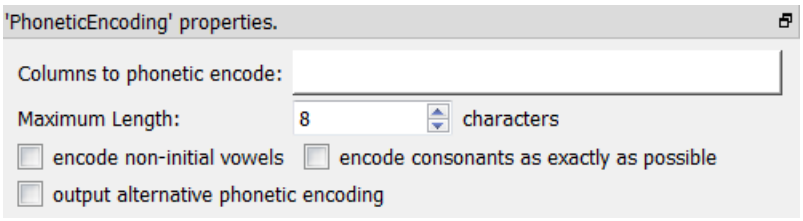


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.9.2. Phonetic Encoding (High-Speed action)

Icon:

Property window:



Short description:
State-of-the-Art Phonetic encoder.

Long Description:

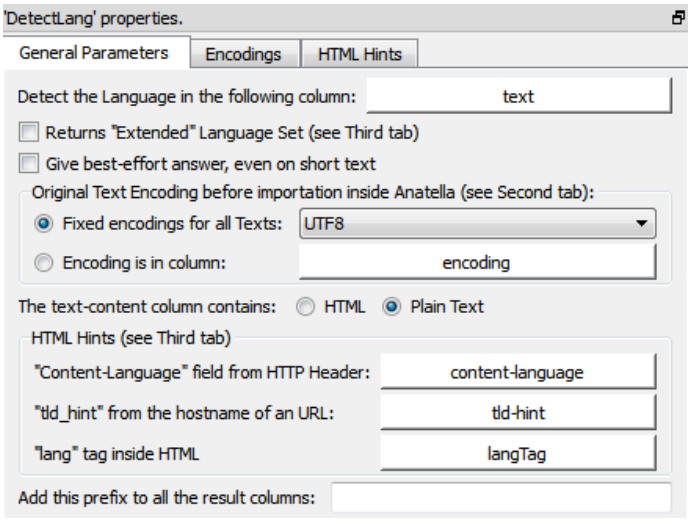
In opposition to the CorrectSpelling Action, the PhoneticEncoder Action is typically not used to "clean" the database: It rather allows you to "bypass" data quality issues when performing (for example) table-joins. This Phonetic Encoder is used, for example, inside Google Refine (The data quality tool from Google).

The typical usage of a phonetic encoder is the following: You have 2 tables that contains different piece of information about the same individuals. You want to create one unified view of these 2 tables (i.e. you want to join the 2 table into one). You must use the individual's names to join 2 rows (from the 2 tables) together but these names contains some miss-spellings (i.e. the primary keys are individual's names). To bypass these spelling errors***, you can join the 2 tables using the phonetic encoding of the names (rather than simply using the plain names)***. You thus need a phonetic encoder that is able to understand how names written in English, Dutch, French, German, Spanish are pronounced. The Metaphone 3 encoder (i.e. the Phonetic Encoder used inside Anatella) is one of the very few phonetic encoder (maybe the only one) that is able to "understand" a very wide variety of languages automatically (other encoders are limited to one language only). It's thus the perfect encoder for such a task…

5.9.3. Detect Language (High-Speed action)

Icon:

Property window:



Short description:
Detect the Language of a text field.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
The engine used to detect the Language of a text field is the same engine used to detect the language of a webpage inside Google Chrome. The engine's name is "CLD2".

An official comparative study between many Language Detection Engine demonstrates that "*...across the full set of 65 languages, CLD2 is the single best-performing system*". This comparative study is: "*Accurate Language Identification of Twitter Messages*" from Marco Lui and Timothy Baldwin (NICTA VRL - Department of Computing and Information Systems - University of Melbourne, VIC 3010, Australia)

The supported languages are (name followed by ISO code in parenthesis):

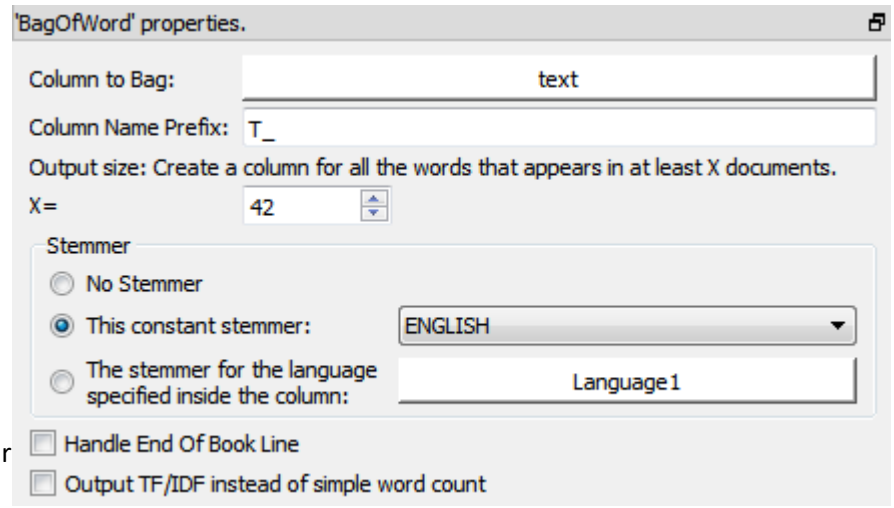
ENGLISH(en); DANISH(da); DUTCH(nl); FINNISH(fi); FRENCH(fr); GERMAN(de); HEBREW(iw); ITALIAN(it); JAPANESE(ja); KOREAN(ko); NORWEGIAN(no); POLISH(pl); PORTUGUESE(pt); RUSSIAN(ru); SPANISH(es); SWEDISH(sv); CHINESE(zh); CZECH(cs); GREEK(el); ICELANDIC(is); LATVIAN(lv); LITHUANIAN(lt); ROMANIAN(ro); HUNGARIAN(hu); ESTONIAN(et); TG_UNKNOWN_LANGUAGE(xxx); UNKNOWN_LANGUAGE(un); BULGARIAN(bg); CROATIAN(hr); SERBIAN(sr); IRISH(ga); GALICIAN(gl); TAGALOG(tl); TURKISH(tr); UKRAINIAN(uk); HINDI(hi); MACEDONIAN(mk); BENGALI(bn); INDONESIAN(id); LATIN(la); MALAY(ms); MALAYALAM(ml); WELSH(cy); NEPALI(ne); TELUGU(te); ALBANIAN(sq); TAMIL(ta); BELARUSIAN(be); JAVANESE(jw); OCCITAN(oc); URDU(ur); BIHARI(bh); GUJARATI(gu); THAI(th); ARABIC(ar); CATALAN(ca); ESPERANTO(eo); BASQUE(eu); INTERLINGUA(ia); KANNADA(kn); PUNJABI(pa); SCOTS_GAELIC(gd); SWAHILI(sw); SLOVENIAN(sl); MARATHI(mr); MALTESE(mt); VIETNAMESE(vi); FRISIAN(fy); SLOVAK(sk); CHINESE_T(zh-Hant); FAROESE(fo); SUNDANESE(su); UZBEK(uz); AMHARIC(am); AZERBAIJANI(az); GEORGIAN(ka); TIGRINYA(ti); PERSIAN(fa); BOSNIAN(bs); SINHALESE(si); NORWEGIAN_N(nn); XHOSA(xh); ZULU(zu); GUARANI(gn); SESOTHO(st); TURKMEN(tk); KYRGYZ(ky); BRETON(br); TWI(tw); YIDDISH(yi); SOMALI(so); UIGHUR(ug); KURDISH(ku); MONGOLIAN(mn); ARMENIAN(hy); LAOTHIAN(lo); SINDHI(sd); RHAETO_ROMANCE(rm); AFRIKAANS(af); LUXEMBOURGISH(lb); BURMESE(my); KHMER(km); TIBETAN(bo); DHIVEHI(dv); CHEROKEE(chr); SYRIAC(syr); LIMBU(lif); ORIYA(or); ASSAMESE(as); CORSICAN(co); INTERLINGUE(ie); KAZAKH(kk); LINGALA(ln); PASHTO(ps); QUECHUA(qu); SHONA(sn); TAJIK(tg); TATAR(tt); TONGA(to); YORUBA(yo); MAORI(mi); WOLOF(wo); ABKHAZIAN(ab); AFAR(aa); AYMARA(ay); BASHKIR(ba); BISLAMA(bi); DZONGKHA(dz); FIJIAN(fj); GREENLANDIC(kl); HAUSA(ha); HAITIAN_CREOLE(ht); INUPIAK(ik); INUKTITUT(iu); KASHMIRI(ks); KINYARWANDA(rw); MALAGASY(mg); NAURU(na); OROMO(om); RUNDI(rn); SAMOAN(sm); SANGO(sg); SANSKRIT(sa); SISWANT(ss); TSONGA(ts); TSWANA(tn); VOLAPUK(vo); ZHUANG(za); KHASI(kha); SCOTS(sco); GANDA(lg); MANX(gv); MONTENEGRIN(sr-ME); AKAN(ak); IGBO(ig); MAURITIAN_CREOLE(mfe); HAWAIIAN(haw); CEBUANO(ceb); EWE(ee); GA(gaa); HMONG(hmn); KRIO(kri); LOZI(loz); LUBA_LULUA(lua); LUO_KENYA_AND_TANZANIA(luo); NEWARI(new); NYANJA(ny); OSSETIAN(os); PAMPANGA(pam); PEDI(nso); RAJASTHANI(raj); SESELWA(crs); TUMBUKA(tum); VENDA(ve); WARAY_PHILIPPINES(war); NDEBELE(nr); X_BORK_BORK_BORK(zzb); X_PIG_LATIN(zzp); X_HACKER(zzh); X_KLINGON(tlh); X_ELMER_FUDD(zze); X_Common(xx-Zyyy); X_Latin(xx-Latn); X_Greek(xx-Grek); X_Cyrillic(xx-Cyrl); X_Armenian(xx-Armn); X_Hebrew(xx-Hebr); X_Arabic(xx-Arab); X_Syriac(xx-Syrc); X_Thaana(xx-Thaa); X_Devanagari(xx-Deva); X_Bengali(xx-Beng); X_Gurmukhi(xx-Guru); X_Gujarati(xx-Gujr); X_Oriya(xx-Orya); X_Tamil(xx-Taml); X_Telugu(xx-Telu); X_Kannada(xx-Knda); X_Malayalam(xx-Mlym); X_Sinhala(xx-Sinh); X_Thai(xx-Thai); X_Lao(xx-Laoo); X_Tibetan(xx-Tibt); X_Myanmar(xx-Mymr); X_Georgian(xx-Geor); X_Hangul(xx-Hang); X_Ethiopic(xx-Ethi); X_Cherokee(xx-Cher); X_Canadian_Aboriginal(xx-Cans); X_Ogham(xx-Ogam); X_Runic(xx-Runr); X_Khmer(xx-Khmr); X_Mongolian(xx-Mong); X_Hiragana(xx-Hira); X_Katakana(xx-Kana); X_Bopomofo(xx-Bopo); X_Han(xx-Hani); X_Yi(xx-Yiii); X_Old_Italic(xx-Ital); X_Gothic(xx-Goth); X_Deseret(xx-Dsrt); X_Inherited(xx-Qaai); X_Tagalog(xx-Tglg); X_Hanunoo(xx-Hano); X_Buhid(xx-Buhd); X_Tagbanwa(xx-Tagb); X_Limbu(xx-Limb); X_Tai_Le(xx-Tale); X_Linear_B(xx-Linb); X_Ugaritic(xx-Ugar); X_Shavian(xx-Shaw); X_Osmanya(xx-Osma); X_Cypriot(xx-Cprt); X_Braille(xx-Brai); X_Buginese(xx-Bugi); X_Coptic(xx-Copt); X_New_Tai_Lue(xx-Talu); X_Glagolitic(xx-Glag); X_Tifinagh(xx-Tfng); X_Syloti_Nagri(xx-Sylo); X_Old_Persian(xx-Xpeo); X_Kharoshthi(xx-Khar); X_Balinese(xx-Bali); X_Cuneiform(xx-Xsux); X_Phoenician(xx-Phnx); X_Phags_Pa(xx-Phag); X_Nko(xx-Nkoo); X_Sundanese(xx-Sund); X_Lepcha(xx-Lepc); X_Ol_Chiki(xx-Olck); X_Vai(xx-Vaii); X_Saurashtra(xx-Saur); X_Kayah_Li(xx-Kali); X_Rejang(xx-Rjng); X_Lycian(xx-Lyci); X_Carian(xx-Cari); X_Lydian(xx-Lydi); X_Cham(xx-Cham); X_Tai_Tham(xx-Lana); X_Tai_Viet(xx-Tavt); X_Avestan(xx-Avst); X_Egyptian_Hieroglyphs(xx-Egyp); X_Samaritan(xx-Samr); X_Lisu(xx-Lisu); X_Bamum(xx-Bamu); X_Javanese(xx-Java); X_Meetei_Mayek(xx-Mtei); X_Imperial_Aramaic(xx-Armi); X_Old_South_Arabian(xx-Sarb); X_Inscriptional_Parthian(xx-Prti); X_Inscriptional_Pahlavi(xx-Phli); X_Old_Turkic(xx-Orkh); X_Kaithi(xx-Kthi); X_Batak(xx-Batk); X_Brahmi(xx-Brah); X_Mandaic(xx-Mand); X_Chakma(xx-Cakm); X_Meroitic_Cursive(xx-Merc); X_Meroitic_Hieroglyphs(xx-Mero); X_Miao(xx-Plrd); X_Sharada(xx-Shrd); X_Sora_Sompeng(xx-Sora); X_Takri(xx-Takr);

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.9.4. Bag of Word (High-Speed action)

Icon:

Property window:



Short description:
Bag-Of-Word: Structur

Long Description:
The objective is to extract from the text field a table that can be used inside TIMi to create a predictive model.

Typical usage scenariis include building models for:

- Churn/Cross-Selling/Up-Selling modeling: We'll complement the customer profiles by adding some fields (usually around 10.000 new fields) coming from text analytics. The main objective here is to increase the quality of the predictive models (to get more ROI out of it). For example, we can easily build high-accuracy Predictive Models that looks at a text database containing the calls made to your "hot line". If these texts contain words like "disconnections", "noisy", "unreachable", etc. it can be a good indicator of churn (for a telecom operator).
- Document classification: We'll automatically assign tags to documents.

To be able to create predictive models based on text, classical analytical tools are forcing you to define "Dictionnaries". These "Dictionnaries" contain a limited list of words pertaining to the field of interest. Each of these word/concept directly translates to a new variable added to the analytical dataset used for modeling (i.e. to the dataset injected into TIMi). Typically, you can't have more than a few hundreds different words or concept because classical modeling tools are limited in the number of variables that they can handle.

In opposition, TIMi supports a practically unlimited number of variables. This allows us to use a radically different approach to text mining. Instead of limiting the construction of the analytical dataset to a few hundred words/concepts, we'll create an analytical dataset that contains ALL the words/concepts visible inside the text corpus to analyze (and even more, if desired)(To be more precise, we are using the Bag-of-Word principle on the set of all stemmed words of the text corpus. Our stemmer supports a wide variety of languages). This approach has several advantages over the classical approach (used in classical tools such as SAS, SPSS, etc):

- We don't need to create "Dictionnaries" because TIMi will automatically select for us the optimal set of words and concepts that are the best to predict the given target. Since we are using an optimal selection, rather than a hand-made approximative selection, the preditive models delivered by TIMi are more accurate (and thus generates more ROI).
- We gain a large amount of man-hours because it's not required anymore to create "Dictionnaries". "Dictionnaries" are context-sensitive: For each different tasks/predictive target, you need to re-create a new dictionnary (i.e. you need to select manually the columns

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

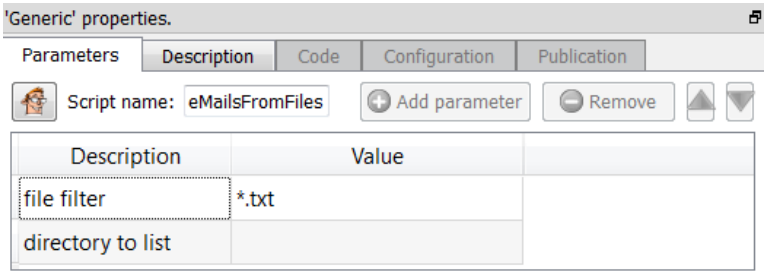
included inside the analytical dataset rather than let TIMi automatically select them for you). Creating new Dictionnaries is a lengthy boring task, easily prone to errors.

- The process is less expensive because you don't need to "buy" these dectionnaries from an external provider: In paticular, SPSS has a big focus on dictionnaries: You can buy pre-made dictionnaries and some SPSS tools are focusing only on the creation of dictionnaries.

5.9.5. E-Mails From Files

Icon:

Property window:



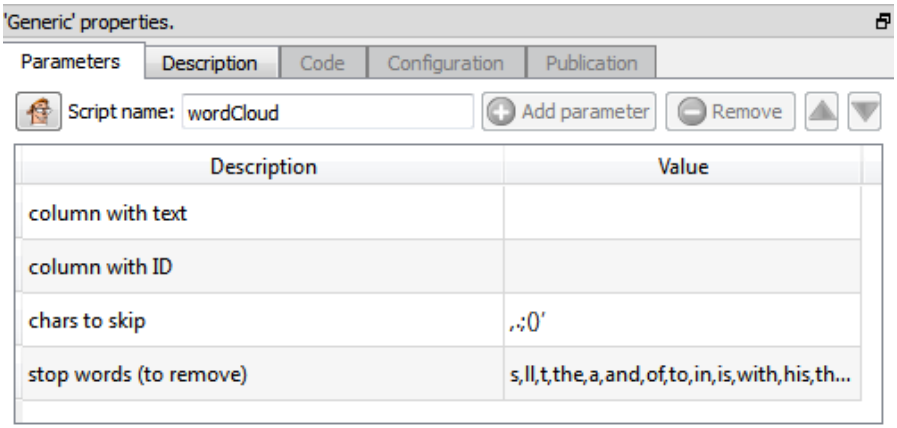
Short description:
Extract the content of emails.

Long Description:
Extract the content of the emails contained inside the classical file structure used by Outlook.

5.9.6. Cloud of words

Icon:

Property window:



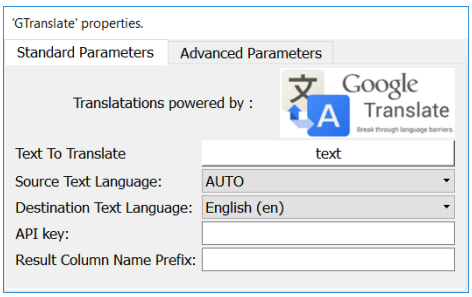
Short description:
Cloud of words.

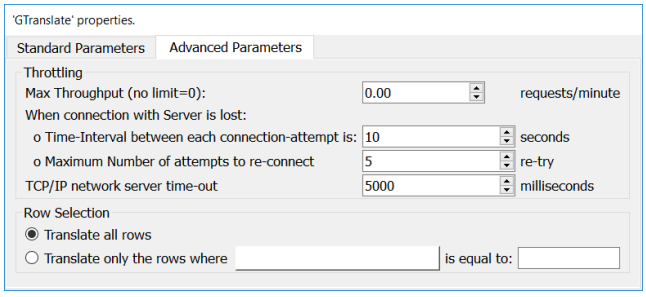
Long Description:
Allow you to create a frequency table for each word. You can thereafter represents the frequency of each word using a simple table or a fancy "Cloud of Words".

5.9.7. Google Translate (High-Speed action)

Icon:

Property window:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

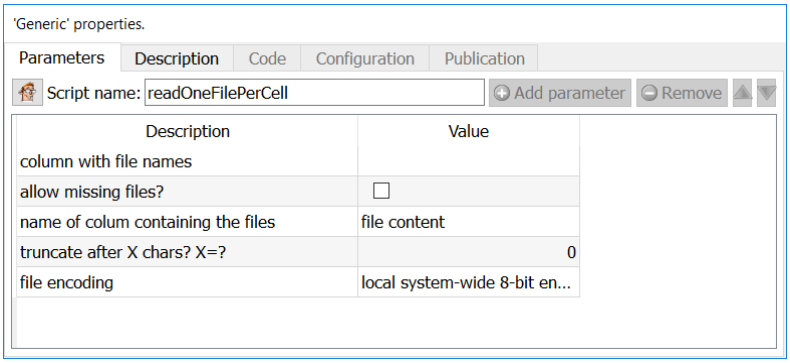
Short description:
Access Google Translate Services

Long Description:
Google Translate is accessible to developpers who have a Google API Key, which can be obtained for free (with some limitations).

5.9.8. Read One File per Cell

Icon:

Property window:



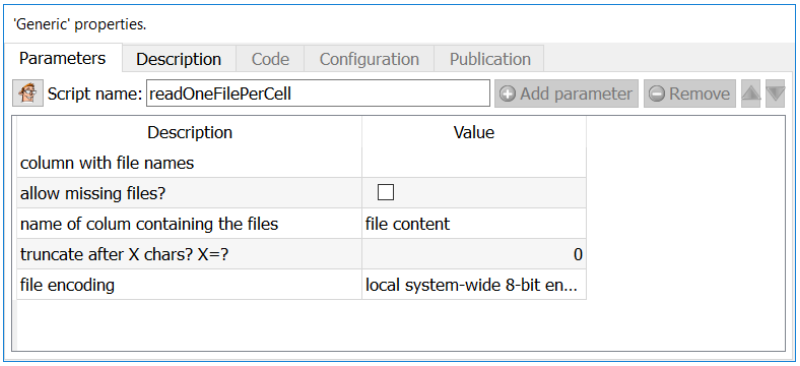
Short description:
Open many text files in separate cells

Long Description:
Allows you to open hundreds or thousands of text files in separate cells. This action is fed by a “List Directory” action in which a column contains the location of each file.

5.9.9. Write One File per Cell

Icon:

Property window:



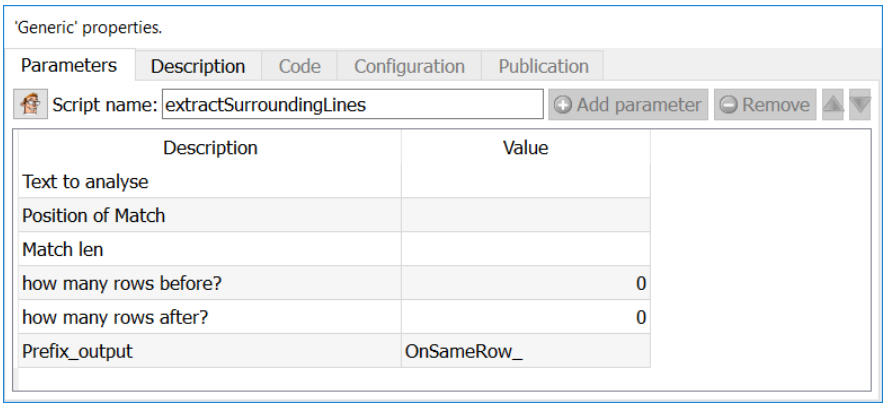
Short description:
write many text files in separate cells

Long Description:
Allows you to save hundreds or thousands of text files in separate cells. This action is fed by a table in which a column contains the location of each file, and a column contains the text to be saved.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.9.10. Extract Surrounding Lines

Icon:



Property window:

Short description:
Extracts surrounding lines

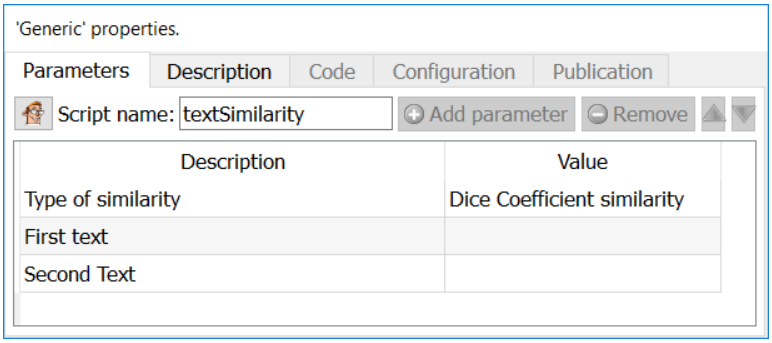
Long Description:
Extracts lines around a specific text. This action is fed by a table containing:

- The text to analyse
- The position of the match (computed with the formula “indexOf)
- The length of the text to match

Simply specify the number of lines before and after the target text, and you’re set.

5.9.11. Text Similarity

Icon:



Property window:

Short description:
Text Similarity

Long Description:
Computes the distance between text using:

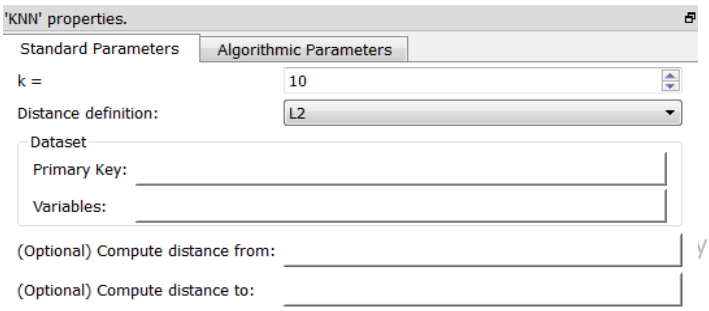
- Damereau Levenstein Similarity
- Jaro Winkler Similarity
- Dice Coefficient
- Damereau Levenstein distance

See the section 5.9.1. Correct Spelling for more information on the topic

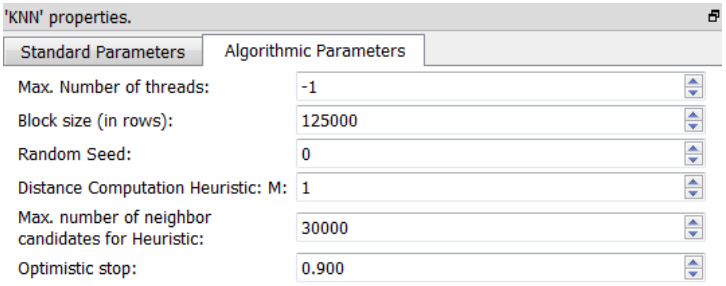
5.10. TA – GIS - Geographical Information System (TA=Transformations Actions)

5.10.1. KNN (High-Speed action)

Icon:



Property window:



Short description:
Solves the K-Nearest-Neighbor problem.

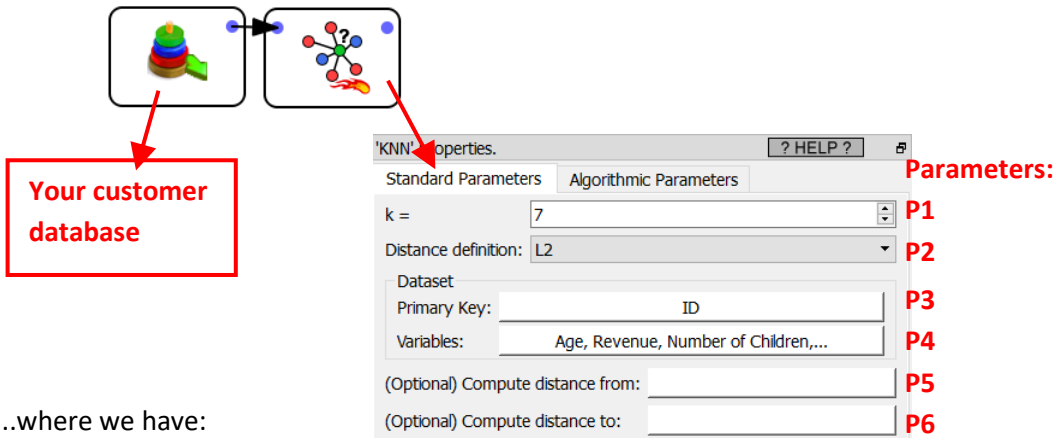
Long Description:

The KNN Action is typically used for the following two use cases:

1. Data imputation

Let's assume that you have a customer database that contains many numerical features (such as "age", "revenue", "Number of Children", "Number of Cars", etc.). So that each customer can be represented by a dot in a space of dimension D (D=number of numerical features inside your customer database). For some customers, some features are missing. You want to "guess" the missing features of a customer X_i by looking at (e.g. most of the time: by "averaging") the characteristics of the k closests neighbors from the customer X_i. The KNN Action finds which customers are actually these k neighbors.

You'll have an Anatella graph such as this one:



..where we have:

- Parameter P1: The number of neighbor(s) k that you want to find.
- Parameter P2: The distance definition. You can choose between:

$$dist_{L2}(X,Y) = \sqrt{\sum_{i=1}^{D}(X_i - Y_i)^2}$$

$$dist_{L1}(X,Y) = \sum_{i=1}^{D}\left|X_i - Y_i\right|$$

$$dist_{\text{near L}_{\text{infinity}}}(X,Y) = 0.9\, dist_{\text{infinity}}(X,Y) + 0.1\, dist_{L1}(X,Y)$$

...where we used: $dist_{\text{infinity}}(X,Y) = \max_i(\left|X_i - Y_i\right|)$

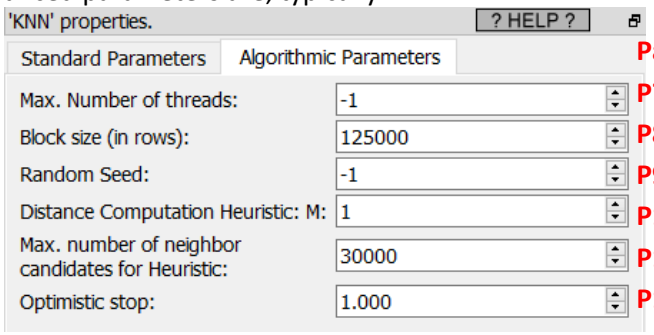
$$dist_{\text{very near L}_{\text{infinity}}}(X,Y) = 0.95\, dist_{\text{infinity}}(X,Y) + 0.05\, dist_{L1}(X,Y)$$

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

$dist_{Earth}(X,Y)$ = Earth-distance in Km (taking into account the curvature of the Earth). X and Y must be GPS coordinates in decimal degree (typically found using the geocode Action).

- Parameter **P3**: A primary key that uniquely identifies the different customers.
- Parameter **P4**: The variables to include inside the distance computations. The dimention D of the search space is defined by the number of variables given in Parameter **P4.**
- Parameters **P5** and **P6** are not used in the current use case (see the next use case about "shop assignment" to have more information about these two parameters). To summarize:
  - Parameter **P5**: A column that contains a 1/0 (true/false) flag. We'll only compute the K-Nearest-Neighbor for the rows where the flag is 1 (true).
  - Parameter **P6**: A column that contains a 1/0 (true/false) flag. The nearest-neighbors given as output of the KNN Action belongs to the set of rows where the flag is 1 (true).

The advanced parameters are, typically:



Before starting any computation, Anatella load into RAM memory all the required features. To load data faster, Anatella allocates memory by large chunks. The chunk size is given in Parameter **P8**.

To decrease computation time: Anatella will use several CPU's in parellel to compute all the distances. The number of CPU used is given in Parameter **P7** (-1=max).

Let's now assume that our customer database contains N customers.

To solve the KNN problem, we could use a "brute-force-approach" algorithm. This algorithm is:

- Loop over all customers inside the database:
  Each iteration computes the K-nearest-Neighbor for a difference customer X_i
  An iteration is:
  - Compute the distance from X_i to all the other customers (there are N-1 such distances) and keep the K customers with the smallest distance.

The "brute-force-approach" is a very slow algorithm because its running time is proportional to N^2 (since there are about N^2 distances to compute).

When the dimension D of the search space (D=number of numerical features inside your customer database) is above 5 or 6, it's very difficult to come with an algorithm that is competitive (in terms of speed) with the simple "brute-force-approach" algorithm (when D is small, "R-trees" are a good family of algorithm). This is why the "brute-force-approach" algorithm is still used most of the time in most KNN tools that regularly deals with high dimentional spaces (D>5).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella does not use a "brute-force-approach" algorithm. The algorithm used to solve the KNN problem in Anatella is optimized for high dimentional spaces (D>5). To reduce computation time, Anatella uses the following algorithm/heuristic:

- Loop over all customers inside the database:
  Each loop iteration computes the K-Nearest-Neighbors for a different customer X_i (i=1,…N)
  The loop is executed in parallel on (Parameter **P7**) CPU's.
  An iteration is:
  - Select quickly a small "candidate neighbor" of X_i.
  - Compute all the distances from X_i to all the customers inside the "candidate neighbor" of X_i.
  - Keep the K customers in the "candidate neighbor" that have the smallest distance to X_i (these are the K customers that are returned as the K-Nearest-Neighbors of X_i).
  - The (Parameters **P9, P10, P11, P12**) are controlling the "candidate neighbor" size and shape.

Here are more informations about the (Parameters **P9, P10, P11, P12**) that control the "candidate neighbor":

- The selection of the "candidate neighbor" includes a random component (the random number generator is controlled using the Parameter **P9**).
- The "candidate neighbor" size is always smaller than (Parameter **P11**).
- The Parameters **P10** and **P12** are documented in more details inside the Anatella Advanced Guide.
- The Parameter **P10** controls the shape of the "candidate neighbor".
  Most of the time, the best (i.e. the fastest) value for the Parameters **P10** is "1".
- The Parameter **P12** controls the size of the "candidate neighbor". The best value (and the safest value) for the parameter **P12** is the maximum value of "1". A smaller value (e.g. 0.9) has the following effect:
  - it leads to a smaller "candidate neighbor" and thus it reduces the computation time (because there are less distances to compute).
  - Anatella might find a wrong answer: i.e. a "too small" value for the parameter **P12** reduces so much the size of the "candidate neighbor" that we can't be assured anymore that the K-Nearest-Neighbors of X_i are actually inside the selected "candidate neighbor". If such disastrous situation occurs, we'll return a wrong answer: i.e. we won't return the exact KNN of X_i but an approximation of it.
- The parameter **P12** (and **P11**) can be used to balance between a smaller computation time and a more precise/exact answer.

The output of the KNN Action is such a table (with K=1 ; only the first 6 rows of the table are shown):

	ID	This_Idx	NN_Idx_1	NN_Dist_1
1	Frank	0	3	0.119231
2	Sabrina	1	170	0.51415
3	David	2	1	1.40924
4	Cassian	3	88	1.05974
5	Darian	4	37	8.5877
6	Max	5	1	2.28103

The column "ID" contains the "customer-identifier-column" selected using the parameter **P3**.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The column "This_Idx" is produced by the KNN Action: This column assigns a unique Key-Number to each different output row. This Key-Number is used inside the column "NN_Idx_1" to indicate who are the closest neighbors.

Let's give some examples:

- Let's look at the first row ("Frank") in the above table:
  On this row, we see "NN_Idx_1"=3: This means that the closest neighbor to "Frank" is the neighbor with "This_Idx"=3: i.e. the closest neighbor to "Frank" is "Cassian" (and the distance between "Frank" and "Cassian" is "NN_Dist_1"=0.119231).
- Let's look at the third row ("David") of the above table:
  On this row, we see "NN_Idx_1"=1: This means that the closest neighbor to "David" is the neighbor with "This_Idx"=1: i.e. the closest neighbor to "David" is "Sabrina" (and the distance between "David" and "Sabrina" is "NN_Dist_1"=1.40924).

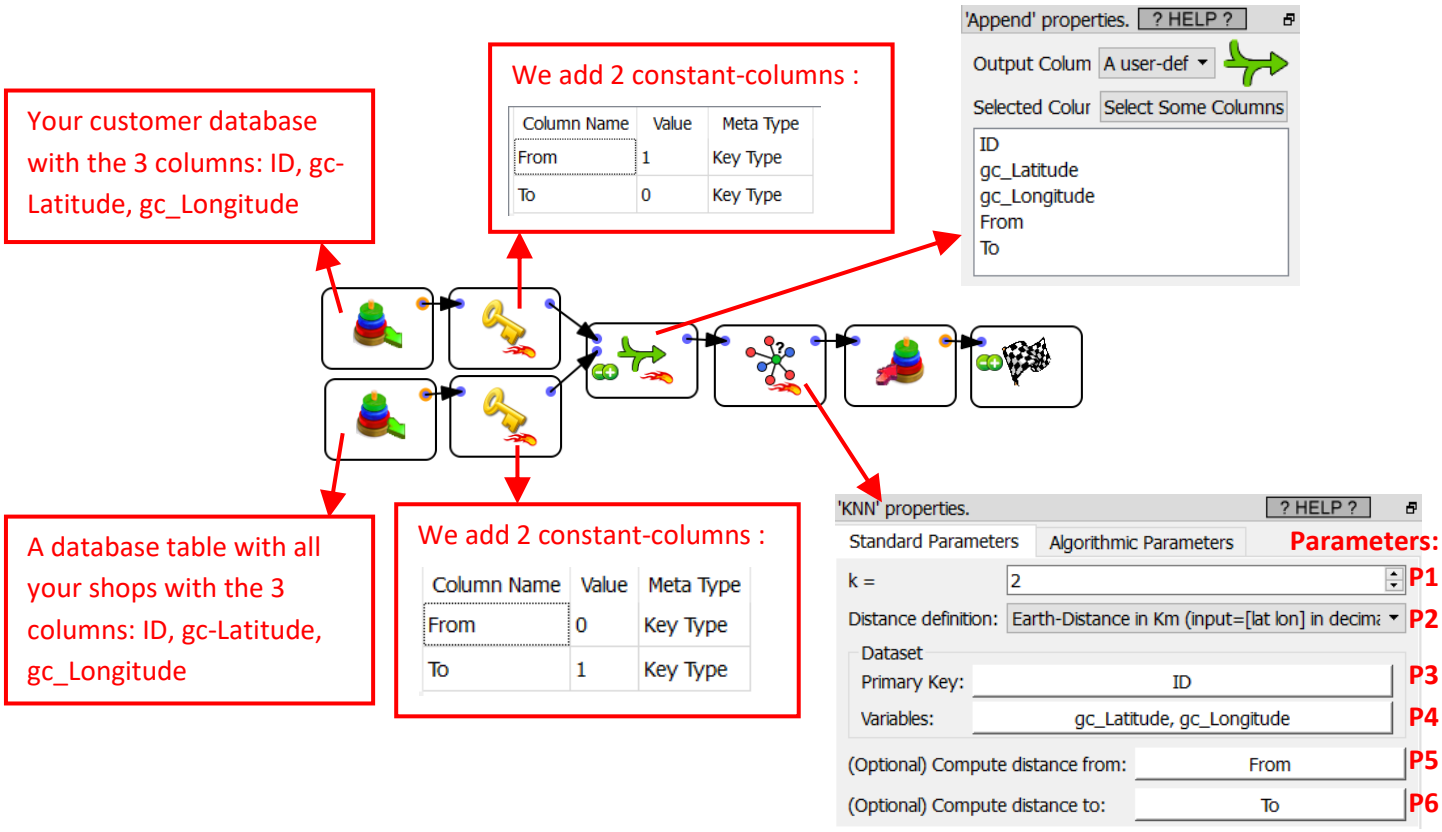
2. Shop assignment/Recommendation

Let's now assume that you have the GPS coordinates of your shops and of your customer's homes (you can easily compute these coodinates using the geocode Action).

The objective of this use case is:

For each of your customers, you want to compute their 2 closest shops.

We'll have an Anatella graph such as this one:



Please note that we selected:

- Parameter **P1**: K=2: We only want to get the 2 closest shops.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Parameter **P2**: Earth Distance
- Parameter **P4**: The 2 columns that contains the latitude and the longitude **in that order**.
- Parameter **P5**: A column that contains a 1/0 (true/false) flag. Anatella only computes the K-Nearest-Neighbor for the rows where this flag is 1 (true). The Parameter **P5** is 1 (true) for all the rows that represents our **customers**.
- Parameter **P6**: A column that contains a 1/0 (true/false) flag. The nearest-neighbors given as output of the KNN Action can only belong to the set of rows where this flag is 1 (true). The Parameter **P6** is 1 (true) for all the rows that represents our **shops**.

The parameters **P5** and **P6** bassicaly says: "*For all our customers, compute the distances to the shops*". The (advanced) parameters **P7** to **P12** are identical to the previous use-case.

An example of output

Let's assume that we have a customer database with 1000 customers.

We'll have as output from the KNN Action a table such as:

ID	This_Idx	NN_Idx_1	NN_Dist_1	NN_Idx_2	NN_Dist_2
Customer_996	995	1026	1.23572	1029	20.4495
Customer_997	996	1002	0.815545	1003	2.33418
Customer_998	997	1020	0.383801	1021	4.14966
Customer_999	998	1020	3.10892	1021	5.44512
Customer_1000	999	1028	15.0644	1029	27.8404
Shops_1	1000				
Shops_2	1001				
Shops_3	1002				
Shops_4	1003				
Shops_5	1004				

Some rows were omitted for clarity

The nearest neighbors are **NOT** computed for the shops (because of the "from=0" for the shops)

These 2 "output" columns (that contains the K-Nearest-Neighbors) only contains key values above 1000 because all the KNN's must be "Shops" (because of the "to=1" only for the shops) and all the keys from the Shops are above 1000.

This cell means that the closest shop to the customer "Customer_997" is the shop "Shops_3" (because the value of the "This_Idx" for the "Shops_3" is 1002). The "Shops_3" is at 0.815545 Km from "Customer_997".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.2. Geocode using Nominatim

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: imGeocode | Add parameter | Remove

Description	Value	Parameters:
Type of request	Free-form	P1
[Free-form] Query search	C1	P2
[Structured] Street		P3
[Structured] City	C1	P4
[Structured] County		P5
[Structured] State		P6
[Structured] Country		P7
[Structured] Postal Code		P8
URL	https://nominatim.o...	P9
Batch request size	1	P10
Only return one answer (the most relevant)	☑	P11
Debug Display?	Basic Debugging Info...	P12
Optional parameters for cURL		P13
Number of "Connection Errors" before Ab...	3	P14
Sleep X seconds between each call. X=?	1	P15

Short description:
Geocode some postal addresses

Long Description:
This action takes as input some postal addresses (either stored in “free-form” or stored in “structured form”) and returns:

- The latitude & longitude of the postal addresses.
- Other fields: boundingBox1, boundingBox2, boundingBox3, boundingBox4, display_name, type, importance, address_house_number, address_road, address_city_district, address_town, address_county, address_state, address_region, address_postcode, address_country, address_country_code, Polygon.

Anatella offers many different engines to geocode addresses. You can use:

- The “Google Map” or the “Geocode Farm” engine: See the section 5.10.6 for more details. This is the most expensive solution.
- The “MS Bing” engine: See the sections 5.23.44 and 5.10.5. for more details. This is a paid solution if you need to geocode more than 125000 addresses per year. This solution tends to become rather expensive when you need to geocode large volumes of data.
- The current “nominatimGeocode” Action. This solution is 100% free. The “nominatimGeocode” Action allows to geocode an unlimited number of postal addresses without paying anything.

Behind the scenes, this action sends “geocoding requests” to a geocoding server based on the “nominatim” technology. These “geocoding requests” are grouped into batch of **P10** addresses (the default value for the parameter **P10** is 50, meaning that Anatella is sending to the nominatim server batches of 50 addresses to geocode). A larger value for the parameter **P10** means a (slightly) reduced computing time.

The “nominatim” geocoding server can be:

- The free server from the open-streemap association.
  To access this server, use as parameter **P9** (the url): https://nominatim.openstreetmap.org
  This server has some limitations:
  - The parameter **P10** (batch request size) must be 1
  - The parameter **P15** (seconds to sleep between each call) must be, at least, 1.
- A pre-compiled nominatim server directly available inside a VM that is just “*ready to be used*”. You can download this “*ready to be used*” nominatim server for free from here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

http://bigfiles.timi.eu/nominatim/

Warning: This is a big download (around 107GB to download).

To use this VM, you need to decompress it (e.g. using winrar) and execute it inside the VMWare Player software that you can download for free from the VMWare website. Inside this VM:

- The "root" password is "anatella" (without the quotes)
- The default user login is "anatella" and his password is "anatella" (without the quotes)

The last update date of the maps included inside this "*ready to be used*" nominatim server is 2021/5. This server accepts batch requests (parameter **P10**) of any size. To access this server, you'll typically use as parameter **P9** (the "url"): http://192.168.23.129:8081/nominatim You can&should also set the parameter **P15** (the "*seconds to sleep between each call*") to zero.

- Your own nominatim server: Please refer to the next section 5.10.2.1. to know the procedure required to setup your own nominatim server. The main advantage of building your own nominatim server is to get the most up-to-date maps from "open street map".

5.10.2.1. Building your own Nominatim Server.

We copied the majority of the steps from the Nominatim install guide and we modified slightly where we got blocked (so if this document gets outdated, please visit https://nominatim.org/release-docs/latest/admin/Installation/) for the updated steps.

Here we will follow the steps for a Ubuntu 20 installation.
The recommended mininum HW requirements are:

- HardDisk : +/- 1TB (NVME disks is recommended)
- RAM : 40GB for EU (64GB for global coverage)
- Fast CPU
- Ubuntu 18 or 20

The startpoint is to have an "empty" ubuntu box (inside a VM or not).

5.10.2.1.1. Installing the Required Software

These instructions expect that you have a freshly installed Ubuntu 20.04.
Make sure all packages are are up-to-date by running:

```
sudo apt update -qq
```

Now you can install all packages needed for Nominatim:

```
sudo apt install -y php-cgi
sudo apt install -y build-essential cmake g++ libboost-dev libboost-system-dev \
      libboost-filesystem-dev libexpat1-dev zlib1g-dev \
      libbz2-dev libpq-dev libproj-dev \
      postgresql-server-dev-12 postgresql-12-postgis-3 \
      postgresql-contrib-12 postgresql-12-postgis-3-scripts \
      php php-pgsql php-intl libicu-dev python3-dotenv \
      python3-psycopg2 python3-psutil python3-jinja2 python3-icu
```

5.10.2.1.2. System Configuration

The following steps are meant to configure a fresh Ubuntu installation for use with Nominatim. You may skip some of the steps if you have your OS already configured.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1. Creating Dedicated User Accounts

Nominatim will run as a global service on your machine. It is therefore best to install it under its own separate user account. In the following we assume this user is called nominatim and the installation will be in /srv/nominatim. To create the user and directory run:

```
sudo useradd -d /srv/nominatim -s /bin/bash -m nominatim
```

To be able to copy and paste instructions from this manual, export user name and home directory now like this:

```
export USERNAME=nominatim
export USERHOME=/srv/nominatim
```

Never, ever run the installation as a root user. You have been warned.
Make sure that system servers can read from the home directory:

```
chmod a+x $USERHOME
```

5.10.2.1.3. Setting up & Tune PostgreSQL

You might want to tune your PostgreSQL installation so that the later steps make best use of your hardware. You should tune the following parameters in your `postgresql.conf` file.
(postgresql.conf is for ubuntu & postgres 12 in /etc/postgres/12/main/)

```
shared_buffers = 2GB
maintenance_work_mem = 10GB
autovacuum_work_mem = 2GB
work_mem = 50MB
effective_cache_size = 24GB
synchronous_commit = off
max_wal_size = 1GB
checkpoint_timeout = 10min
checkpoint_completion_target = 0.9
```

The numbers above seem to work fine for 64GB RAM machine. Adjust to your setup (i.e. make sure you have enough RAM!). A higher number for `max_wal_size` means that PostgreSQL needs to run checkpoints less often but it does require additional space on your disk.

Restart the postgresql service after updating the above config file:

```
sudo systemctl restart postgresql
```

Finally, we need to add two postgres users: One for the user that does the import and another for the webserver which should access the database for reading only:

```
sudo -u postgres createuser -s $USERNAME
sudo -u postgres createuser www-data
```

5.10.2.1.4. Installing the Nominatim code

1. Building and Configuration

Get the source code for the release and unpack it:

```
cd $USERHOME
wget https://nominatim.org/release/Nominatim-3.7.1.tar.bz2
tar xf Nominatim-3.7.1.tar.bz2
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The code must be built in a separate directory. Create this directory, then configure Nominatim in there:

```
mkdir $USERHOME/build
cd $USERHOME/build
cmake $USERHOME/Nominatim-3.7.1
make
sudo make install
```

2. Setting up a webserver (Apache)

The webserver should serve the php scripts from the website directory of your project directory. Therefore, set up a project directory and populate the website directory:

```
mkdir $USERHOME/nominatim-project
cd $USERHOME/nominatim-project
nominatim refresh --website
```

Apache has a PHP module that is reauired for Nominatim. To install this module run:

```
sudo apt install -y apache2 libapache2-mod-php
```

You need to create an alias to the website directory in your apache configuration. Add a separate nominatim configuration to your webserver:

```
sudo tee /etc/apache2/conf-available/nominatim.conf << EOFAPACHECONF
<Directory "$USERHOME/nominatim-project/website">
  Options FollowSymLinks MultiViews
  AddType text/html   .php
  DirectoryIndex search.php
  Require all granted
</Directory>

Alias /nominatim $USERHOME/nominatim-project/website
EOFAPACHECONF
```

Then restart apache:

```
sudo a2enconf nominatim
sudo systemctl restart apache2
```

The Nominatim API is now available at `http://localhost/nominatim/`.

5.10.2.1.5. Importation 1: Creating the Project directory

Before you start the import, you should create a project directory for your new database installation. This directory receives all data that is related to a single Nominatim setup: configuration, extra data, etc. Create a project directory apart from the Nominatim software and change into the directory:

```
mkdir ~/nominatim-project
cd ~/nominatim-project
```

In the following, we refer to the project directory as `$PROJECT_DIR`. To be able to copy&paste instructions, you can export the appropriate variable:

```
export PROJECT_DIR=~/nominatim-project
```

The Nominatim tool assumes per default that the current working directory is the project directory but you may explicitly state a different directory using the `--project-dir` parameter. The following instructions assume that you run all commands from the project directory.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.2.1.6. Importation 2: Configuration setup in .env

The Nominatim server can be customized via an .env configuration file in the project directory. This is a file in dotenv format which looks the same as variable settings in a standard shell environment. You can also set the same configuration via environment variables. All settings have a NOMINATIM_ prefix to avoid conflicts with other environment variables.

```
cd  $PROJECT_DIR
touch .env
```

5.10.2.1.7. Importation 3: Enable Flatnode storage

If you plan to import a large dataset (e.g. Europe, North America, planet), you should also enable flatnode storage of node locations. With this setting enabled, node coordinates are stored in a simple file instead of the database. This will save you import time and disk storage. Add to your .env :

```
NOMINATIM_FLATNODE_FILE="/path/to/flatnode.file"
```

Replace the second part with a suitable path on your system and make sure the directory exists. There should be at least 75GB of free space.

5.10.2.1.8. Importation 4: Additional optional download: Wikipedia/Wikidata rankings

Wikipedia can be used as an optional auxiliary data source to help indicate the importance of OSM features. Nominatim will work without this information but it will improve the quality of the results if this is installed. This data is available as a binary download. Put it into your project directory:

```
cd $PROJECT_DIR
wget https://www.nominatim.org/data/wikimedia-importance.sql.gz
```

The file is about 400MB and adds around 4GB to the Nominatim database. We didn't perform this step to create the VM that contains the "*ready to be used*" nominatim server.

5.10.2.1.9. Importation 5: Additional optional download: Great Britain& USA postcodes

Nominatim can use postcodes from an external source to improve searches that involve a GB or US postcode. This data can be optionally downloaded into the project directory:

```
cd $PROJECT_DIR
wget https://www.nominatim.org/data/gb_postcode_data.sql.gz
wget https://www.nominatim.org/data/us_postcode_data.sql.gz
```

5.10.2.1.10. Importation 6: Get and load the map data

If you only need geocoding for a smaller region, then precomputed OSM extracts are a good way to reduce the database size and import time. Geofabrik offers extracts for most countries. They even have daily updates which can be used with the update process described in this webpage. There are also other providers for extracts. Please be aware that some extracts are not cut exactly along the country boundaries. As a result, some parts of the boundary may be missing which means that Nominatim cannot compute the areas for some administrative areas.

You can now download the data extract for the region of your interest:

- You can find the continents here : https://download.geofabrik.de/
- Or for EU countries : https://download.geofabrik.de/europe.html

```
Download the raw osm.pbf ( or download the zipped version and unzip )
THIS WILL BE THE <<DATAFILE>> for the next step
```

Issue the following command from the **project** directory to start the import (The official guide says to start from the **build** directory but this did not work):

```
nominatim import --osm-file <data file> 2>&1 | tee setup.log
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Warning: The above step takes more than 48 hours!

If you get a database error your user can't access the data → add the users :

```
Sudo su postgres
Psql
Create user <<your user name>>:
 grant all privileges on database nominatim to <<your user name>>;
 \q
Exit
```

If the import fails and you want to retry, it is easier to remove the Nominatim database:

```
Sudo su postgres
Psql
Drop database nominatim
```

5.10.2.1.11. Importation 7: Notes on full planet imports

Even on a perfectly configured machine the import of a full planet takes around 2 days. Once you see messages with `Rank .. ETA` appear, the indexing process has started. This part takes the most time. There are 30 ranks to process. Rank 26 and 30 are the most complex. They take each about a third of the total import time. If you have not reached rank 26 after two days of import, it is worth revisiting your system configuration as it may not be optimal for the import.

5.10.2.1.12. Importation 8: Testing the installation

Run this script to verify all required tables and indices got created successfully:

```
nominatim admin --check-database
```

Now you can try out your installation by running:

```
nominatim serve
```

This runs a small test server normally used for development. You can use it to verify that your installation is working. Go to `http://localhost:8088/status.php` and you should see the message `OK`. You can also run a search query, e.g. `http://localhost:8088/search.php?q=Berlin`.

If you are using the "*read to use*" VM, you can test it by running inside a command-prompt from the VM-host (e.g. from a Windows command-prompt):

```
curl -kg http://192.168.23.129:8081/nominatim/status.php?format=json -X GET
```

5.10.2.1.13. Enable bulk api requests.

Edit the file "website/search.php" inside the **Project** directory (i.e. edit the file "/srv/nominatim/nominatim-project/website/search.php") and replace the following string:

```
@define('CONST_search_batchMode', 'false');
```

…with:

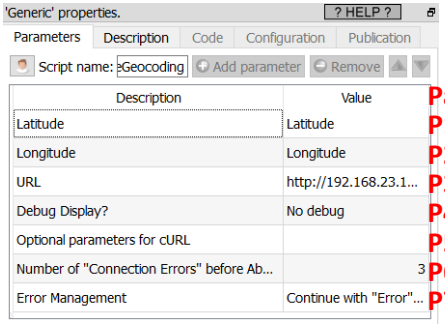
```
@define('CONST_search_batchMode', 'true');
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.3. Reverse Geocode using Nominatim

Icon:

Property window:



Short description:
Reverse Geocode some GPS coordinates

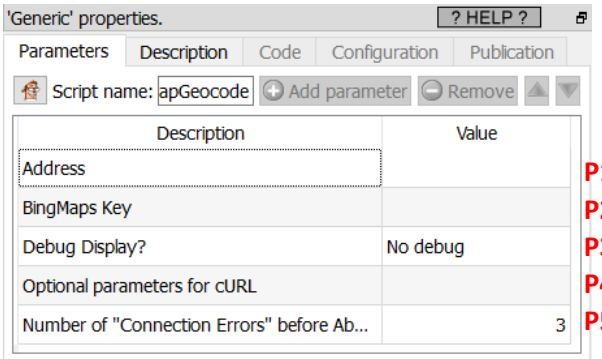
Long Description:
This action takes as input some GPS coordinates and returns the corresponding postal addresses.

Behind the scenes, this action sends “reverse-geocoding requests” to a geocoding server based on the “nominatim” technology. Please refer to the section 5.10.2.1. to know how to install a “nominatim” server.

5.10.4. Geocode Addresses using Bing

Icon:

Property window:



Short description:
Geocode addresses using Bing

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P4** for web-access through a PROXY server.

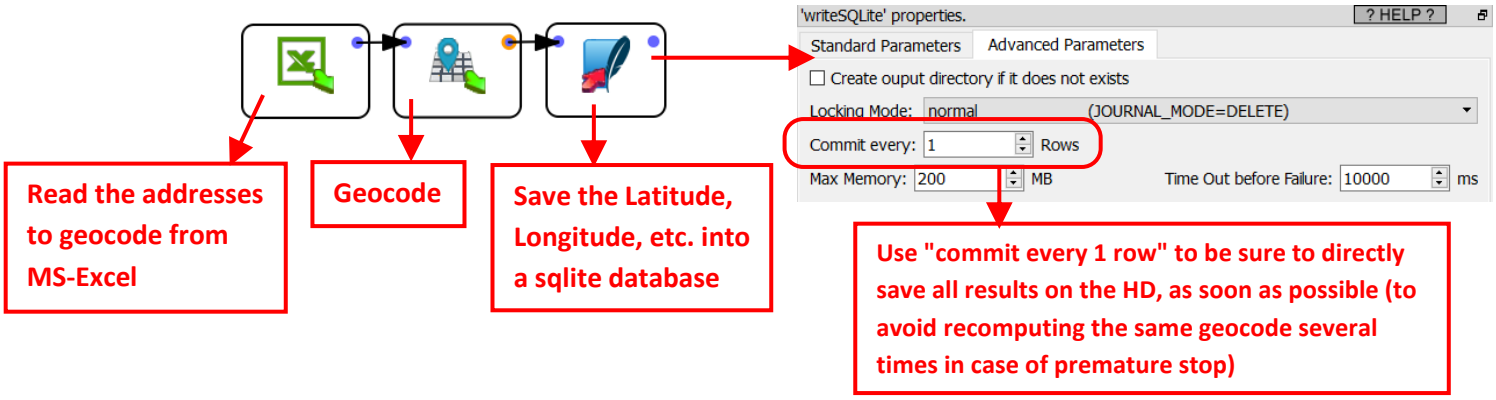
To use this Action, you’ll first need to get a “BingMaps Key” from the Bing Maps website (parameter **P2**). Please see the next section 5.10.4.1 for more details on how to get this key.

Once you have completed the “setup process” described in the section 5.10.4.1, you can use this action. This is pretty straightforward: Just give the column (in parameter **P1**) that contains some postal addresses “as-if” these addresses where written directly inside BING. Then, Anatella will give the following 16 columns in the output table:

Latitude, Longitude, Confidence, statusCode, statusDescription, authenticationResultCode, copyright, EstimatedTotal, AddressLine, AdminDistrict, AdminDistrict2, CountryRegion, FormattedAddress, Locality, PostalCode, MatchCode.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Each call to the BING engine is slow (because all internet-based REST api calls are always slow). So, you want to avoid to geocode 2 times the same address (to avoid losing time). To be sure to save all the geocoding results on the hard-drive as soon as they have been extracted from BING, you should use the writeSQLITE box with the option “Commit every 1 row” enabled:

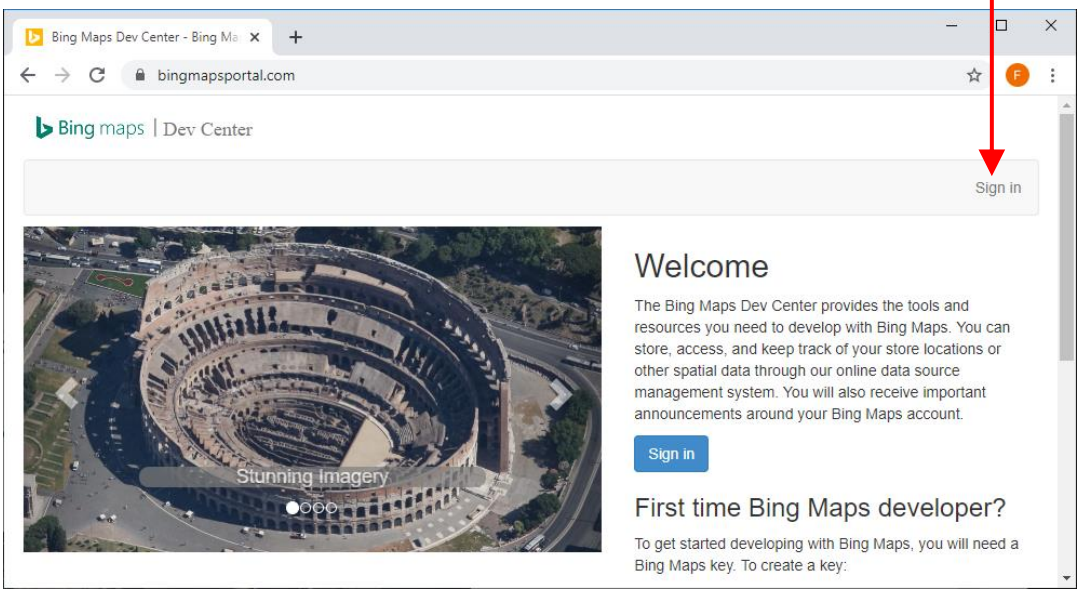


In the above graph, the addresses to geocode are directly extracted from an Excel file. Ideally, you should have a slightly more complex Anatella graph that removes all the already-geocoded addresses (that are stored inside the SQLite file) from the input of the bingMapGeocode Action (to avoid to geocode 2 times the same address).

5.10.4.1. How to get you BING Map Key.

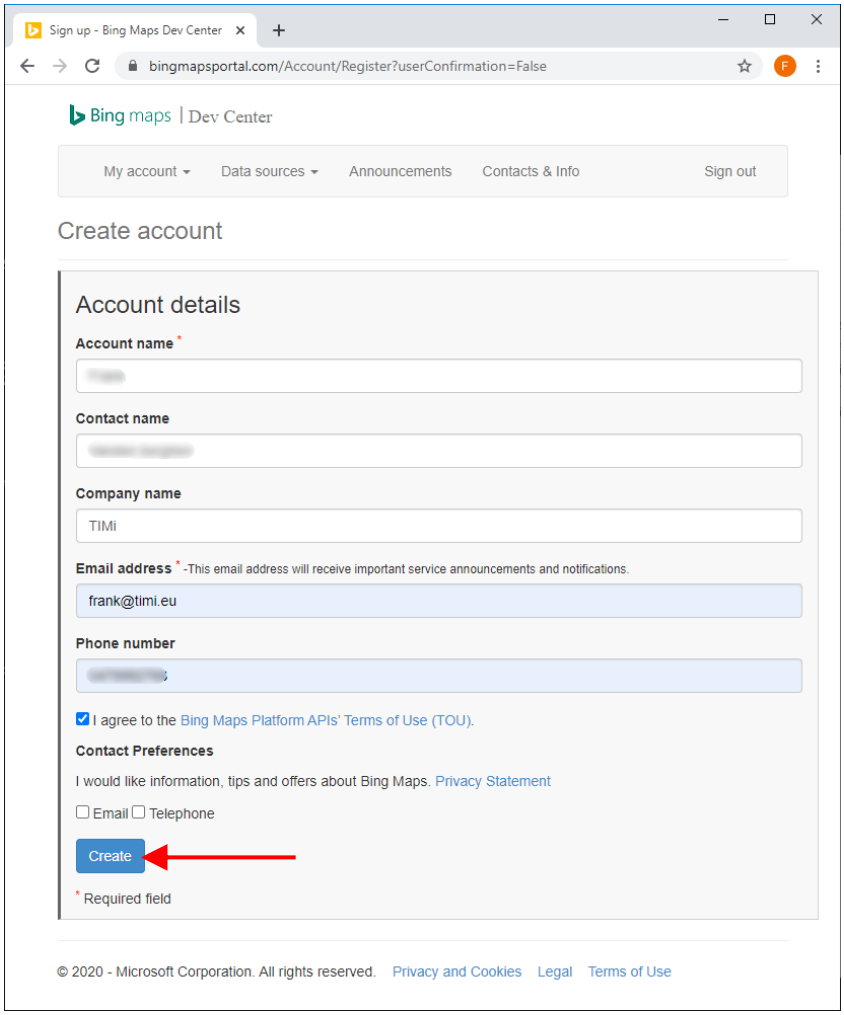
The procedure to get you Bing Map key is:

1. Open the URL https://www.bingmapsportal.com and click the “sign-in” button:

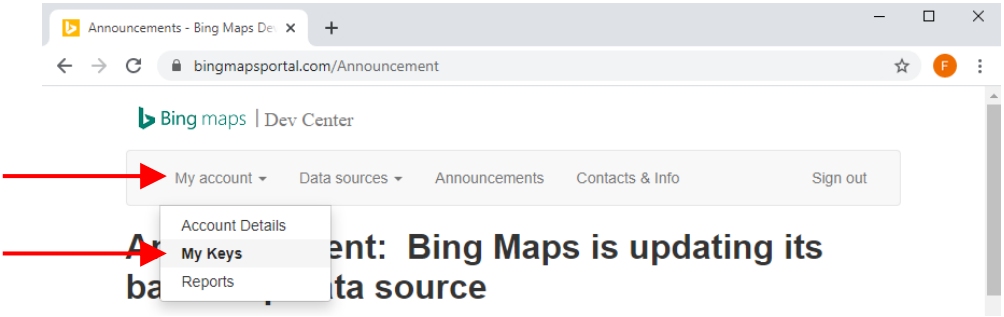


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Fill-in the form to create an account and click the “Create” button:

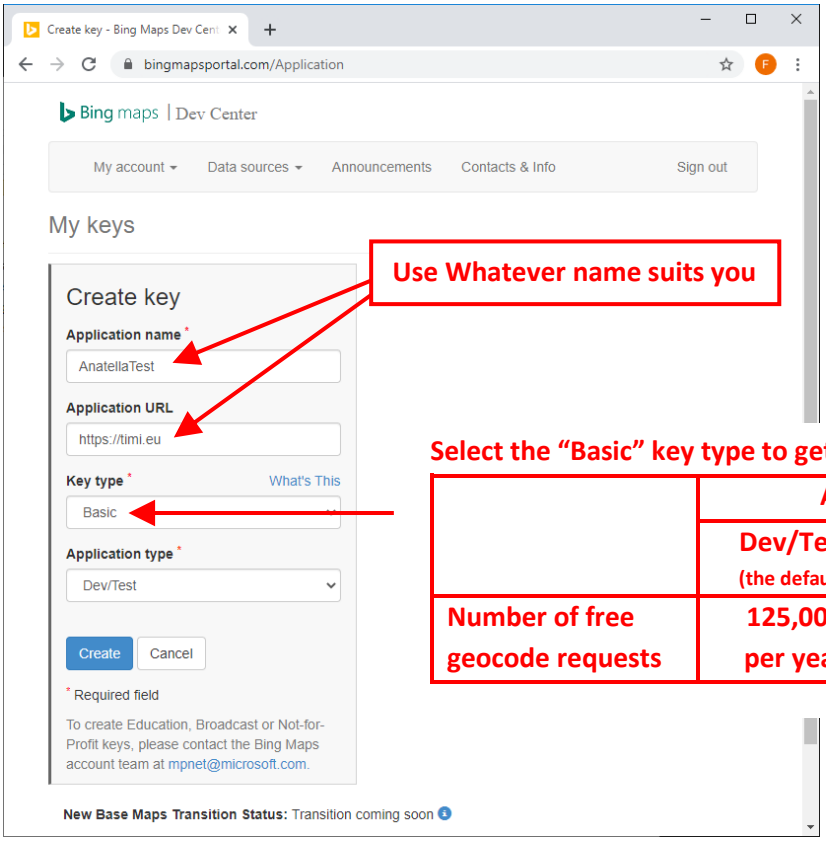


3. Inside the Drop-Down menu “My account”, click on “My Keys”:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. Fill-In the form to get your key:

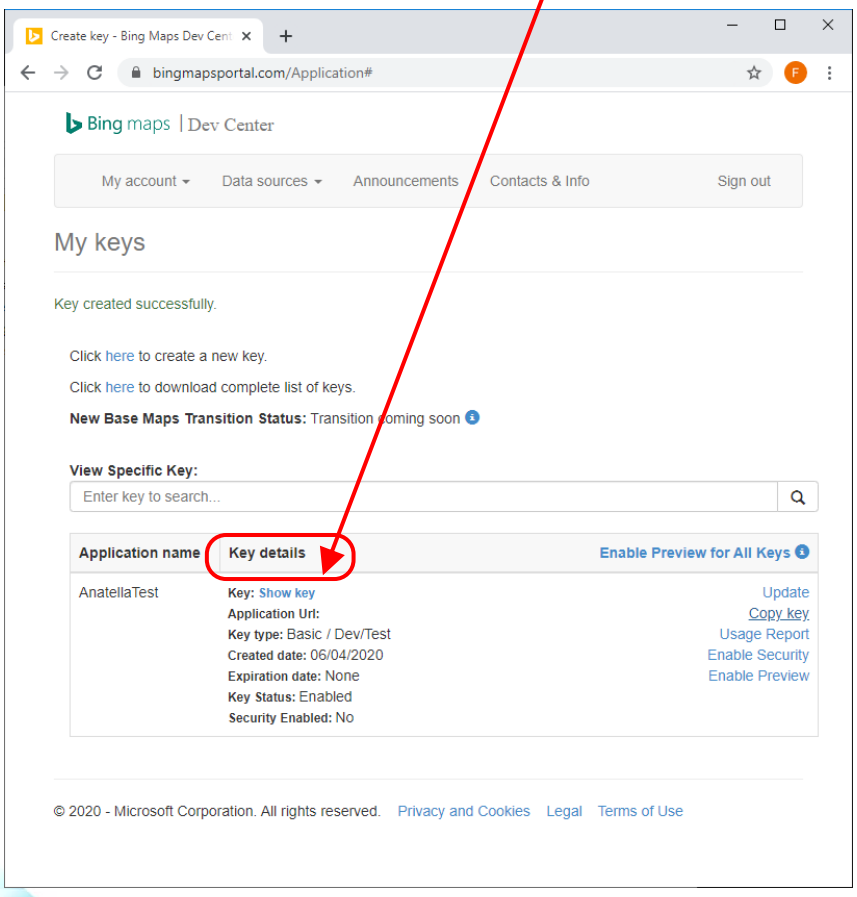


Select the "Basic" key type to get:

	Application type	
	Dev/Test (the default)	Windows App
Number of free geocode requests	125,000 per year	50,000 per day, each day

You'll find more information about the different licensing options for Bing Maps here:
https://www.microsoft.com/en-us/maps/licensing/
Basically, if you need to make more geocoding requests than the limits detailed here , then you need to purchase an "Enterprise Key" (i.e. the "Basic" key is not enough).

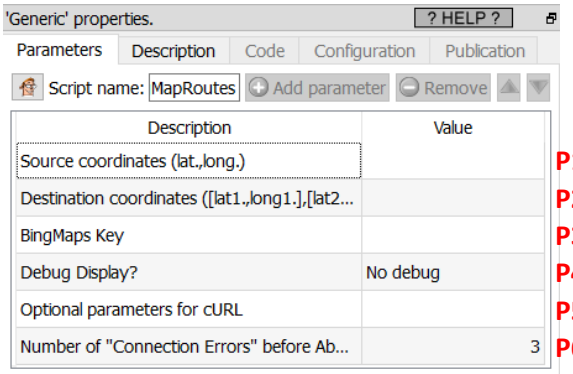
5. You get your key: Click on the "show key" link to reveal it:



5.10.5. Compute the Traveling-time using Bing

Icon:

Property window:



P1
P2
P3
P4
P5
P6

Short description:
Compute travelling time using Bing

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

To use this Action, you'll first need to get a "BingMaps Key" from the Bing Maps website (parameter **P3**). Please see the previous section 5.10.4.1 for more details on how to get this key.

Once you have completed the "setup process" described in the section 5.10.4.1, you can use this action. This action computes the travelling time between one *source* location (Parameter **P1**) and, possibility, several *destination* locations (Parameter **P2**). So, for each row of the input table, we have:

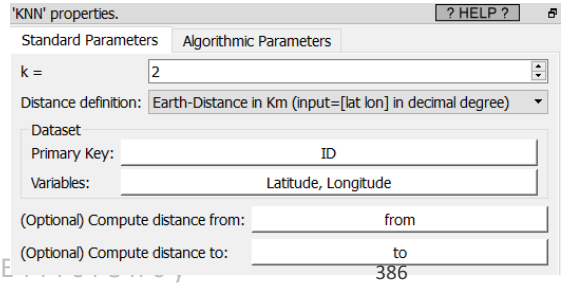
- one source location and several destination locations (maximum number of destinations is around 200)
- one API-call to the BING computation engine

For each (*source, destination*) pair you get as output a table with these 4 columns:
originIndex, destinationIndex, travelDistance, travelDuration

The parameter **P1** is the coordinates of the *source*: it's 2 columns: The first column is the latitude and the second column is the longitude (in decimal degree) of the source location.

The parameter **P2** contains the coordinates of all the D *destinations*: It's 2xD columns: The two first columns are the latitude & longitude of the 1st destination (in decimal degree, in this order). The 2 following columns are the latitude & longitude of the 2nd destination. The 2 following columns are the latitude & longitude of the 3rd destination. Etc.

Let's assume that you want to compute the minimum "Travel Duration" between the home address of your customers and all your shops. Let's assume that you have C customers and S shops. The BING invoice is directly proportional to S x C, so it might be a good idea to reduce to the minimum this product. One good idea to reduce this product to a minimum is to use the KNN action (from section 5.10.1.). Basically, the idea is to use the KNN action to find the 2 closest shops to each of your customer "*as the crow flies*" and then, only sent to BING these 2 shops. In this way, you only have 2 x C routes to compute (instead of S x C routes to compute), the computing-time is reduced, and you get a smaller invoice. Uses these parameters for the KNN action:

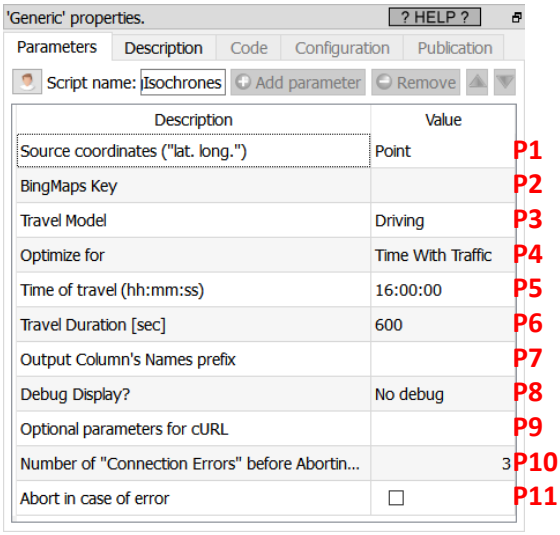


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.6. Compute Isochrones using Bing

Icon:

Property window:



Short description:
Compute Isochrones using Bing

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

To use this Action, you'll first need to get a "BingMaps Key" from the Bing Maps website (parameter **P2**). Please see the previous section 5.10.4.1 for more details on how to get this key.

Once you have completed the "setup process" described in the section 5.10.4.1, you can use this action. This action computes the isochrone polygon for all the given starting points (Parameter **P1**). The polygon in output is in WKT format: if required, use the ConvertGIS Action (see section 5.10.16.) to convert it to another format.

This action returns time-specific, isoline polygons for the distance that is reachable from a given location and supports multiple modes of transportation (i.e., driving, walking, and public transit). Use this solution to plan the area that can be reached from a designated starting point within a set time period. The isoline polygon area, good for visualization, can be used to filter for spatial queries, which opens up a wide variety of applications for spatial filtering. Here are some use cases of where you might use isochrones:

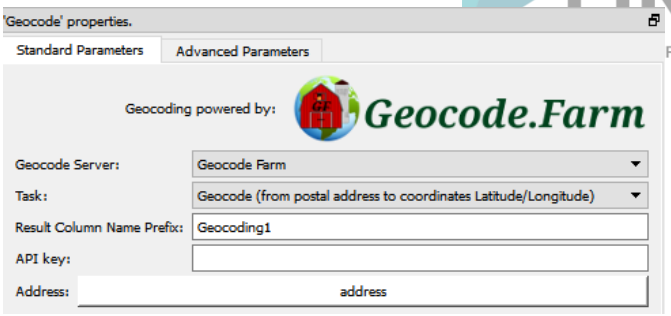
- **Store Locators** – Show me all users that are within 10 minutes of a shop.
- **Stolen Vehicle Recovery** – Where could a vehicle have travelled to since it was stolen.
- **Real Estate** – Limit search results such that only those that are within the users preferred commute preferences to work. For example, show me all homes that are within a 30-minute drive of work.
- **Job Search Sites** – Show all jobs that are within 45 minutes of my home when taking public transit.
- **Geofence Generation** – Generate a polygon that be used as a geofence that alerts users when they are within a certain travel time or distance of a location.
- **Field management** – Show me all workers who are within a 15-minute drive of a job.
- **Recreation** – Where could I walk to within 30 minutes of where I am.
- **Emergency Services Planning** – Where could an emergency vehicle travel too within 5, 10, and 15 minutes.

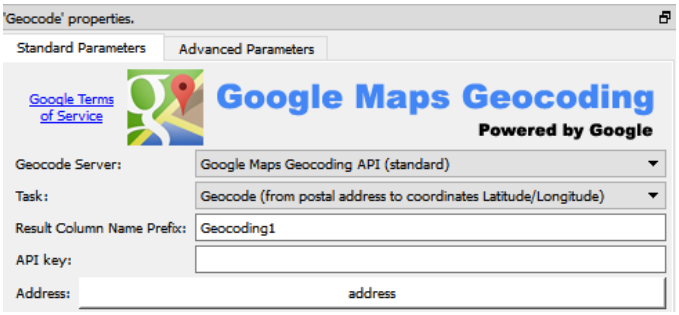
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

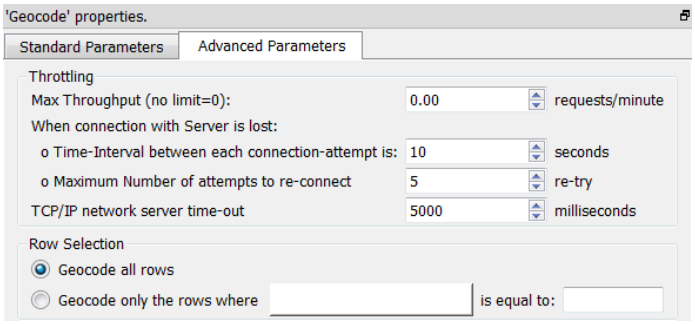
5.10.7. Geocoding & Reverse Geocoding

Icon:

Property window:

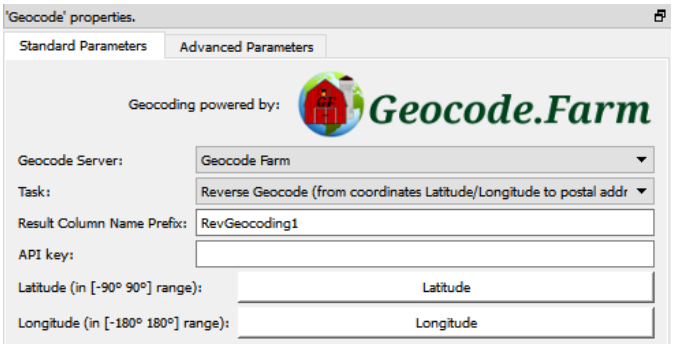






Short description:
Geocode: Find the (Longitude, Latitude) of different Postal Addresses.

Inverse Geocode: Find the postal Addresses at different given (Longitude, Latitude).



Long Description:

The geocode Action that is detailed in this section uses two external service providers to compute the geocoding and the reverse-geocoding: Geocode.farm and Google. One last geocoding provider (i.e. BING MAPS) is accessible throught the bingMapGeocode Action detailed in the section 5.22.39. On June 2020, BING MAPS is the fastest and cheapest geocoding provider available. So, we strongly encourage you to use the bingMapGeocode Action instead of the current Action detailed in this section.

During geocoding, the geocode Action takes as input a table with a column containing a postal address (by default this column is named “address”).

The address formatting (in input) is very simple: For full addresses, just send the request in the same format you would use for a piece of mail (eg. 530 W Main St Anoka MN 55303 US). Just be sure to include the country at the end of the address to ensure accurate results. For zip codes, send the zip code along with the country in your request (eg. 55303 US). International addresses and characters are permitted by the API (Full unicode is supported).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The geocode Action ouputs a table containing all the fields from the input table plus these additional fields (the "<column_prefix>" is the parameter given here:):

- **<column_prefix>_Latitude**
- **<column_prefix>_Longitude**
- **<column_prefix>_Accuracy**
  The "accuracy" field can have the following values:

VERY ACCURATE	This is the highest level of accuracy and usually indicates a spot-on match.
GOOD ACCURACY	This is the second highest level of accuracy and usually indicates a range match, within a few hundred feet most.
ACCURATE	This is the third level of accuracy and usually indicates a geographical area match, such as the metro area, town, or city.
UNKNOWN ACCURACY	The accuracy of this result is unable to be determined and an exact match may or may not have been obtained.

- **<column_prefix>_Status:**
  This field can have the following values:

SUCCESS	There are no account issues and query was successfully executed. Results were found and are displayed.
FAILED, ACCESS_DENIED	There are account issues and query was not successfully executed. Please see "access" field response for details.
FAILED, NO_RESULTS	There are no account issues and query was successfully executed. There were no results found for query, so no results are displayed. Check your query to ensure it is formatted properly or try reformatting the query.
HTTP ERROR	There is a network problem contacting the www.geocodefarm.com server.
JSON ERROR	The JSON response that we received from the www.geocodefarm.com server is invalid.

- **<column_prefix>_AddressReturned**
  The real address for which the (latitude, longitude) is returned.

- **<column_prefix>_Access**
  This field can have the following values:

KEY_VALID, ACCESS_GRANTED	There are no account issues and query can proceed.
API_KEY_INVALID	You entered your API Key incorrectly. Please double-check that your API Key entered matches that shown in the Dashboard Home screen.
ACCOUNT_NOT_ACTIVE	You need to check your email for the activation link for your account. Clicking on this link will activate your account and this message should then stop appearing. Ensure you check

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

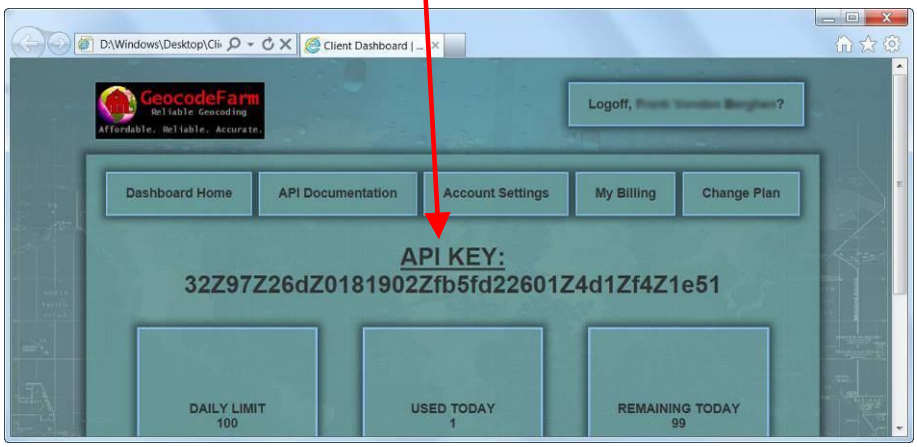
	your "Spam" or "Junk" folders to ensure that your Spam Filter did not tag our email as Spam.
BILL_PAST_DUE	You are on a Paid Plan and your bill has not been paid. Please Login to your geocodeFarm.com Dashboard and pay the monthly fee to restore your API access.
OVER_QUERY_LIMIT	You have hit your daily limit for today. Upgrade your Plan if this occurs frequently or very often. This limit will reset at Midnight (12pm) PST and your account will be able to process until the limit is his again that day.

- **<column_prefix>_RemainingQueries**
  The quantity of remaining available queries fro the current day. Depending on your plan, you can have a maximum of 100.000 or 50.000 or 25.000 queries per day.

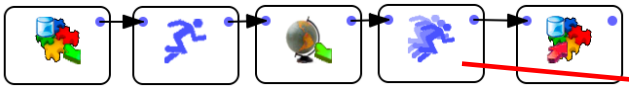
Some queries might fail for different reasons (maybe the internet connection is down?). When a query fails, Anatella makes several attempts to try to still get the answer (the number of attempts is the parameter named "*Maximum Number of attempts to re-connect*"). If Anatella still can't get any valid answer after several attempts, all the remaining rows to process are set to an "Error status". A row that is in "Error status" has:

- The fields "Longitude", "Latitude", "Accuracy", "AddressReturned", "Access", "RemainingQueries" set to NULL.
- The field "Status" set to either FAILED, ACCESS_DENIED; FAILED, NO_RESULTS; HTTP ERROR or JSON ERROR.

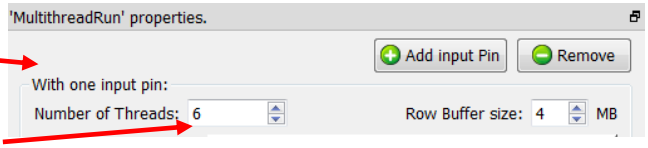
The parameter "API KEY" is the key that defines your geocodefarm account.
You can get it on the www.geocode.farm website:



To increase querying speed, you can place the geocode Action inside a "N-Way" Multithreaded section (see section 5.3.2.2. for more information about "N-Way" Multithreaded sections): For example:



Select a high "Number of threads" to get more speed:

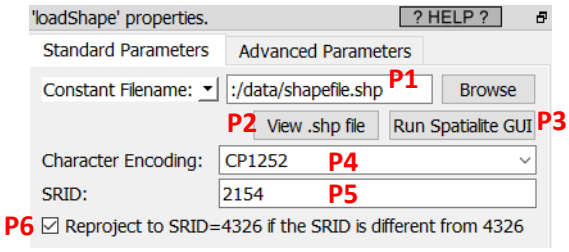


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

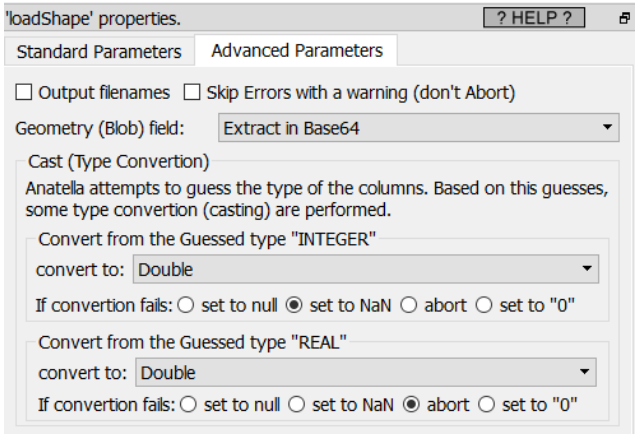
Warning: The Geocoding server might stop responding if you select a “too high” number of Threads.

5.10.8. Load Shape

Icon:



Property window:



Short description:
Load a .shp file.

Long Description:
This action loads a .shp file and all the related files the are typically “linked” to it (i.e. the .shx file, the .dbf file and the optional .prj file). If you want to load the same data, but without the geometry column, it’s more efficient to open the .dbf file using the readXDF action (more details on this action in the section 5.2.30).

What’s inside a .shp file ?

A .shp file contains a single table with a few columns. One of these columns is special: it’s the “geometry” column. Here is an example with a shape file that describes all the zip/postal codes in France (downloaded from here):

DataTable (6 048 rows - 8 columns) (complete)

	PK_UID	id	lib	dep	surf	pop2010	men2010	Geometry
466	466	75019	Paris	75	6.51806	186652	84344.8	AAHmEAAA4BXjM...
467	467	33190	La Réole	33	178.405	14501	6191.58	AAHmEAAAEwQy6...
468	468	33200	Bordeaux	33	7.79738	45412	23229.6	AAHmEAAA1SXUC...
469	469	33210	Langon	33	168.596	23098	10082.4	AAHmEAAAT2Rw3...
470	470	33220	Sainte-...	33	163.35	15262	6867.34	AAHmEAAA9EexV...
471	471	33440	Ambarè...	33	49.7222	16354	6155.99	AAHmEAAA+dkoH...
472	472	76117	Incheville	76	7.9558	1334	585.666	AAHmEAAARubBD...
473	473	33230	Coutras	33	229.008	22643	9578.16	AAHmEAAA4iY7yJ...

In the screenshot above, the column “lib” contains the usual name of the “postal codes” and the column “Geometry” describes the shape/geometries of each postal code. Inside a .shp file, one “Geometry” column contains a unique type of geometry (you cannot mix several types). The available geometry types inside a .shp files are:

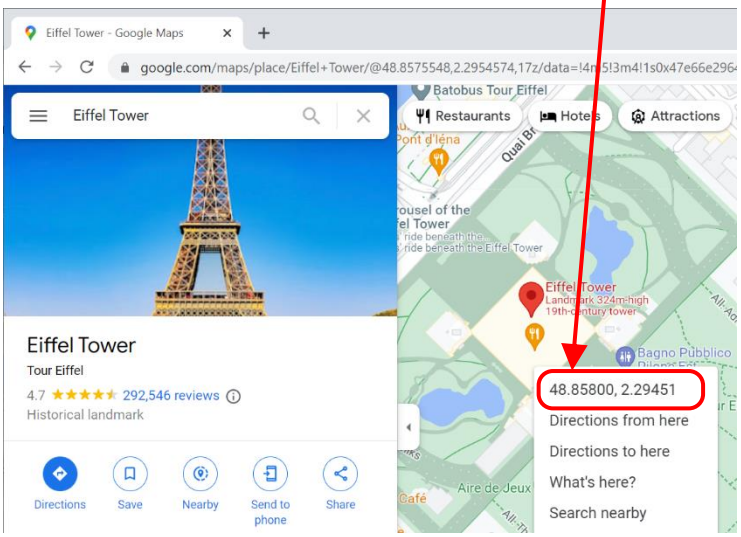
- POINT
- MULTIPOINT
- POLYGON
- MULTIPOLYGON
- LINESTRING
- MULTILINESTRING

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Of course, Anatella handles all these geometry types. Inside the table in the above screenshot you see that the "Geometry" column contains a long string "AAHmEAAA4BXjM..". This is the Base64 representation of the geometry. To see this geometry as a meaningful picture, just use the "saveShapeGIS" action: You'll find more details on how to visualize geoemetries in the section about the "saveShapeGIS" action. To get more details about a specific geometry (such as its type: POINT, POLYGON, MULTIPOLYGON, etc. or its SRID), you'll use the GeometryInfo Action.

Inside Anatella, a geometry can be stored/represented using 5 different formats:

1. **Point Coordinates "Latitude Longitude"** (only for the POINT geometry type)
   This format is self-explanatory. The Latitude and Longitude must be stored in the same column, separated with a space char, in decimal degree. This is exactly what you see when you right-click on a Google Map: For example, when you right-click you see:



2. **Rectangle "Latitude1 Longitude1 Latitude2 Longitude2"**
   This format is self-explanatory. The 4 numbers must be stored in the same column, separated with a space char, in decimal degree. This representation is only valid for RECTANGLE geometries. The first point is is the coordinate of the top-left corner of the RECTANGLE and the second point is the coordinate of the bottom-right corner of the RECTANGLE.

3. **Geometry data in WKT format**
   "WKT" stands for "Well-Known-Text". A WKT string can represent any type of geometry: POINT, POLYGON, MULTIPOLYGON, etc. You can create/draw WKT strings very easily using this interactive mouse-based editor:
   https://timi.eu/wkt.html
   Here is an example of WKT string that represents a single point:
   POINT(3.7403 50.616)

   There is one disturbing fact about the WKT representation: the WKT representation is reversing the order of the coordinates compared to the traditional way of citing geographical coordinates: e.g. in the above WKT example, the first number (3.7403) is the longitude and, traditionally, the first number is the latitude. This is somewhat disturbing.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. **Geometry data in HEX format**
   This format is for compatibility with PostGreGIS. It can represent any type of geometry. You can easily recognize this format because it looks like a long string without many minuscule (small caps) letter (the only allowed chars in the string are: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F).

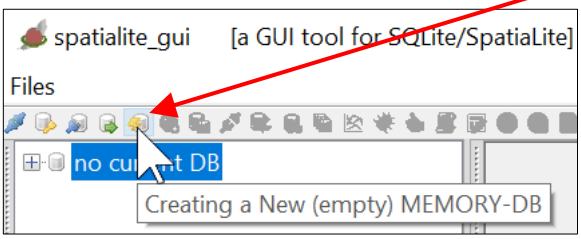
5. **Geometry data in BASE64 format**
   This is the most common format inside Anatella for GIS information. You can easily recognize this format because it looks like a long string with a mixture of minuscule and majuscule letters. Inside Anatella, each different Base64 geometry inside a table can have a different TYPE (POINT, POLYGON, MULTIPOLYGON, etc.) and even a different SRID. …but it's strongly not advised to do so because other file formats are much more limited: e.g. the .shp format only support one unique type of geometry and one unique SRID per file.

The most important parameter of the loadShape action is the parameter **P5** that is named "SRID". The SRID is the name of the mathematical projection/transformation that is used to project the 3D coordinates on the earth (i.e. the latitude, longitude and altitude) on a planar map (that has only 2 dimensions). The most common projection/SRID is the "4326" SRID that corresponds to the standard "decimal degrees" used inside most modern on-line maps. You'll find more details on the SRID here and inside the section 5.10.10. about the ReProject Action.

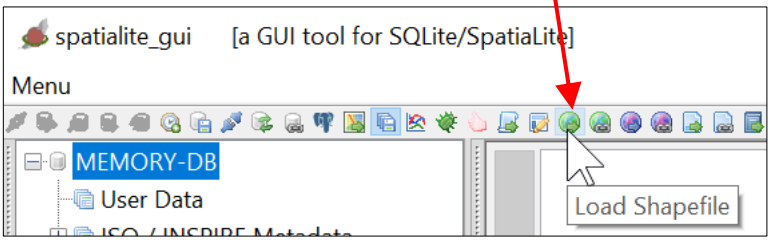
Many Anatella actions are only working with the "4326" SRID, so the parameter **P6** allows to directly re-project all your geometries so that they are correctly rendered in the "4326" SRID. For this re-projection to work properly, you must first give the correct value for the parameter **P5**.

If your shape files are not using the "4326" SRID, you'll need to specify their SRID using the parameter **P5**. You can guess the value of the parameter **P5** (i.e. you can guess the original SRID of your shape files) using this 5-steps procedure:

1. Click the button **P3**.
2. In the toolbar of the "Spatialite Gui" tool, click the button:



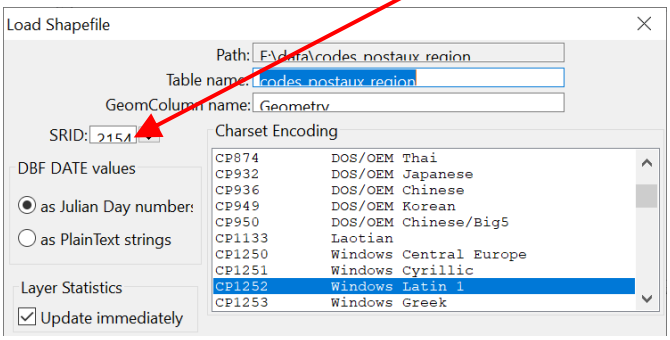
3. In the toolbar of the "Spatialite Gui" tool, click the button:



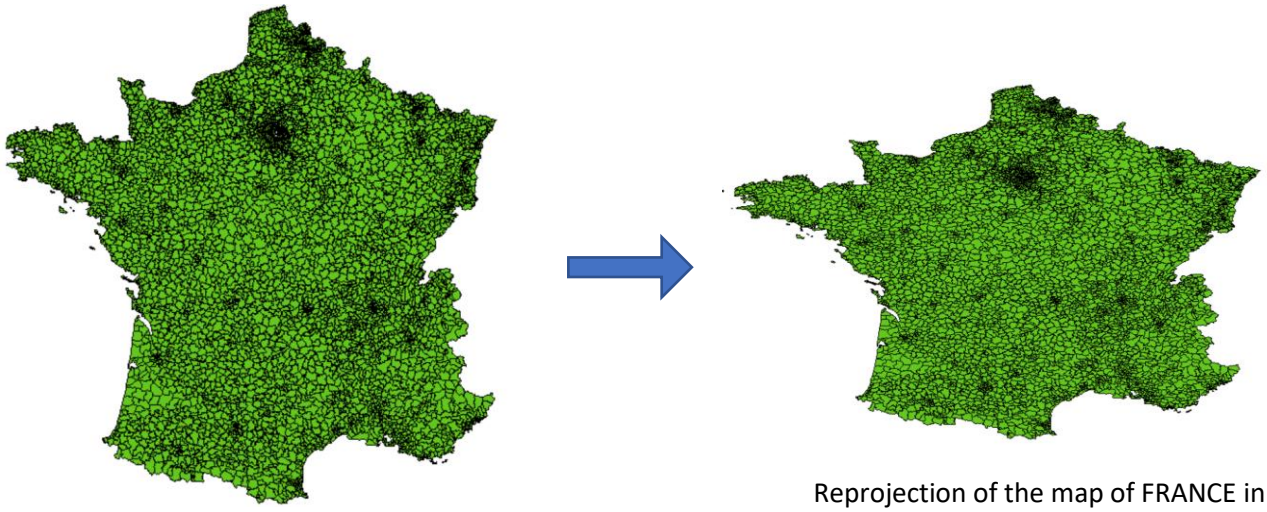
4. Select your .SHP file inside the file explorer.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5. The guessed original SRID of your shape file is here:



Here is an example of re-projection for the map of FRANCE (click **P2** to display your shape file):



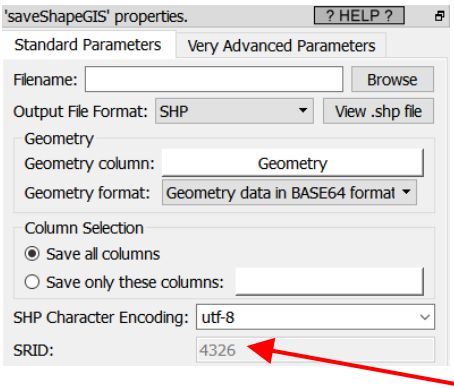
Classical view of the map of FRANCE using the SRID=2154. The X and Y axes are distances in **meters**.

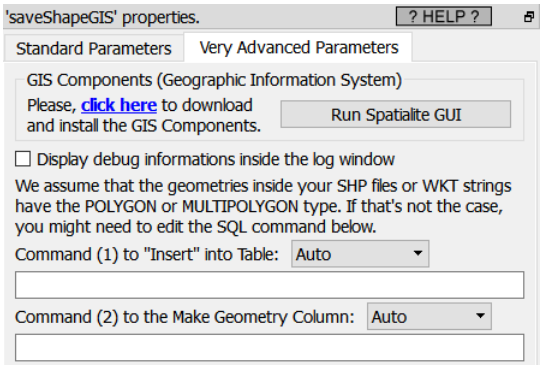
Reprojection of the map of FRANCE in the SRID=4326. The X and Y axes are longitude and latitude in **decimal degree**.

5.10.9. Save and view Shapes

Icon:

Property window:





Short description:
Create a .SHP file
Or a .html file with the geometries

Long Description:
This action is used:

- To create .shp or .html files that contains geographical informations (geometries).
- To visualize your geometries using a nice graphical display.

All the parameters of this action are self-explanatory with the exception of the SRID parameter here:
You'll find more exaplantion about this parameter inside the section 5.10.10.2.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

There are three different approaches to visualize geographical (GIS) informations:

- Look at small html file inside your browser.
- Look at a .shp file inside the small viewer embedded inside Anatella (from Avangardo.com)
- Design a beautiful&complex multi-layer representation using .shp files inside QGIS.

Here are the pros&cons of each approach:

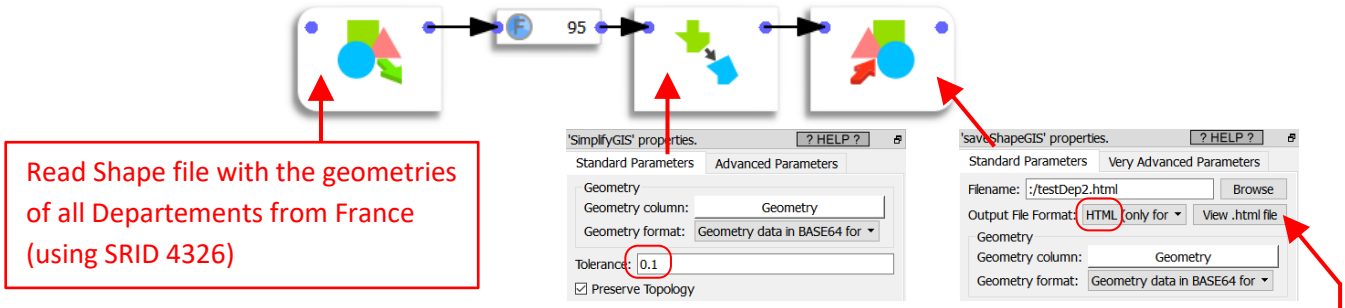
Approach	PRO	CON
HTML	You can easily share the .html file with your collegue since its 100% self-contained. Flexible format: e.g. you can mix POLYGON with MULTIPOLYGON inside the same file and it still works. The map background is "Google Maps": it allows to see your geometries "in context" inside a beautifully colored map (from Google). Quick to generate and view.	You can only look at geometries that are inside the 4326 SRID (i.e. with the coordinate system that is in "decimal degrees"). Limited in size: Your browser (i.e. Chrome, Firefox) is displaying the geometries and it's limited to relatively "simple" geometries with less than 1000 points (otherwise it becomes slow and irresponsive). One way to still be able to use the html approach to look at your geometries is to use the simplifyGIS action to simplify your geometries. You cannot choose the map background: it's always "google maps".
SHP +quick viewer	To share the results, you only need to send 4 files: the SHP file, the SHX file, the DBF file and the PRJ file (you should put these 4 files inside a single zip and send the zip). You can view geometries that have any SRID (i.e. with any coordinates system). Unlimited in size: Look at very complex geometries with thousands of points instantaneously. Quick to generate and view.	The file format is not flexible: i.e. a single geometry TYPE is allowed and a single SRID is allowed (for each different .shp file). Does not support the POINT and MULTIPOINT geometry type. There is no map background to see your geometries "in context".
SHP + QGIS	You can create a really beautiful & complex vizualisation with several transparent layers (usually, one layer = one .shp file). Unlimited in size: Look at very complex geometries with thousands of points. You can choose any map background: By defaut, it's "open street map" but you can also use "google maps" (if you have a key) or use any other Tile provider.	It's more complex to share the results: a screenshot of QGIS is usually the easiest solution. Slightly More Complex to use.

Let's give an example of vizualisation for each approach.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

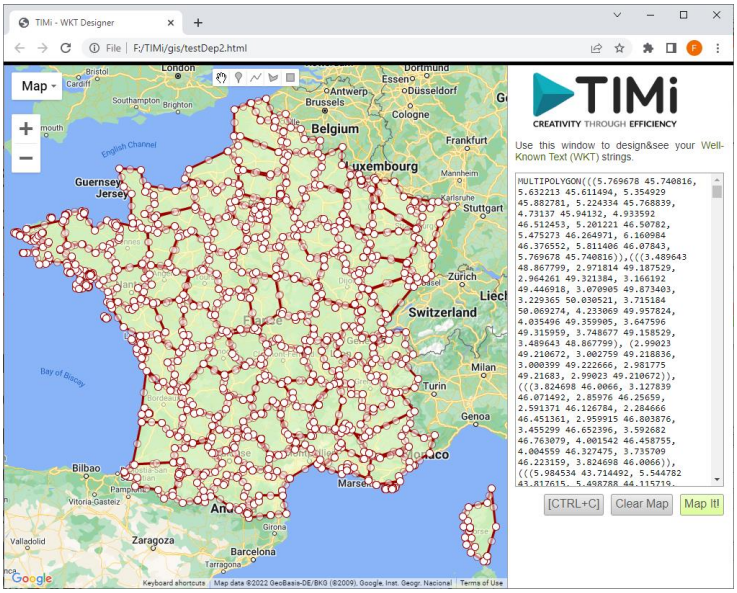
5.10.9.1. HTML visualization

Let's look at the different departments in France:



In this example, we process a shape file with 95 rows (i.e. there are 95 departments in France) and with 38 268 points (i.e. the geometry of the boundaries of each department is complex). Since this is too much points for your browser to display, we use the SimplifyGIS action (with a tolerance of 0.1 decimal degree) to reduce the total number of points to 1221 inside the geometries. Then, we run the saveShapeGIS action (i.e. right-click the saveShapeGIS box and select the "play" button) and, finally, we click on the "View .html file" button here:

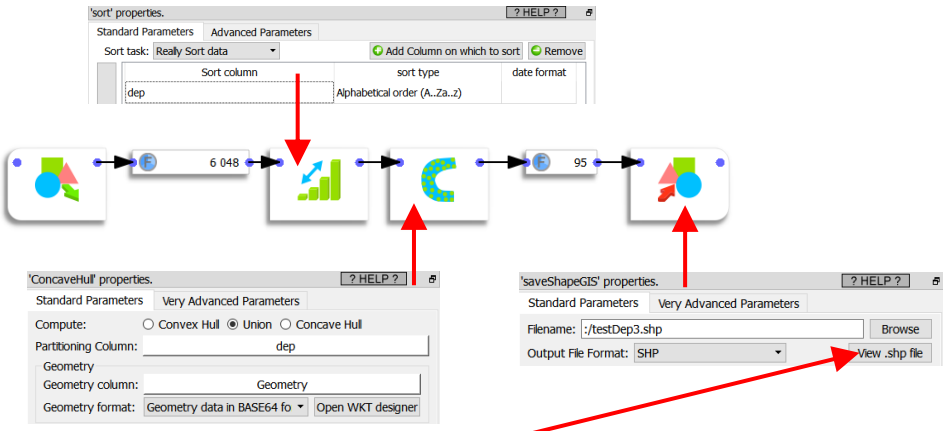
We get:



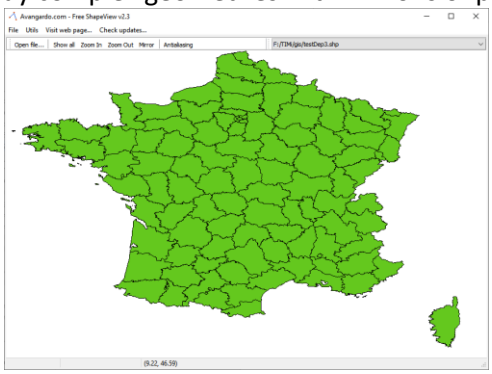
It should be noted that, in the above example, inside the simplifyGIS action, we used a tolerance of 0.1 decimal degree. Although the end result looks visually ok, it's still better to use the simplifyGIS action in a different coordinate system (i.e. not inside the coordinate system with the 4326 SRID). For example, in the example above it would have been better to: (1) reproject the geometries inside the 2154 SRID, (2) simplify the geometries inside this better (cartesian) coordinate system and, (3) finally, reproject the geometries in the 4326 for visualization.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.9.2. Quick SHP visualization



In this example, we process a shape file with 6048 rows (i.e. there are 6048 postal/zip codes in France) and with 228870 points. We use the ConcaveHull action to "group together" all the geometries that belongs to the same department. Finally, we click here: to display the departements inside the quick shape file viewer (that can display complex geometries with millions of points):



5.10.9.3. SHP + QGIS visualization

You can download QGIS from here (1GB download):
https://qgis.org/en/site/forusers/download.html
The installation wizard is simple and straightforward.

Let's now assume that we have two shape files:

- **"SHOPS.shp":** a shape file with the geometry column that contains the coordinates (latitude, longitude) of the shops. One row per shop.
- **"SHOPS_CATCHMENT_AREA.shp":** a shape file with the geometry column that contains the polygons that defines the catchment area for each shop. One row per shop.

  See the section 5.10.13 about the Tesselate action for an example on how to compute&create these catchment areas.

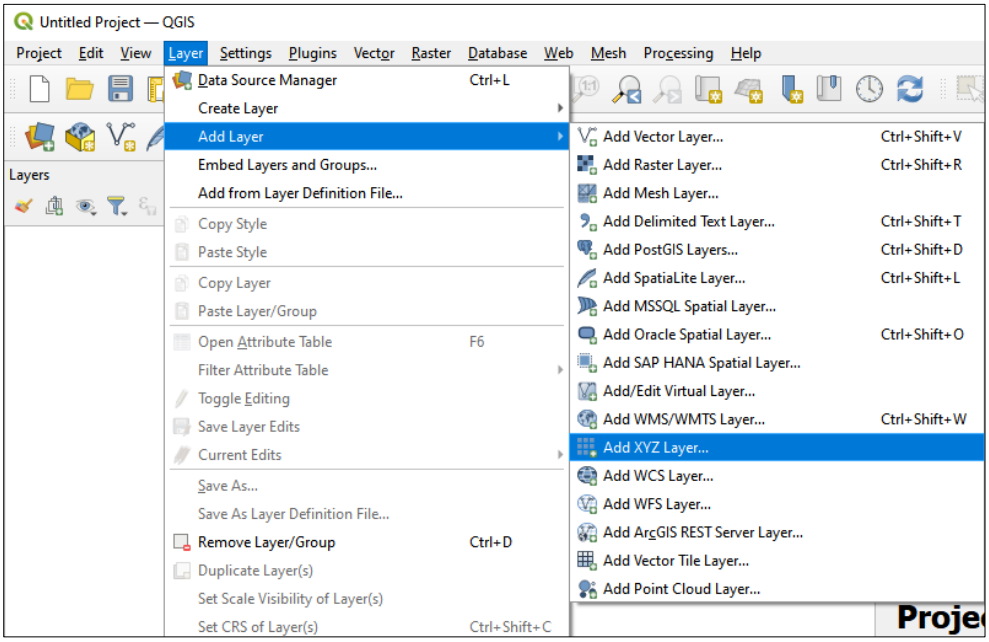
We'll create a new vizualisation inside QGIS with 3 layers:

- **Bottom layer:** the "Open Street Map" Tile layer to have "context".
- **Middle layer:** the catchment areas for all shops (i.e. the geometries from "SHOPS_CATCHMENT_AREA.shp").
- **Top layer:** many dots with the position of the shops and their name/id (i.e. the data from "SHOPS.shp").

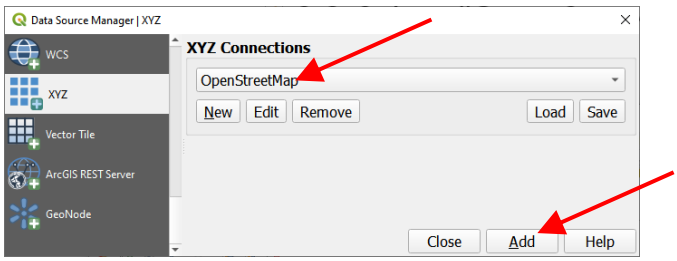
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here are the steps:

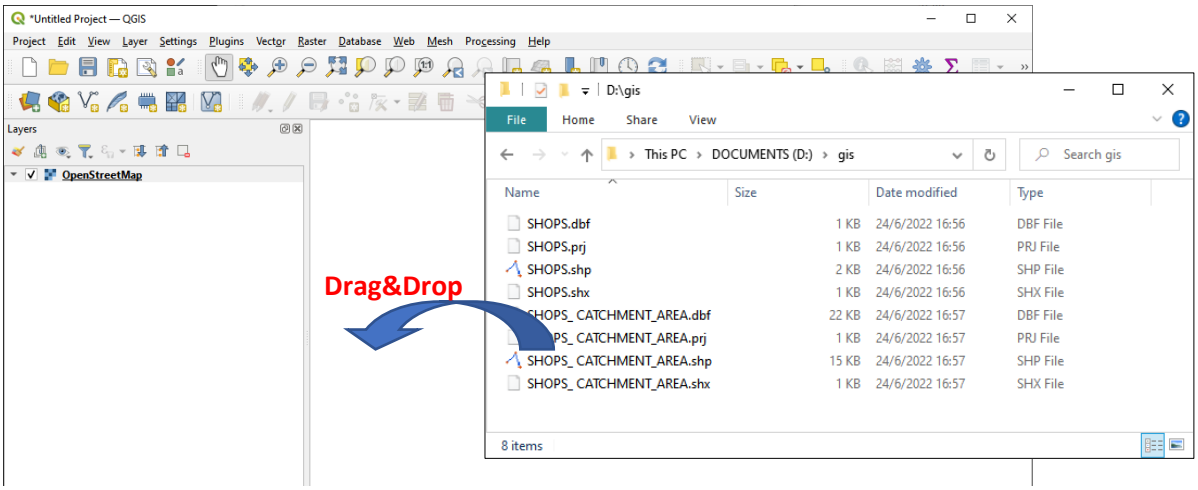
1. Run “QGIS Desktop”.
2. Add the “Open Street Map” layer:
   2.1. Open the dropdown menu “Layer”, select “Add Layer”, click on “Add XYZ Layer”:



   2.2. Select “Open Street Map” (this should be already selected) and click the “Add” button:

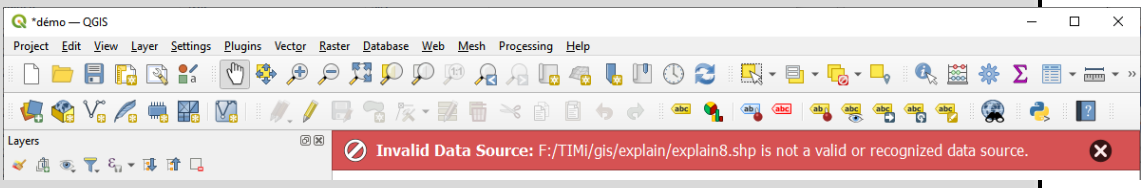


3. Add the layer with the “catchment area” for each shop:
   3.1. Drag&Drop the “SHOPS_ CATCHMENT_AREA.shp” inside the QGis window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

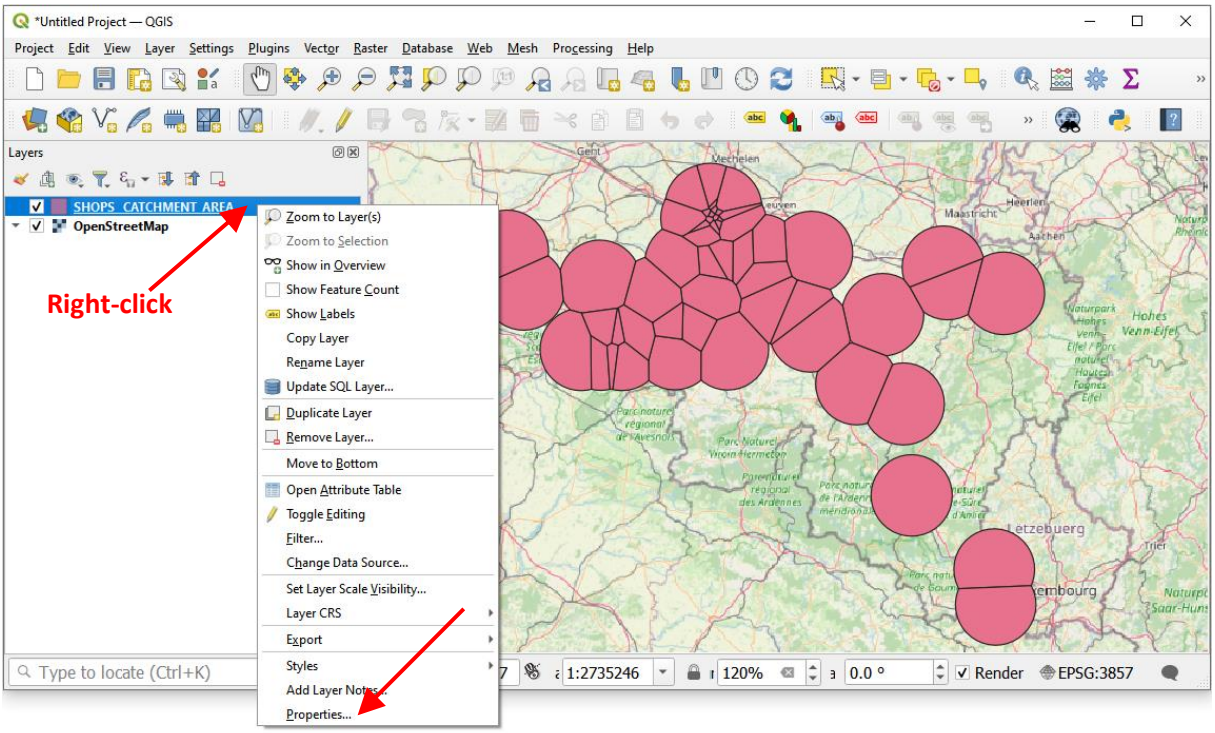
If you see this error message:



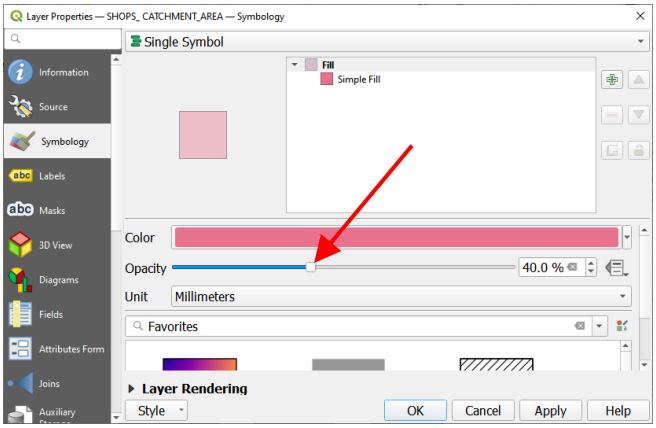
… it means that you have a .shp file but one of the other required accompagnying files (i.e. the .dbf file, the .shx file or the .prj file) is missing.

3.2. Make this layer semi-transparent:

- Right-click the new layer and select “Properties” inside the context-menu:

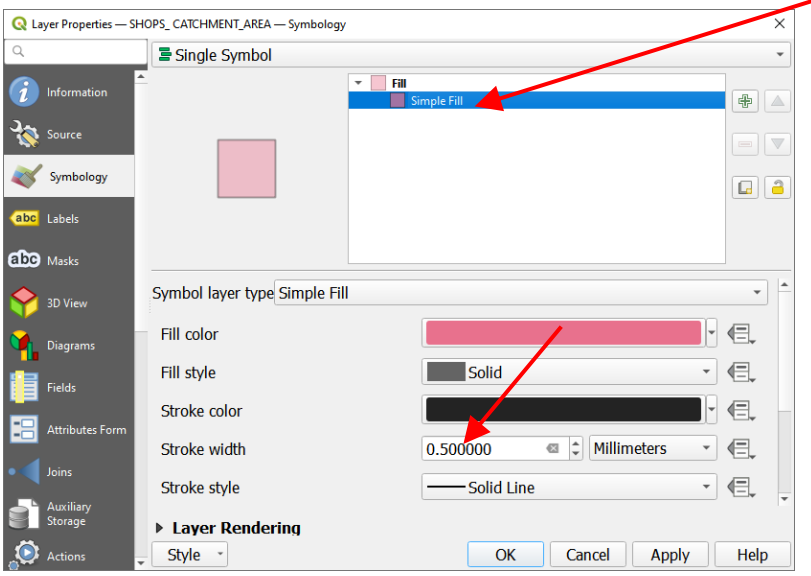


- Move the “Opacity” slider to 40%:

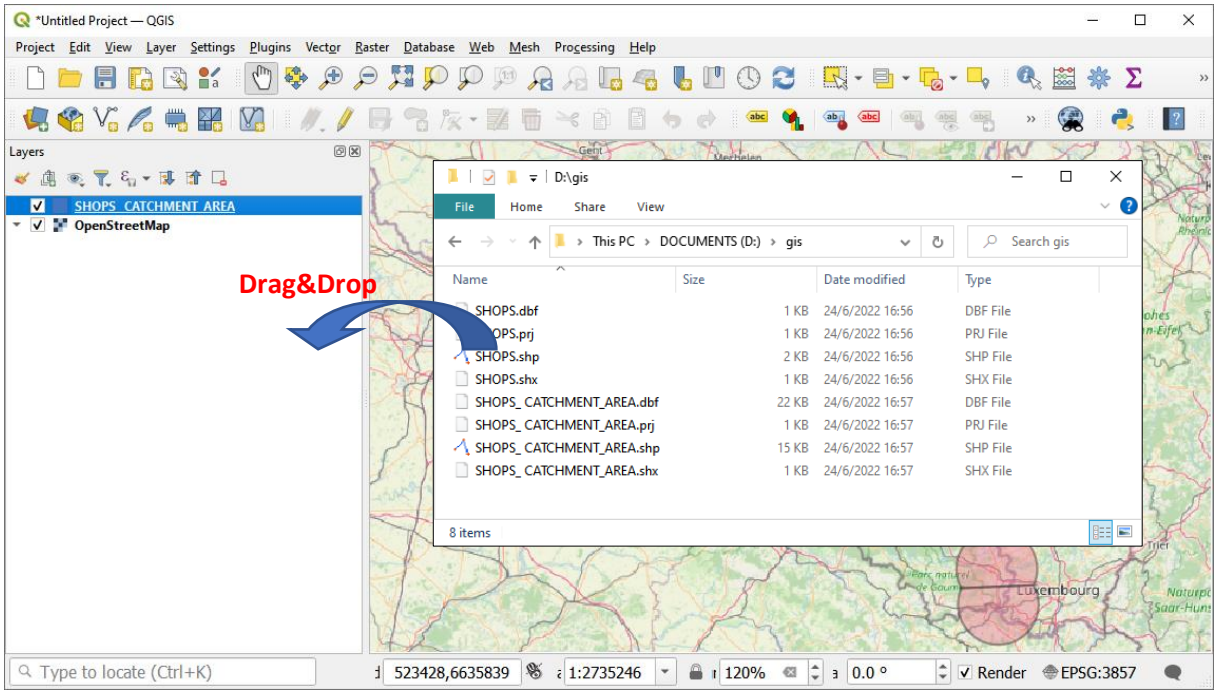


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3.3. Increase the thickness of the border of the geometries: Click on the “Simple Fill” style:
…and increase Stroke width to 0.5 millimeter:

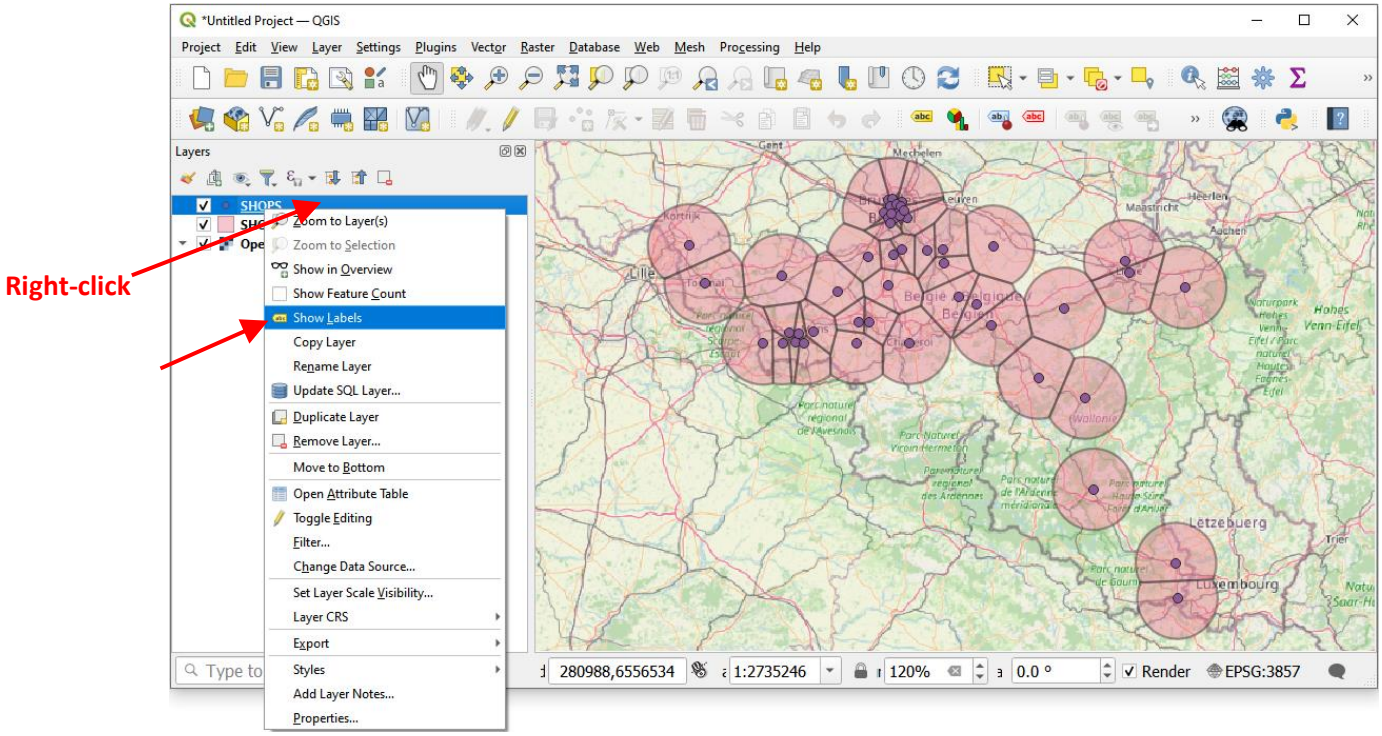


4. Add the layer with the positions of the shops:
4.1. Drag&Drop the “SHOPS.shp” inside the QGis window:



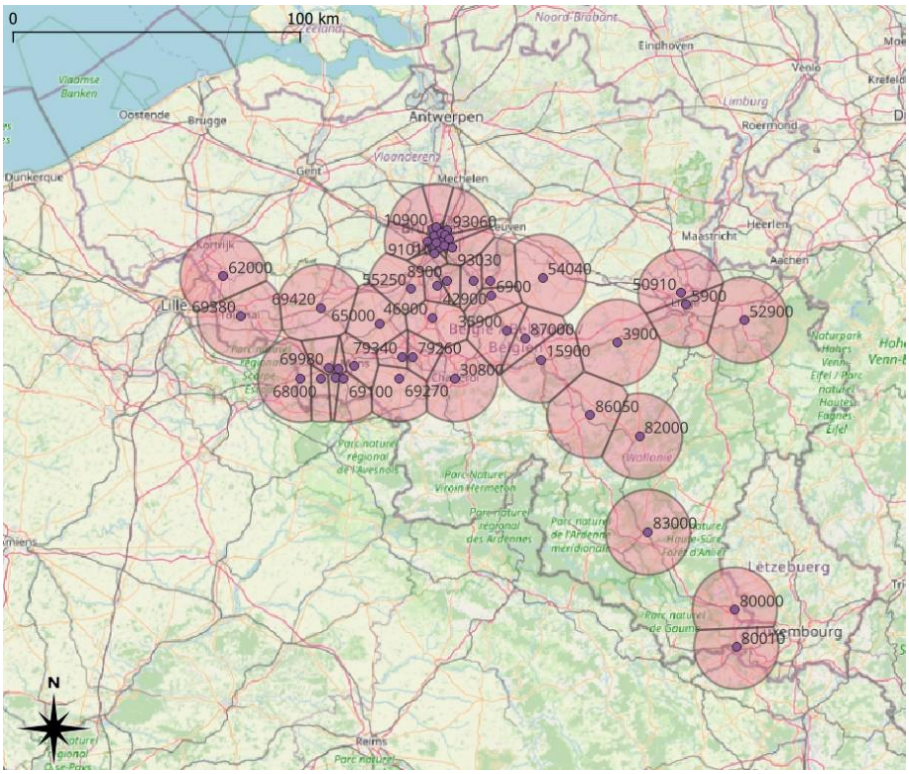
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.2. Add Labels to your shops: Right-click the new “SHOPS” layer and select “Show Labels” inside the context-menu (the Labels come from the content of the first column inside the .shp file):



5. (optional) Add decorations: Open the drop-down menu “View”, select “Decorations” and add a “Scale Bar” and a “North Arrow”.
6. (optional) Save your design (press [CTRL]+[S]) to be able to reuse your design to quickly produce an updated map with new, updated shape files.

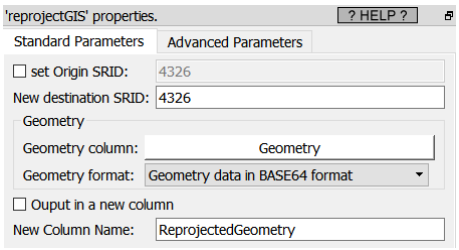
Here is the final result:

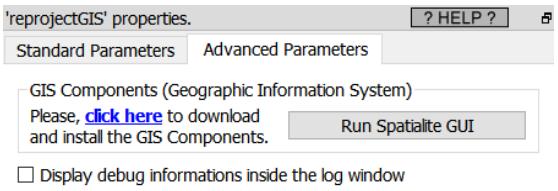


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.10. Reproject GIS

Icon:



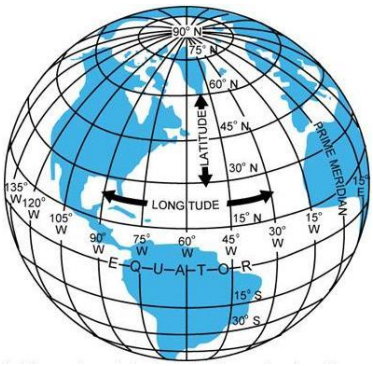


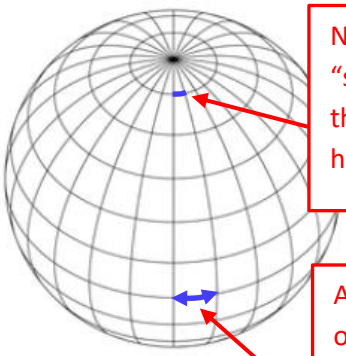
Property window:

Short description:
Reproject geometries
Inside a new coordinate system

Long Description:
Most of the visualization Actions inside Anatella do require to get their geometries described in decimal degrees (i.e. inside the coordinate system with the 4326 SRID).

One particularity of the "*SRID=4326 coordinate system*" is that a "step" of 10 degree of longitude (i.e. along the direction of the "X" axis) represents 354 Km at the equator (i.e. when the latitude is zero) and the same "step" of 10 degree of longitude near the Noth Pole (i.e. when the latitude is near 90°) is nearly zero Km. All this is explained in the following illustrations:

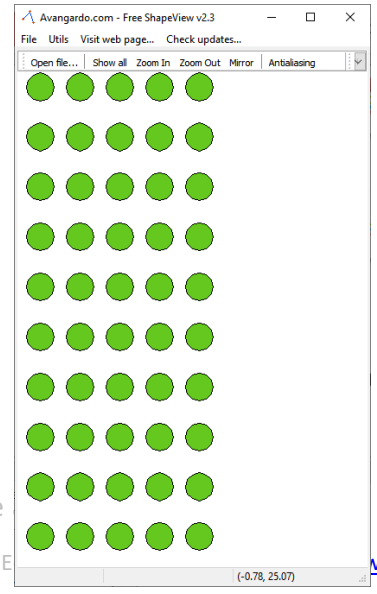




Near the North pole, a "step" of 10 degrees along the "X" axis (longitudinal) has a length of nearly 0 Km.

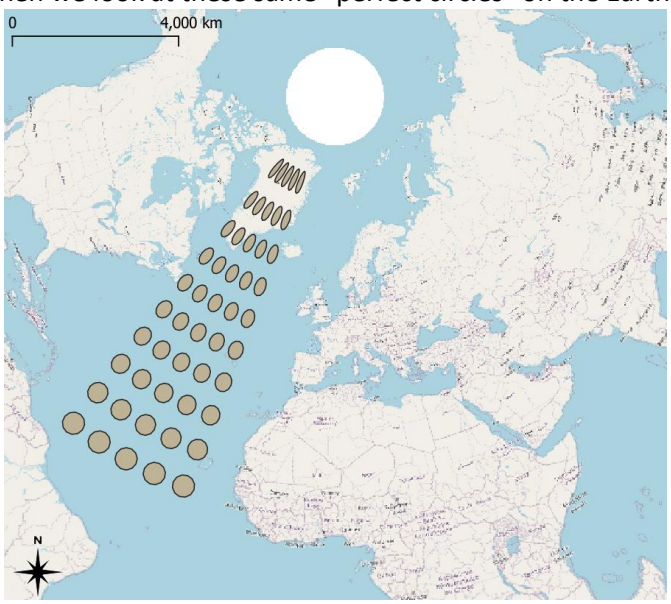
At the equator, a "step" of 10 degrees along the "X" axis (longitudinal) is equivalent to 354 Km.

In other words, inside the 4326 SRID, a step of one unit along the X axis does not always "moves us" of the same distance (it depends of the value of the Y axis). Hopefully, in the direction of the Y axis – the latitude – all the steps have the same contant length. So, moving in the X direction (i.e. changing longitude) and moving on the Y direction (i.e. changing latitude) is definively not the same. Here is an example of this strange phenomenon: We created some shapes that looks like perfect circles inside the "*SRID=4326 coordinate system*":



©2010 TIMi SPRL. E v.timi.eu

…but when we look at these same “perfect circles” on the Earth spheroid, we actually see:



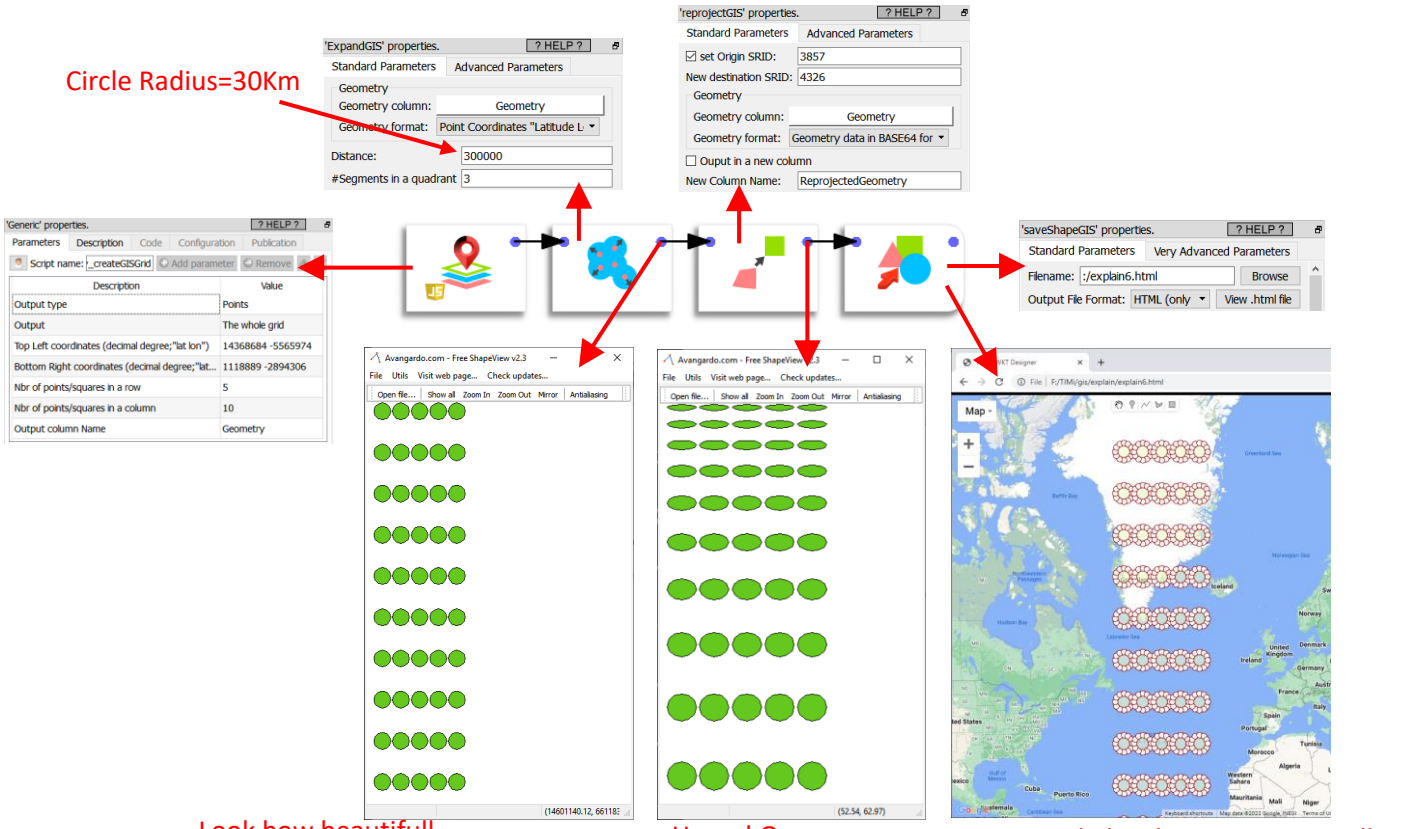
This example illustrates very well that the “X axis” (i.e. the longitude direction) “shrinks” when we approach the poles.

Many complex GIS Actions (such as these Actions: KNN, ConcaveHull, SimplifyGIS, ExpandGIS, TesselateGIS) do not work very well inside this “*SRID=4326 coordinate system*” because these Actions all assume that we are in a standard planar cartesian coordinate system (they totally ignore the fact that our geometries were first defined on a spheroid): i.e. doing a “step” of one unit-length always “move us” of the same constant distance whatever the direction of the step. Hopefully there is a solution to this problem: re-projection! The Anatella Reproject action allows to take the geometries defined inside the problematic 4326 and reproject them inside a new coordinates system that has the nice “planar&cartesian” properties (i.e. all directions are equivalent). Once inside this new coordinate system, you can thus safely use all the complex GIS Actions (KNN, ConcaveHull, SimplifyGIS, ExpandGIS, TesselateGIS) to do all your computations. Also, typically, at the end of all the computations, you’ll reproject inside the 4326 SRID, just for visualization purposes (because, most of the time, the visualizations require the 4326 SRID).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.10.10.1. A good, working example

Let's go back to our previous example: We again want to create a small grid of "perfect circles" on the Earth spheroid, but this time, we'll succeed:



Look how beautifull our grid of "perfect circle" is! (inside the 3857 SRID)

Ho no! Our "perfect circles" are completely distorted !

Yes! This distortion is actually required inside the 4326 SRID and we get back our beautifull grid of "perfect circle"

The 4 steps (for the 4 actions here above) are:

1. We generate a simple grid of points inside the 3857 SRID (using createGISGrid)
2. We transform each point into a small disc of radius 30Km (using ExpandGIS)
3. We want to display our grid of "prefect circles" on an Earth Map: To do so, we first need to reproject into 4326 SRID.
4. We create the .html Google Map based on the geometries in the 4326 SRID.

5.10.10.2. SRID and Storage Formats

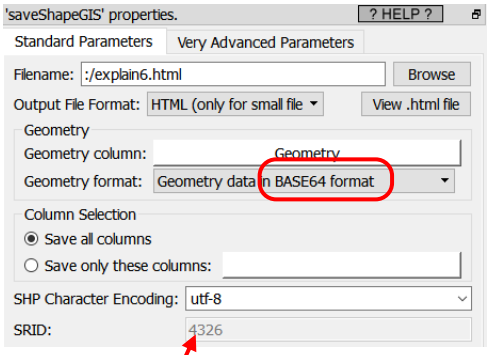
Inside Anatella, the geometries are stored/represented using 5 different formats:

1. Point Coordinates "Latitude Longitude"
2. Rectangle "Latitude1 Longitude1 Latitude2 Longitude2"
3. Geometry data in WKT format
4. Geometry data in HEX format
5. Geometry data in BASE64 format

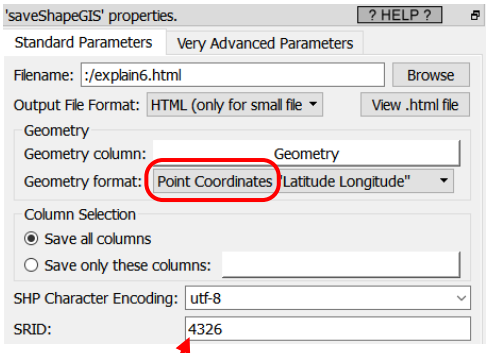
The data for the geometries stored inside the last 2 formats (the HEX and the BASE64 formats) include everything required to define the geometry: …And this includes the SRID! Using the GeometryInfo Action, you can see the SRID of the geometries (that are stored in HEX or BASE64 format).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In opposition, when using the first 3 formats (i.e. the POINT, the RECTANGLE and the WKT formats), some Actions will ask you to manually specify the SRID (because the SRID is not embedded into the data for these more primitive geometry formats). For example, when you use the saveShapeGIS action:



No need to specify the SRID because it's embedded inside the BASE64 format.



You need to manually specify the SRID because the geometry format is "Point coordinates".

5.10.10.3. How to select a good SRID/Coordinate system?

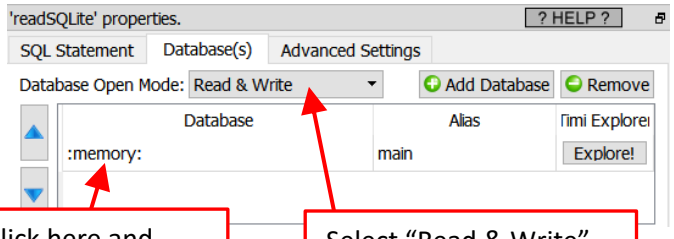
Usually, SRID's are linked to countries (SRID 2154 for France, SRID 31370 for Belgium, SRID 4269 for North America, etc.). So, if you are manipulating geometries from a specific country, a good starting point it to select the SRID for this country: This will be the SRID with the less distortion for your geometries.

Anatella supports reprojection into 4922 different SRID's. To get a table with the complete list of all the supported SRID's, execute the readSQLite action with the following parameters:

- The SQL command to run is:

```
select * from spatial_ref_sys_all
```

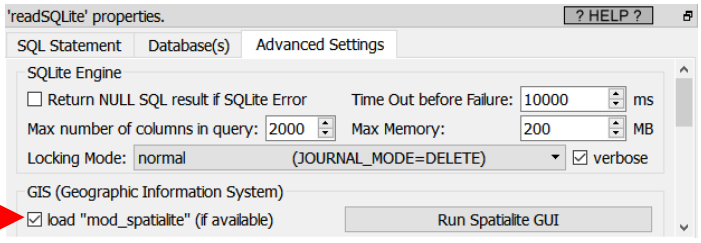
- The database name is ":memory:" and the "Database Open mode" is "Read & Write":



Right-click here and write ":memory:"

Select "Read & Write"

- Enable the GIS extensions:



Check this checkbox

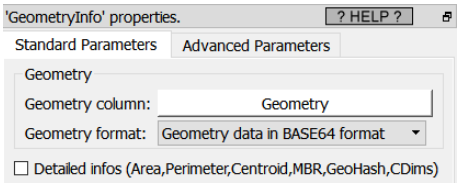
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When searching internet for a correct SRID, you can also search for EPSG because the SRIDs from spatialite (used in Anatella) are compatible with EPSG.

5.10.11. Geometry Info

Icon:

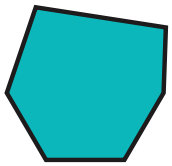
Property window:



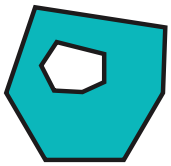
Short description:
Extract information from Geometries

Long Description:
The Extracted informations are:

- **GeometryType:** it can be: POLYGON,MULTIPOLYGON,POINT, MULTIPOINT,etc.
- **NPoints:** Number of points inside the geometry
- **IsValid:** 0 or 1
- **NRings:** Number of Rings:

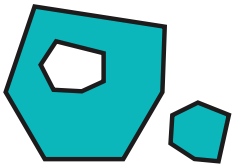


1 POLYGON with 1 RING



1 POLYGON with 2 RINGS

1 POLYGON with 3 RINGS



1 MULTIPOLYGON with 3 RINGS and 2 POLYGONs in it

- **NumGeometries:** The Number of geometries inside a MULTI-geometry: For example: The number of POLYGON inside a MULTIPOLYGON geometry
- **SRID:** see previous section 5.10.10. for more details on SRID
- **Dimension:** 0 for POINT, 1 for LINE, 2 for POLYGON
- **NDims:** 2 (when you have Lattitude and Longitude) or 3 (when you have Lattitude, Longitude and Elevation)
- **Is3D:** when nDims=3
- **IsEmpty:** 0 or 1
- **IsSimple:** 0 or 1

If you check the "detailed info" checkbox, you also get:

- **Area:** The area is usually expressed in m^2 (i.e. divide by 1.000.000 to get it in Km^2). In particular, for the SRID 4326, you also directly get the area in m^2 (personnaly, I expected the area in $degree^2$ for the SRID 4326 but it's in m^2. Cool!).
- **Perimeter:** The Perimeter is usually expressed in m (this is also true for the SRID 4326 !).
- **Centroid:** the POINT at the center of the geometry (in WKT format). It might be outside the POLYGON if the geometry as a "U" shape.
- **Envelope:** the minimum bounding rectangle (MBR) around the geometry (in WKT format)
- **GeoHash:** the geohash of the POINT (null if it's not a POINT)
- **CoordDimension:** most of the time: "XY", sometime: "XYZ"

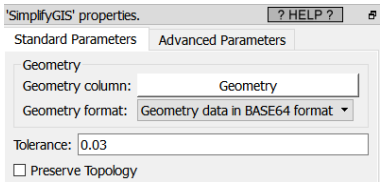
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

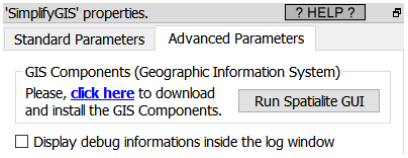
To remind you, a .shp file can only contain 1 type of geometry (e.g. it cannot contains both the POLYGON and the MULTIPOLYGON geometry type) and 1 unique SRID. Using the geometryInfo action is usefull to see if you can export your Anatella table to a .shp file: if you see several different geometry types or multiple different SRID, then the exportation will fail (this is the most common source of error when exporting to a .shp file).

5.10.12. Simplify GIS

Icon:

Property window:





Short description:
Simplifies geometries

Long Description:
Returns a "simplified" version of the given geometry computed using the Douglas-Peucker algorithm. This will actually do something only with (multi)lines and (multi)polygons but you can safely call it with any kind of geometry.

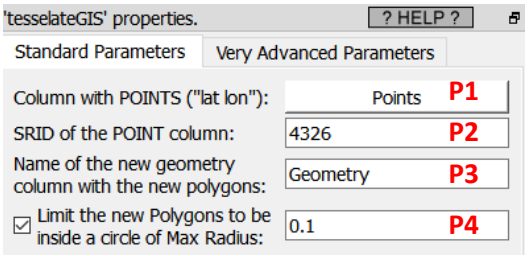
The Preserve Topology flag prevents oversimplification, otherwise the circle can at most become a square.

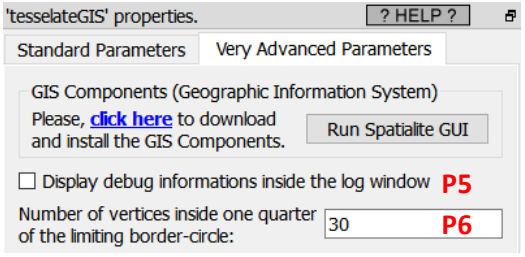
The Tolerance parameter is expressed in the actuall SRID of the geometries (i.e. it's in decimal degree if the SRID of your geometries is 4326). See the section 5.10.9.1. for an example of usage. This Action assumes that the geometries are defined in a standard PLANAR coordinate system (i.e. not inside the SRID 4326), so for best results, you shouldn't use it inside the SRID 4326.

5.10.13. Tesselate GIS

Icon:

Property window:





Short description:
Tesselate the plane

Long Description:

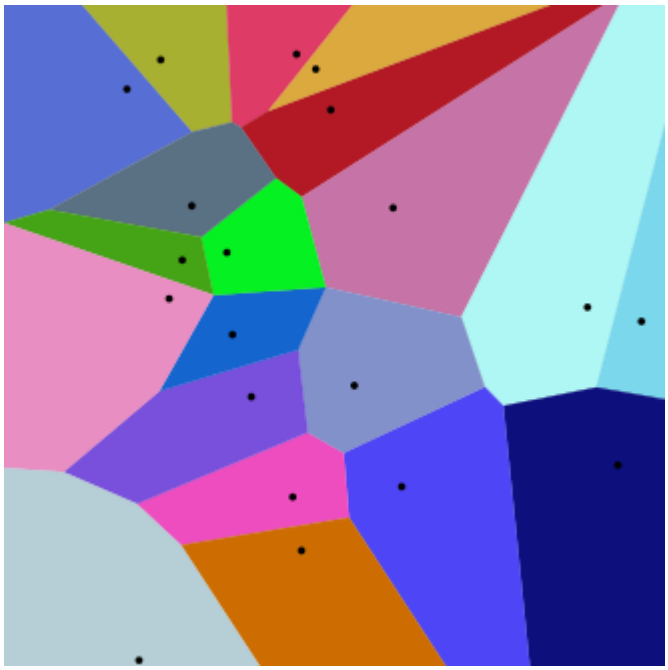
To contruct some POLYGONs from POINTs inside Anatella, you can either use the tesselateGIS action or the ConcaveHullGIS Action.

The general principle of the tesselateGIS is the following: It divides the plane into many small (convex) POLYGONs. The input of this Action is a set of POINTs that are named "Seeds". For each seed,

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

there is a corresponding POLYGON, called a Voronoi cell, consisting of all the points in the plane that are closer to that seed than to any other seed.

Here is an illustration:



The tesselateGIS action uses an algorithm named "Voronoi diagram". Unfortunately, these "Voronoi diagrams" are ill-defined on the border of the map: i.e. when we refer to the the "basic" definition of a Voronoi cell (i.e. the convex geometries given as the output of this Action), we see that these cells are **not finite** on the border of the map: i.e. They extend to the infinity. To avoid infinite geometries, Anatella limits the size of the cells to a disc centered at the seed of each cell. The radius of this disc is defined using the parameter **P4**. The number of points defining each disc is the parameter **P6** multiplied by 4.

The typical usage of the tesselateGIS action is to create "catchment area" for your shops. To do so:

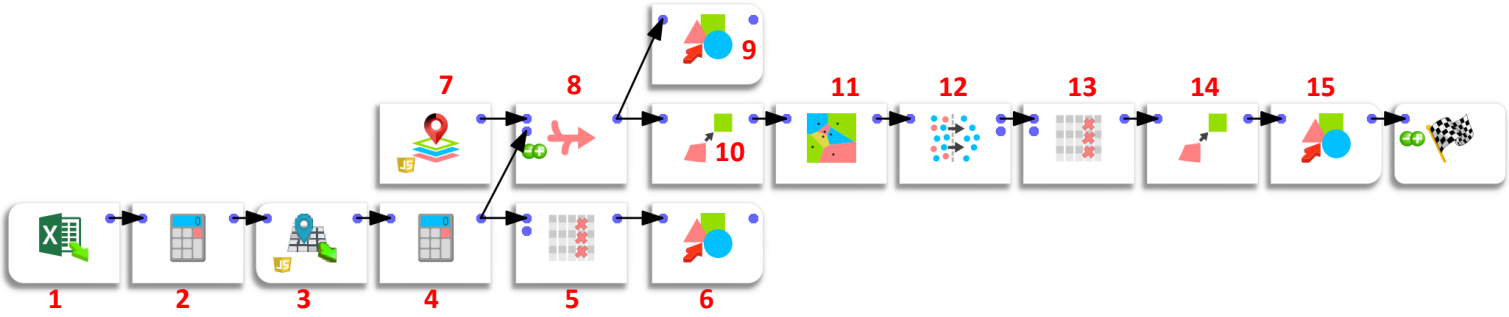
- You use as "seeds" (as input of the tesselateGIS action) the position of your shops.
- You define the maximum distance that a customer would deem reasonable to travel to your shop using the parameter **P4**.

..and then you run the tesselateGIS action. Then, you can look at your "catchment area" for each of your shop: Just follow the procedure given in section 5.10.9.3: This section contains this exact example about displaying "catchment areas" for your shops.

This tesselateGIS Action assumes that all your geometries are defined in a standard PLANAR coordinate system (i.e. not inside the SRID 4326), so for best results, you shouldn't use it inside the SRID 4326 (i.e. use the ReprojectGIS action to do all computations in a better SRID).

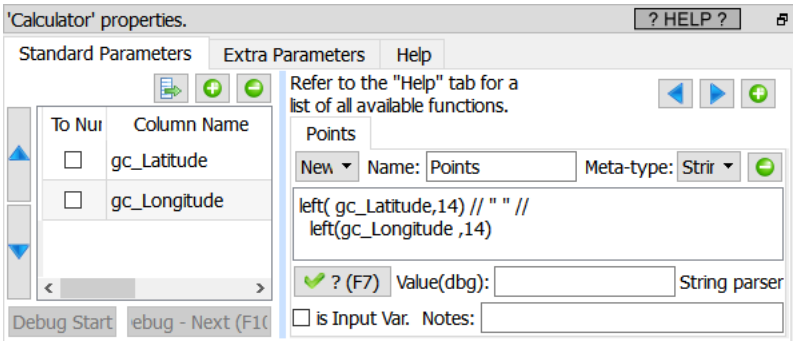
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is a complete project, from start to finish, to create a “catchment area” for each of your shops:

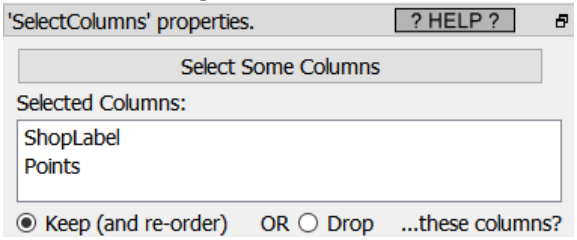


Here is a small description of each of the steps (illustrated with a RED NUMBER in the Anatella graph here above):

1. Extract the postal address of each of you shops from an Excel file.
2. Compute an “address” column that is the concatenation of StreetName, StreetNumber, Zip code, Town, Country
3. Use BING to find the Latitude and Longitude of your shops, using the “Address” column that we just defined during step 2 (see section 5.10.4 for more details about the bingMapGeocode action).
4. Create a “POINTS” column that is the concatenation of the columns “gc_Latitude” and “gc_Longitude” originating from BING. This column contains the “seeds” that we will send to the tesselateGIS Action to compute the “catchment areas” for your shops.

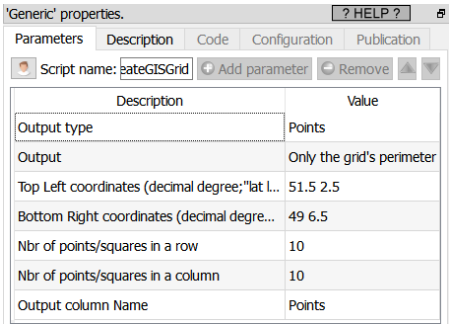


5. We want to create a .shp file with all the coordinates of all our shops (we do that in step 6). The first column of the .shp file contains the “label” of each point/shop. We use the selectColumn Action to select a good “label” column as our first column inside the .shp file:

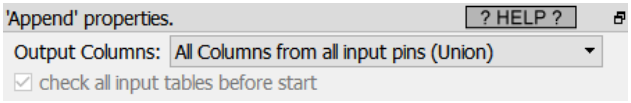


6. We create the “SHOPS.shp” file (see an example on how to use it inside the section 5.10.9.3.)
7. The tesselateGIS Action has the bad tendency to truncate geometries at the border of the map. So, we add some “artificial” points on the outline/border of the map (that we’ll remove during step 12) so that no geometry truncation happens for the “real” points that really represents our shops (that we obtained during step 4).

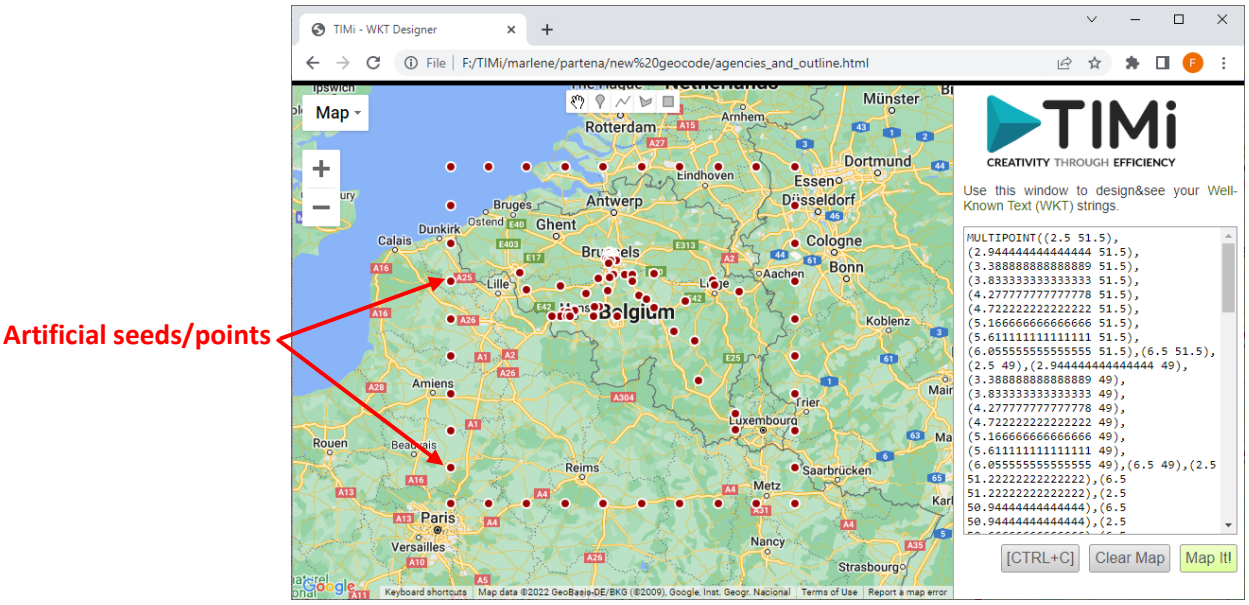
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



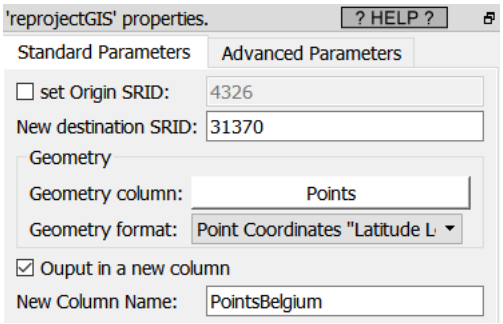
8. Create a new table with the "artificial" points on the borders (from step 7) and the "real" points (from step 4).



9. Validate visually the positions of the "artificial" seeds on the borders (from step 7) and the "real" seeds (from step 4):

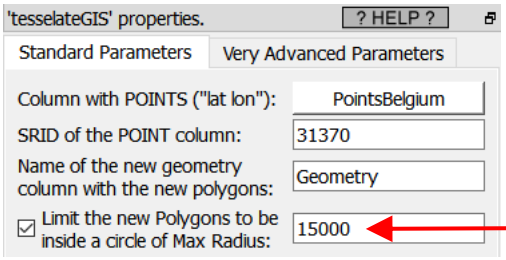


10. Reproject all the seeds defined in SRID 4326 inside the column "POINTS" to a new coordinate system (SRID 31370) which is the best for computation on Belgium. The new seed coordinates (inside the SRID 31370) are stored inside the new column named "PointsBelgium".



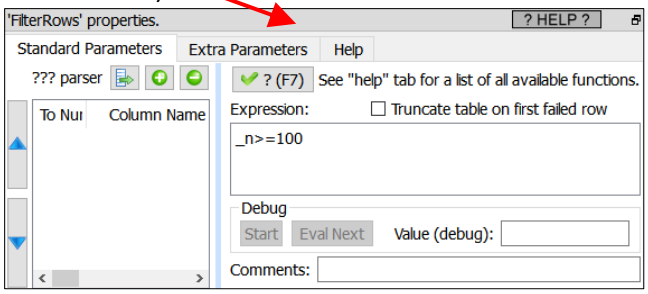
11. Compute the Catchment Area for all the shops inside the SRID 31370 (because we used as "seeds" the column "PointsBelgium" that is in SRID 31370).

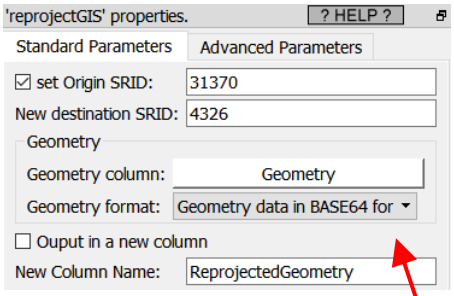
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



We decided that the maximum travelling distance that a customer is willing to do to arrive to our shop is 15 Km.

12. Remove the "artificial" seeds because we don't need them anymore (i.e. remove the first 100 rows):

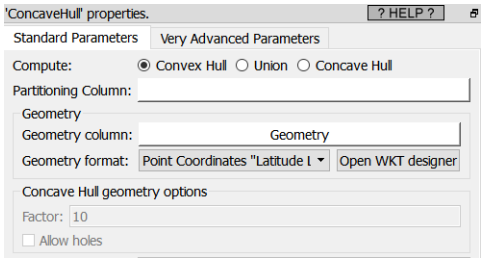


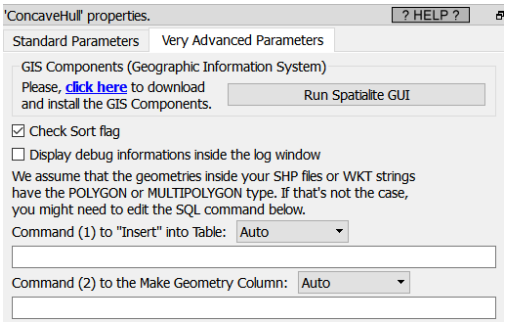


13. Remove/drop the column "PointBelgium" because we don't need it anymore.
14. Reproject our geometries (i.e. our "catchment areas") inside the SRID 4326 for visualization.
15. Save our geometries into the "SHOPS_ CATCHMENT_AREA.shp" shape file with the geometry column that contains the polygons that defines the catchment area for each shop. One row per shop.

5.10.14. ConcaveHull GIS

Icon:





Property window:

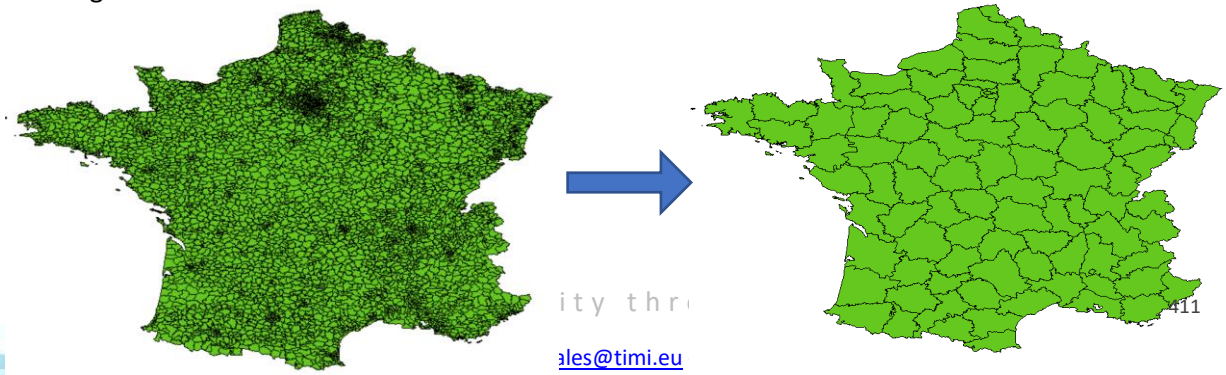
Short description:
Create POLYGON's from GEOMETRIES

Long Description:

To contruct some POLYGONs from POINTs inside Anatella, you can either use the tesselateGIS action or the ConcaveHullGIS Action.

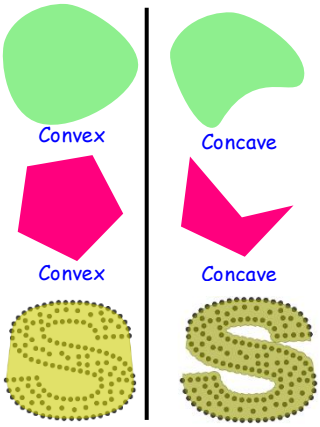
There are, basically, 2 use cases to use this Action:

- Use the "Union" mode to merge many small POLYGONs into bigger MULTIPOLYGONs: See the section 5.10.9.2. for an example where we merge tother 6048 geometries into only 95 geometries:



- Use the Convex/Concave mode to recreate a POLYGON from a set of points. If the input geometries are not points, Anatella will dissolve the input geometries into many points before recreating POLYGONs from the points.

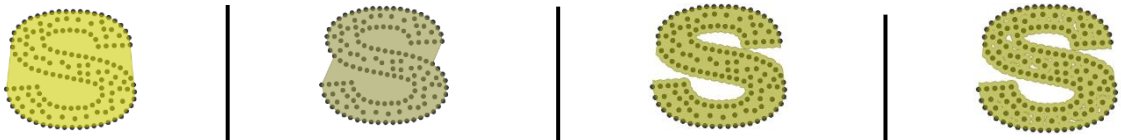
A small definition:



A polygon is convex when there are NO "dents" or indentations in it (no internal angle is greater than 180°)

The opposite idea is called "concave".

Given in input a set of points that “looks like” a “S” shape, the ConcaveHull GIS action produces as output this geometry (in yellow):



Mode=Convex

Mode=Concave
Factor=0.99

Mode=Concave
Factor=0.90

Mode=Concave
Factor=0.90
Allow Hole=True

To create a ConcaveHull, the basic approach is that it first creates a convexhull of the geometry and then uses the ST_ClosestPoint function to cave in the hull to transform it into a concave hull.

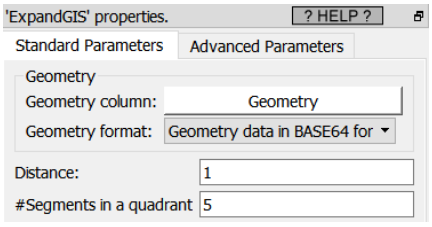
The typical use case is to create a POLYGON from a set of coordinates that represents different given classes of Customers (one class per partition). The ConcaveHullGIS Action will then (re-)create a POLYGON that encompass all the Customers inside the same class.

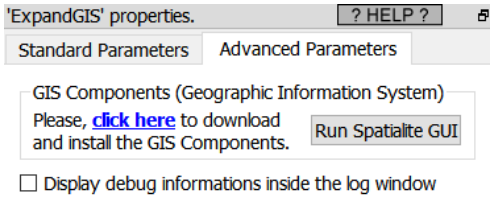
This Action assumes that the geometries are defined in a standard PLANAR coordinate system (i.e. not inside the SRID 4326), so for best results, you shouldn’t use it inside the SRID 4326 4326 (i.e. use the ReprojectGIS action to do all computations in a better SRID).

5.10.15. Expand GIS

Icon:

Property window:





Short description:
Expand the contour of Geometries

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

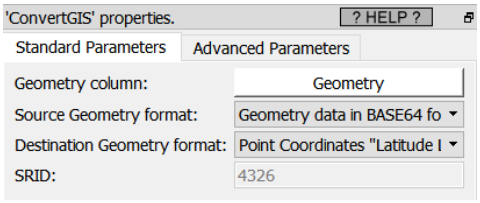
Long Description:
Self-explanatory. See the section 5.10.10.1. for an example of use.

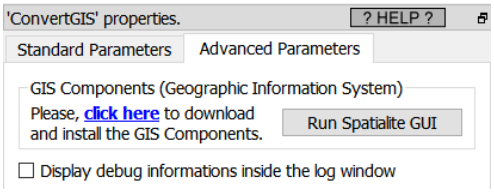
This Action assumes that the geometries are defined in a standard PLANAR coordinate system (i.e. not inside the SRID 4326), so for best results, you shouldn't use it inside the SRID 4326 4326 (i.e. use the ReprojectGIS action to do all computations in a better SRID).

5.10.16. Convert GIS

Icon:

Property window:





Short description:
Convert the format used to store Geometries

Long Description:
Inside Anatella, the geometries are stored/represented using 5 different formats:

1. Point Coordinates "Latitude Longitude"
2. Rectangle "Latitude1 Longitude1 Latitude2 Longitude2"
3. Geometry data in **WKT format**
4. Geometry data in HEX format
5. Geometry data in BASE64 format

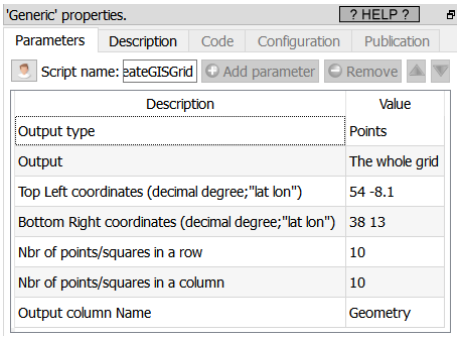
This action allows you to convert from one format to another.

When using the ConvertGIS action, most of the time, you have as many rows in output as in input. The only exception to this rule is when you convert towards the geometry format "Point Coordinates". When this happens, Anatella first dissolves all the input geometries into the POINTS that composes them. Then, it ouputs one different row for each different POINT. For example, this means that, if you have in input a table with 2 rows and, on each row, a POLYGON composed of 4 POINTs, then you'll get as ouput a table with 8 rows/POINTS.

5.10.17. Create GIS Grid

Icon:

Property window:



Short description:
Create grid of (points or rectangles)

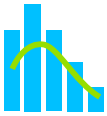
Long Description:
Self-explanatory. Some examples:

- The paragraphs 7 and 9 inside the section 5.10.13 (about Tesselation).
- Section 5.10.10.1.

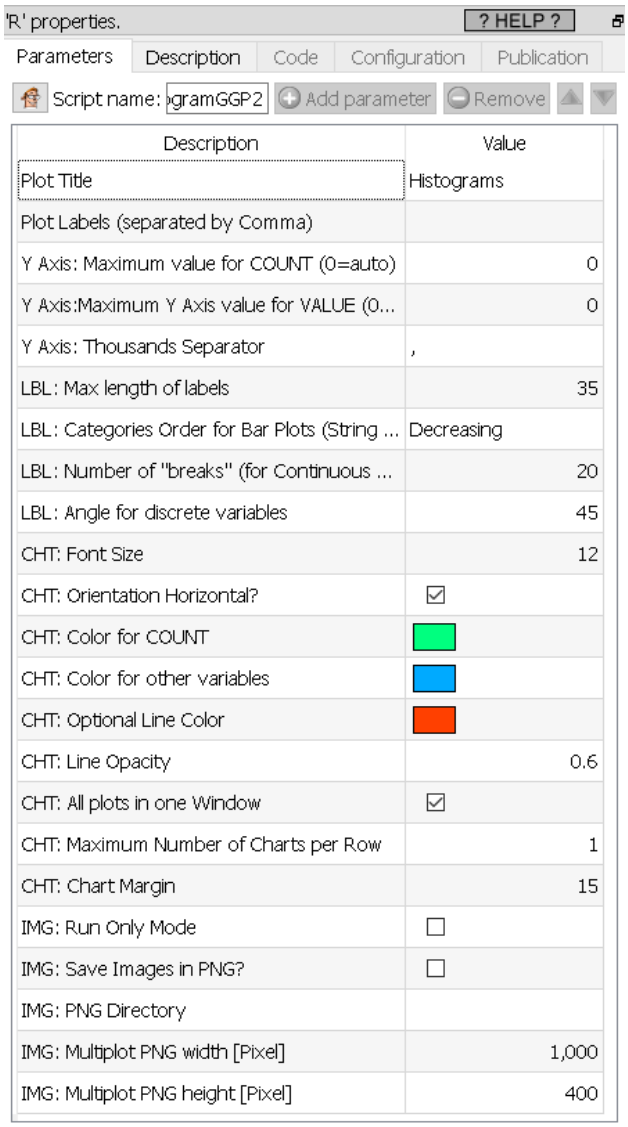
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.11. TA – R Visualization (TA=Transformations Actions)

5.11.1. Histogram (R action)

Icon: 

Property window:



Short description:
Create a Histogram chart from an already aggregated table

Long Description:
Create a Histogram chart from an already aggregated table. Each input corresponds to the results of and Aggregate action.
If you add a third dimension, it will automatically be set as a line.

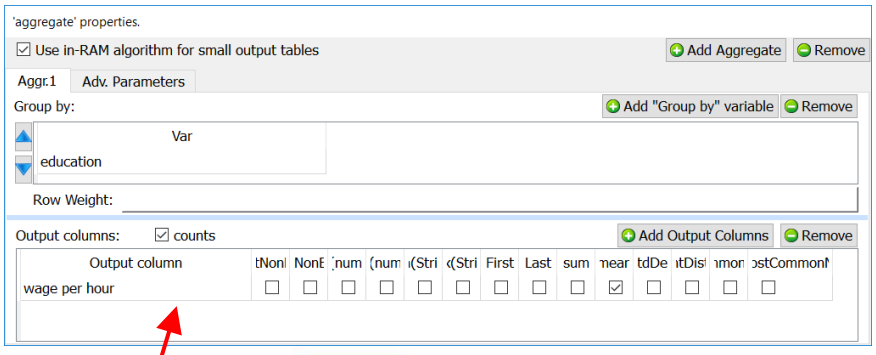
Parameters:

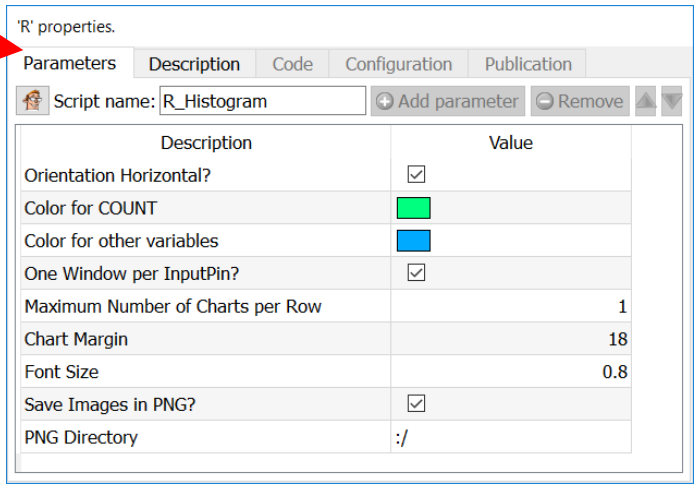
Y Axis Maximum value: set the maximum value on the plot, so you can compare all your variables on the same scale.
Y Axis thousands separator: what character to uso to display the thousands
LBL Max length of labels: maximum number of characters displayed
Orientation: By default, histograms are verticals (categories are on the X axis). Use this option to put the categories on the Y axis
Color for COUNT: select the color for a simple count
Color for other variables: select the color for *double* values (mean, stdev, etc.)
One window per input pin: if you have multiple charts, you may display them in a single window.
Maximum number of charts per row: if you selected the previous option, how many charts can fit in a row.
Margin between charts: set the parameter to make charts easier to read.
Font Size: set the font size (default is 0.8)
Save Images as PNG: choose whether to save the output as a PNG file or not
PNG Directory: choose where to save the file (by default in the active directory ":/")

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Example

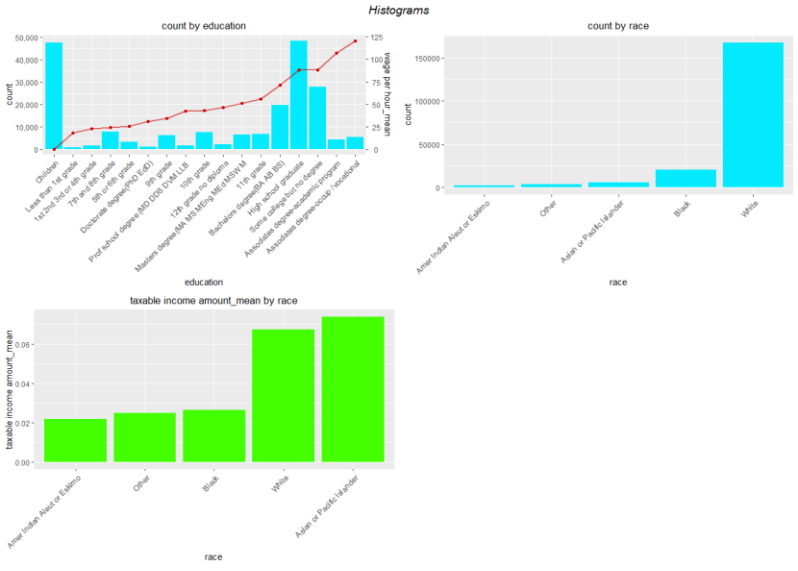
Using the Census Database, we want to make histograms of the difference in wage per hour, for each education level, We will first make an aggregate to compute the values to display:





.. and set the proper parameters to display the histogram. In this case, we simply modified the default value of "Chart Margin" to properly display the labels of the education levels, which can be quite long.

This automatically generate the following plots. Note that we chose to keep all plots in the same window, in many cases it is preferable to have one single plot per chart. To do so, simply unckeck the option "*All plots in one Window*". You can add up to 20 aggregations in a single "histogram plot".

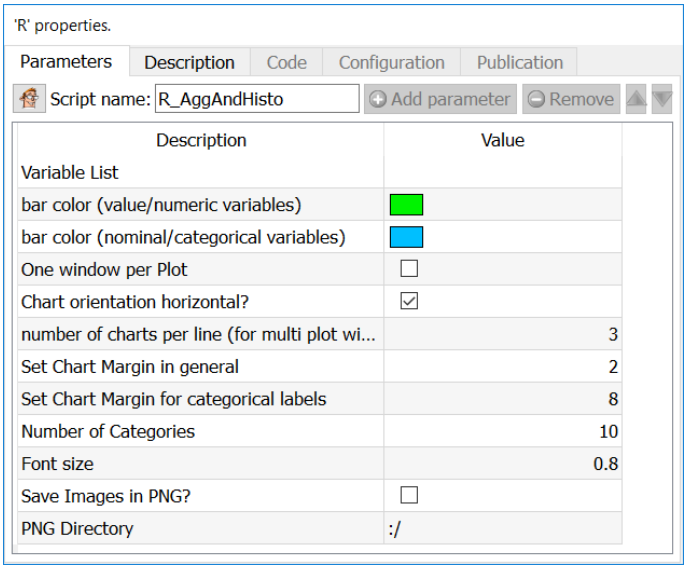


5.11.2. Aggregate & Histogram (R action)

Icon:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Property window:



Short description:
Aggregate a table and display the variables content in a Histogram chart.

Long Description:
Since the aggregation is performed by the R engine, it's limited to tables that fits into RAM (and it's quite slow). To display histogram on unlimited size tables, use the Histogram Action from the previous section.

Parameters:

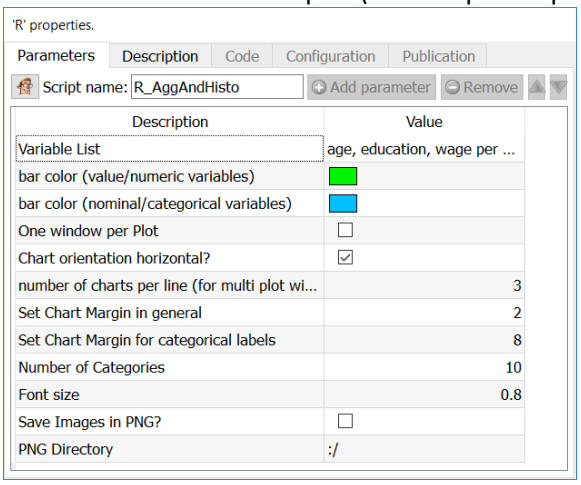
Variable list: Select the variables on which to compute the histograms (or feed from second pin)
LBL: Number of categories: set the number of bins in the histogram (X Axis)
LBL: Label angle for discrete variables: self explanatory
LBL Max Length of Label: set the maximum number of characters to display on X axis.
LBL Font Size: set the font size (default is 0.8)
LBL Thousands Separator: set the thousands separator for easier readability
LBL Categories Order: set the order to display (aqlphabetical, or based on value)
CHT Bar Color for Nominal Variables: select the color for categorical variables
CHT Bar Color for Numerical Variables: select the color for value variables
CHT One Window per Plot: put the histogram of each variable in a separate plot
CHT Chart orientiation horizontal: by default, plots are vertical histograms
CHT Number of charts per line: set how many charts are on the same line, if you have not chosen to make one window per plot.
IMG Run Only Mode: Generate the chart to be saved, but do not display them
IMG Save Images as PNG: choose whether to save the output as a PNG file or not
IMG PNG Directory: choose where to save the file (by default in the active directory ":/")

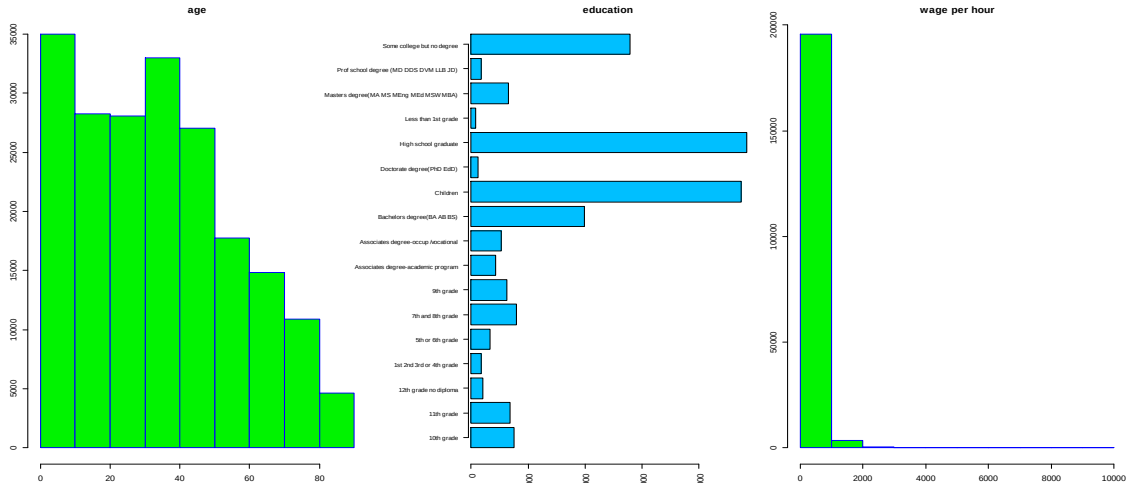
Example
Using the Census Database, we want to make histograms of wage per hour, education level, and age. We will first set to "FLOAT" the type of data for age and Wage per Hour (the type of variable is automatically sent to R, and treated as such).

Then, simply select the variables you wish to generate a histogram for (in our example, age, education, and wage per hour). As "age" and "wage per hour" are both numeric, R will automatically generate a histogram, while we will get a count plot for each category of the "education" variable. As in the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

previous example, we kept all the plots in a single window. By selecting the option "*One Window per Plot*" this woud have create a new window for each plot (three separate plots)





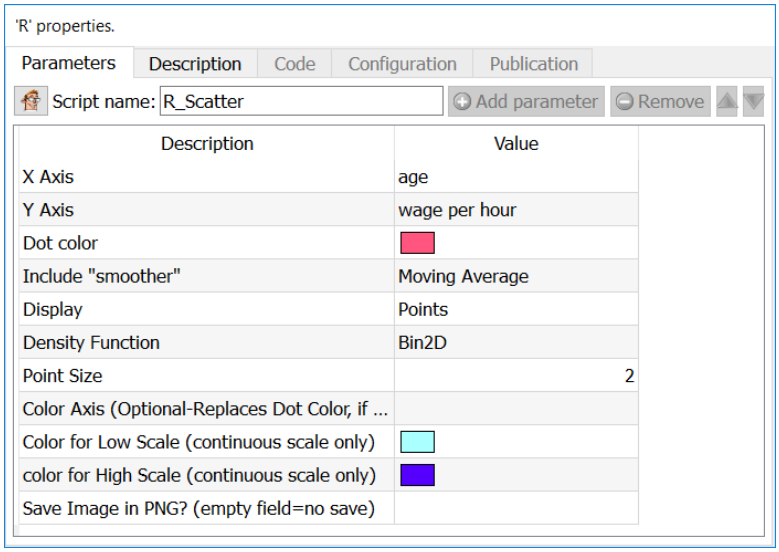
5.11.3. Scatter Plot (R action)

Icon:

Property window:

Short description:
Display a Scatter Plot

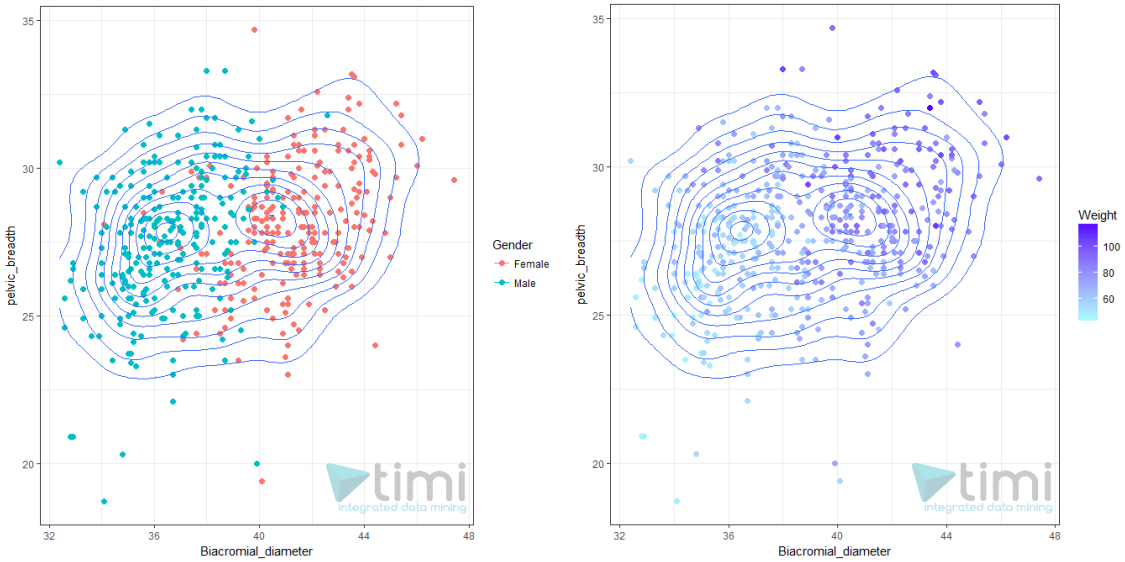
Long Description:
Display a Scatter Plot of two series.



To add several series on the same plot, simply append your series vertically, and specify a "partition variable".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Variables must be numerical, and are automatically converted as they are sent to R for plotting.
The **density function** allows you to visualize the most likely bivariate distributions. The **CHT Color Axis** include a "Color Axis", which can either be "nominal" (left chart) or "continuous" (right chart):

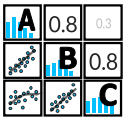


"Include Smoothen" will display a smoothen line to illustrate the relationship between the two variables. It can either be a linear trend (correlation) or a moving average. In either case, the confidence interval is also automatically displayed. You can also add a custom function, or pre-made elasticity curve to explore the relationship between Price and Demand.

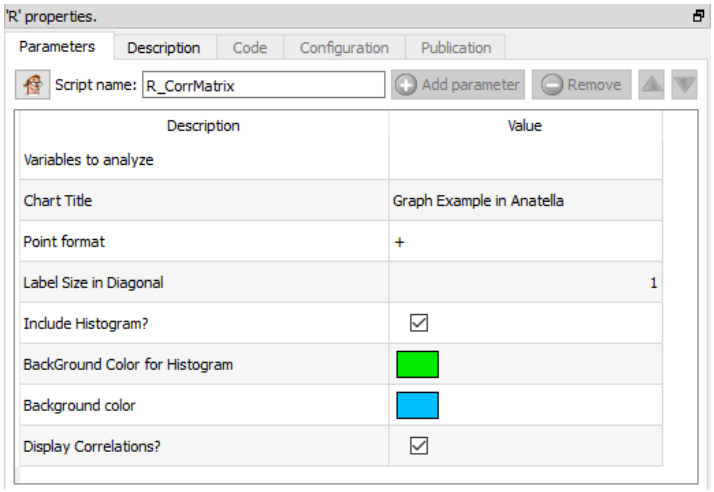
Y Axis is a count: in many situations, we want to show aggregated data (for example, to show a demand curve). This makes very few points displayed on the chart, and the "n" row information will make no sense. To keep n as the true number of rows, click this option and the plot will display the sum of Y.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.11.4. Correlation Matrix (R action)

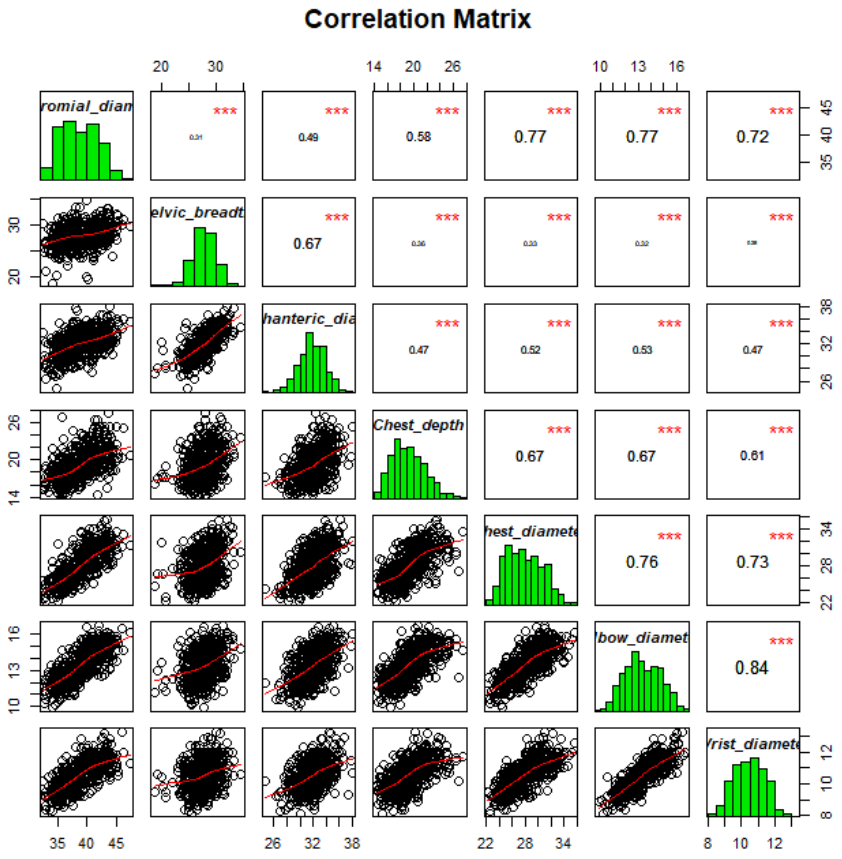
Icon: 

Property window:



Short description:
Display a correlation matrix

Long Description:
This is usefull to visually check if some variables are correlated.
There are two versions of Correlation Matrix included in Anatella, one uses regular plots, and allows the combination of scatter pluts, histogram and trends (moving average)

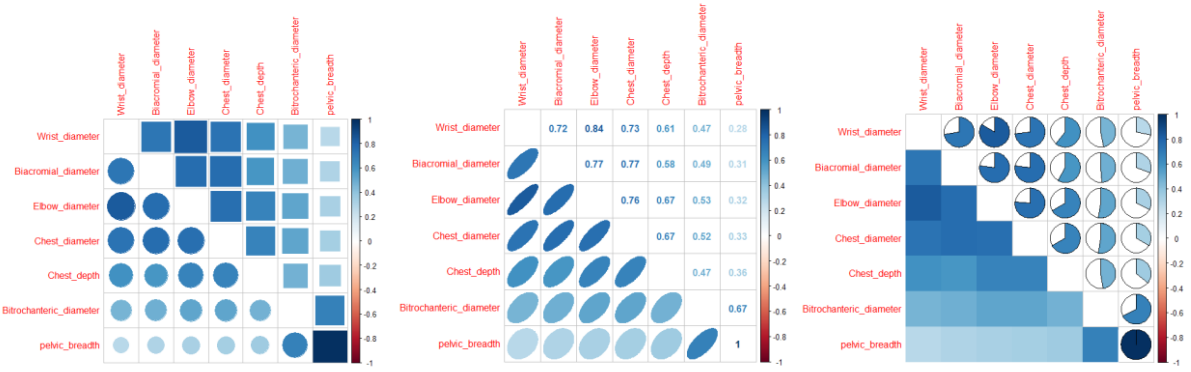


While this offers a sound understanding of the variable distributions, and perhaps provides a better representation of the data, this is not the mode “business ready” visualization.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.11.5. Correlation Matrix V2 (R action)

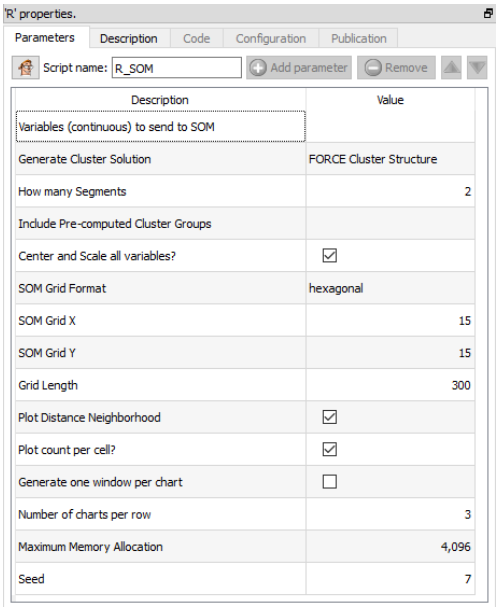
The newer (and much faster) Correlation Matrix Action allows more graphical representations. The following examples show a representation of circles, squares, ellipse, number, shade and pie.



5.11.6. Self Organizing Maps for data visualization (R action)

Icon:

Property window:



Short description:
Display (and optionnaly compute) a clustering using Self-Organizing-Map.

Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

SOM (Self Organizing Maps), or Kohonen Maps, are a crude form of dimension reduction. The multivariate space is reduced to two dimensions with a preset number of ordinal categories, and the density is then represented by a heat map.

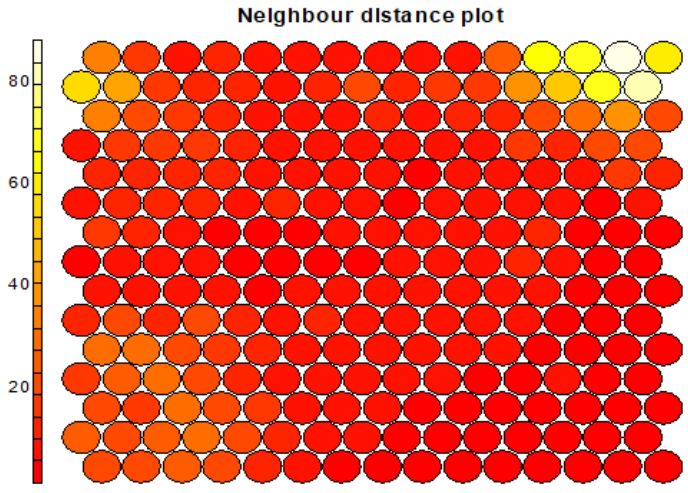
The objective of the SOM action is, mainly, to illustratre, using nice colors, some meaningful segments created with another segmentation algorithm (such as K-means, Wards, etc.). Do no use the SOM action to actually compute any segmentation because it will most likely find a segmentation that does

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

not really exists in your dataset (i.e. the SOM algorithm is one of the worst algorithm that you can find to discover the true segments hidden in your datasets).
In this implementation, we are using the CLASS and KOHONEN packages in R, which allow for the following settings:
Variables (continuous) to send to SOM: select the variables on which to compute the Kohonen space. The variables must be numerical
Generate Cluster Solution:

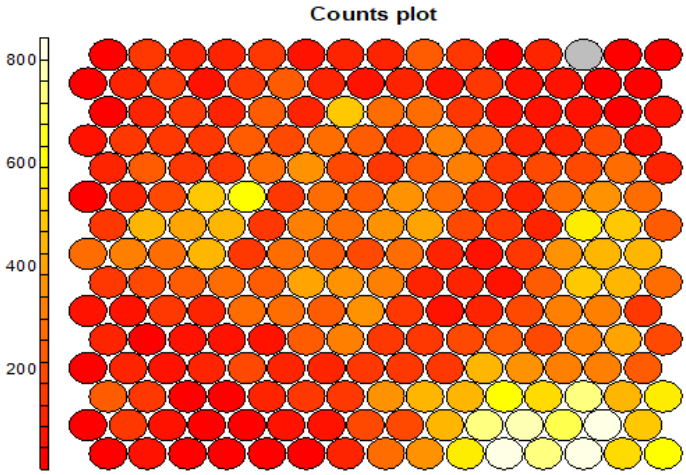
1. NO Clustering: this will generate a neat Kohonen map, nothing tricky about it.
2. Apply Cluster Variable: the space generated in step 1 is respected, but we will apply the cluster membership from the previously computed segment
3. FORCE Cluster solution: we will force the segment structure to affect the kohonen space, ensuring segments are well represented, but at the cost of a harder to read map
4. Compute from SOM: this will run a hierarchical clustering on the available data, respecting the original Kohonen space, but at the risk of having segments that may not exist. DO NOT USE THIS OPTION since it will generate poor quality segments.

How Many Segments: Select the number of segments to compute. If you force a variable, this value is overruled.
Include Pre-computed cluster group: Select the variable with (numerical) cluster membership
Center and scale variables: self explanatory. Although this feature is mostly used when variables are on different scales, it usually gives better maps when used.
SOM Grid Format: Select the topologuy of the SOMGRID object, Circular or hexagonal.
SOM Grid X: Select how many nodes will be included on the X axis.
SOM Grid Y: Select how many nodes will be included on the Y axis.
Grid Length: the number of times the dataset will be presented to the network
Plot distance Neighborhood: choose wheter to output the distance plot. Plot show how distant each node is from its neighbors. In this example, the lighter the color, the larger the distance, hence the first two nodes of the second line, and the last 4 nodes of the first two lines are relatively far from the rest of the distribution.



Plot Count Per Cell: each node typically includes a variable number of respondents (records). This chart give a good feel of how unbalanced the map is. The lighter the color, the higher the number (the lower right portion has cells of over 800 records, while the left corners have small groups of less than 200 records)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

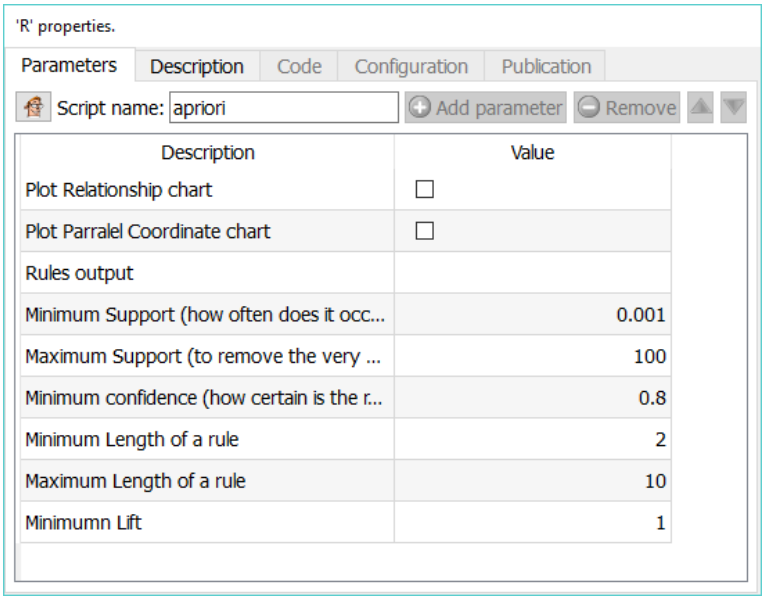


Generate One window Per Chart: Select if you want each variable in a separate map, or a map with all the variables next to one another
Number of charts per rows: if you did not select the previous option, this sets how many chart per rows will be included in the plot window.
Seed: set the random seed so you can reproduce the exact same map, or set other starting value if you do not like the results.

5.11.7. A priori (R action)

Icon:

Property window:



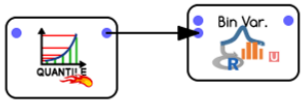
Short description:
Computes a priori association rules. Provides the infrastructure for representing, manipulating and analyzing transaction data and patterns (frequent itemsets and association rules). Also provides interfaces to C implementations of the association mining algorithms Apriori and Eclat by C. Borgelt (https://cran.r-project.org/web/packages/arules/arules.pdf)

Long Description:
Apriori looks for patterns in transactional data. There are many additional parameters in this package https://cran.r-project.org/web/packages/arules/arules.pdf that are not defined as user-parameters in

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

this Anatella action, feel free to complete the current list of parameters with any additional parameters to your liking.

Unlike other implementation, this action does not require the data to be in transactional form (such an invoice) as it will look for patterns across columns, so very little data preparation is needed. Basically, we need the data in the form of one line per transaction, with ALL DATA CATEGORICAL. If you have continuous data, such as age, or income, use the quantile and BinVar functionalities to recode them



The data you typically will analyze looks like:

Customer	Items
1	orange juice, coke
2	milk, orange juice, window cleaner
3	orange juice, detergent
4	orange juice, detergent, coke
5	window cleaner

The goal of association rules it to establish relationships. Here is an example:

	OJ	Wi Cl	Milk	Coke	Detergent
OJ	4	1	1	2	2
Wi Cl	1	2	1	0	0
Milk	1	1	1	0	0
Coke	2	0	0	2	1
Detergent	1	0	0	0	2

Rules are detected and kept based on two criterion: Support and Confidence

- **Support**: How often is the rule occurred? (usefulness)
  It is the percentage of transactions that contains all items in the rule
  - Support ($A \Rightarrow B$) = $P(A \cap B)$
  - Example: For the rule "*If Coke then OJ*" (or "*if OJ then Coke*"): In all 5 transactions, 2 contains both coke and OJ. The support of the rule is 40%
- **Confidence**: How certain is the rule?
  - Confidence ($A \Rightarrow B$) = $P(B \mid A)$
  - Example1 :
  - The rule ("If Coke then OJ") has a confidence of 100% because "Coke $\Rightarrow$ OJ = 2 / 2 = 1 (100%)"
  - Example2: What is the confidence of the rule "If OJ then Coke"?
    Out of the 4 transactions with OJ, only 2 have coke
    Confidence OJ $\Rightarrow$ Coke = 2 / 4 = 0.5 (50%)

Useful rules have improvement (lift) greater than 1. The lift is defined as:

$$Improvement(Lift) = \frac{\text{Confidence of the rule}}{\text{Support of result only}}$$

$$= \frac{P(\text{condition and result both occur})}{P(\text{condition only})\ P(\text{result only})}$$

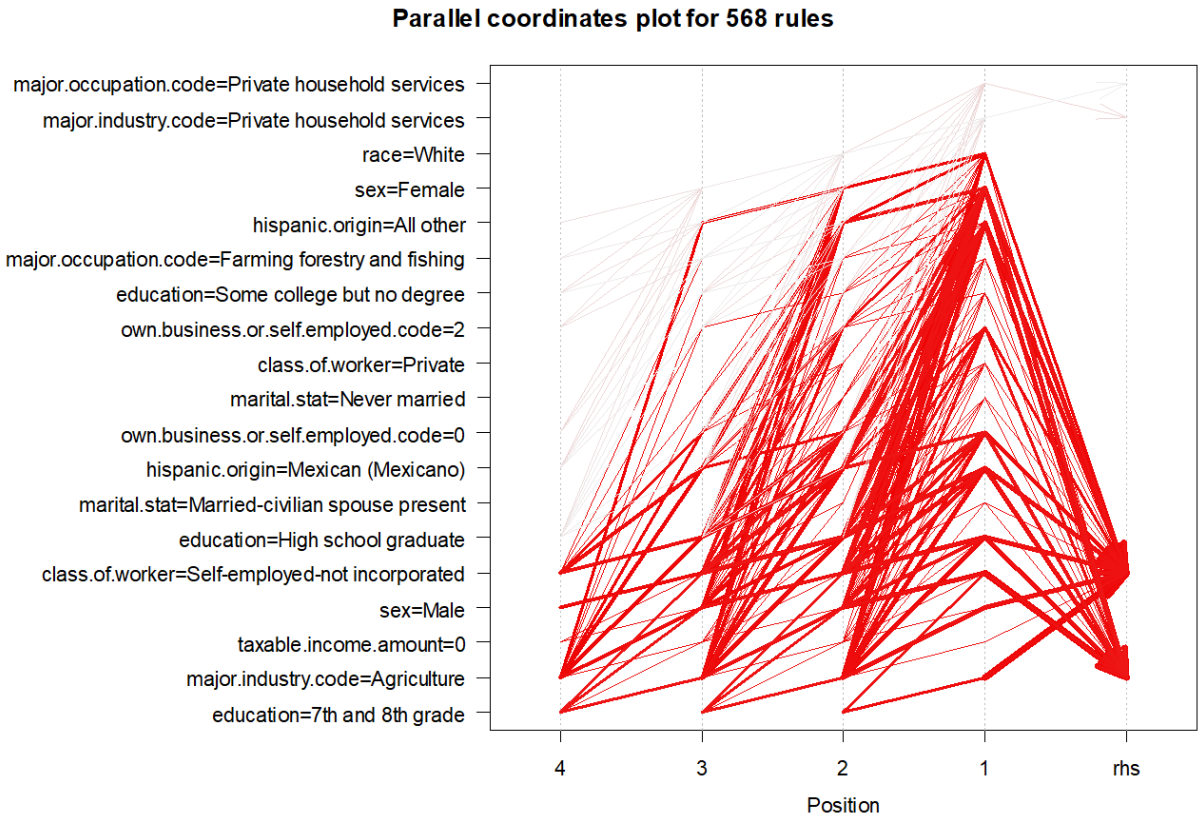
$$= \frac{\text{Support of the rule}}{(\text{Support of condition only})\ (\text{Support of result only})}$$

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

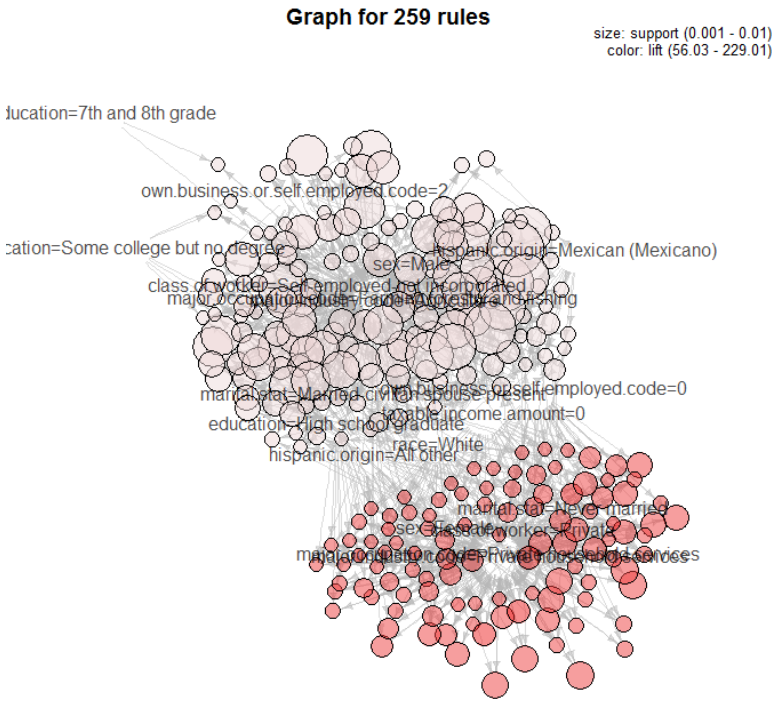
There can be hundreds of thousands of rules in a data set, so if you try to plot it, R will simply die on you.

Use this functionalities in two steps:

1. Identify the rules you want to work with
   - Set the confidence and support (it is a good idea to look for relatively high confidence, and low support, to avoid finding the the water is wet)
   - **Minimum support**
   - **Maximum Support**
   - **Minimum Confidence**
   - **Minimum Length of a rule:** how many items do we want on the "left side" of the relationship? Longer rules make analisis quite complex, small rules are usually very obvious
   - **Maximum strength of a rule:** same as above
   - **Minimum Lift:** minimum criteria to keep a rule. Alift of 1 means that there is no "gain of knowledge" with this rule, that is there is correlation, most likely not causality.
2. Once you have selected the rules that are interesting, you may want to graphically explore your data by selected the two chart options:

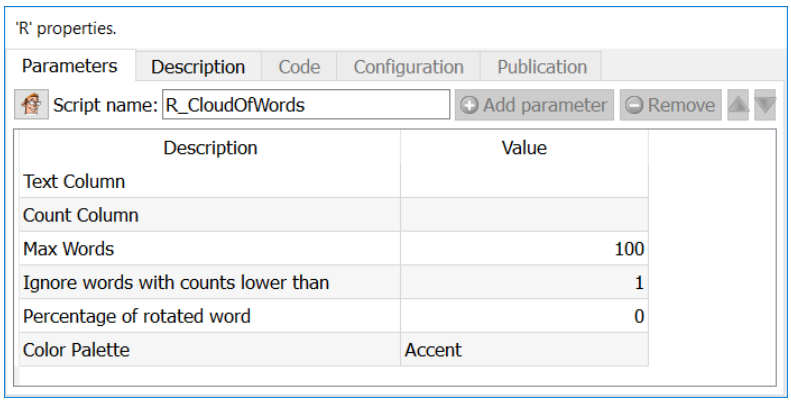


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



5.11.8. Cloud of Words (R action)

Icon: Cloud of Words



Property window:

Short description:
Generates a cloud of words based on text data

Long Description:
Generates a cloud of words based on text data.

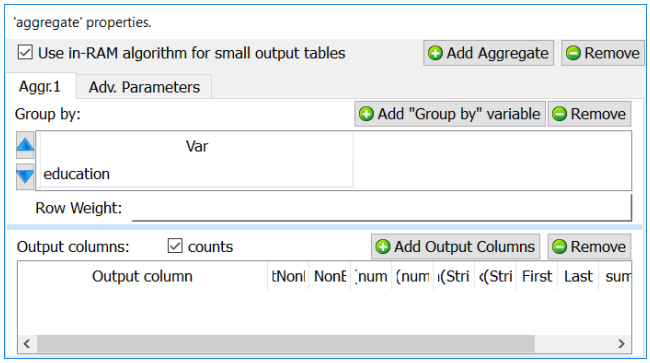
Text Column: the column with the word information
Count Column: the column with the frequency of occurrence of the word
Max Words: Maximim of different words to display
Ignore words with counts lower than: min threshold to display in the chart
Percentage of rotated words: percentage of words that will be displayed vertically
Color Palette: choose color scheme.

There are two ways of using this visualization:

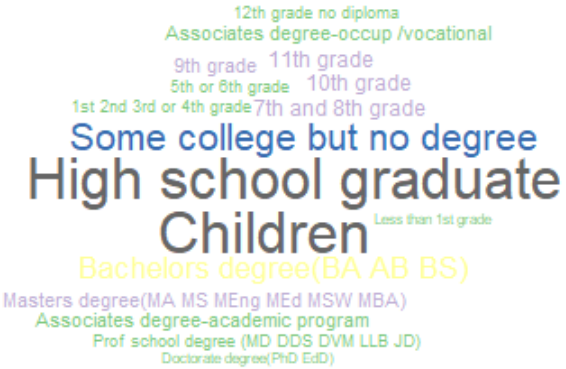
1. Generate account of occurrence of preset text}
2. Generate a count of occurrence of words in open text.

The first one is straight forward: simply use an aggregate and extract the count, for example, let's explore the education level in the Census Income database a bit differently. We simply use Aggregate on Education, and select the Count

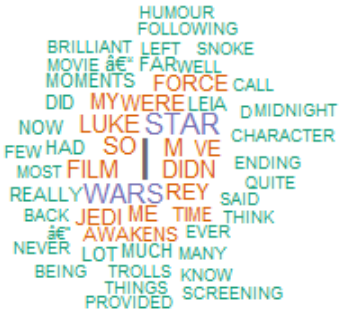
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



We then send the information to the Cloud of Word and we have:



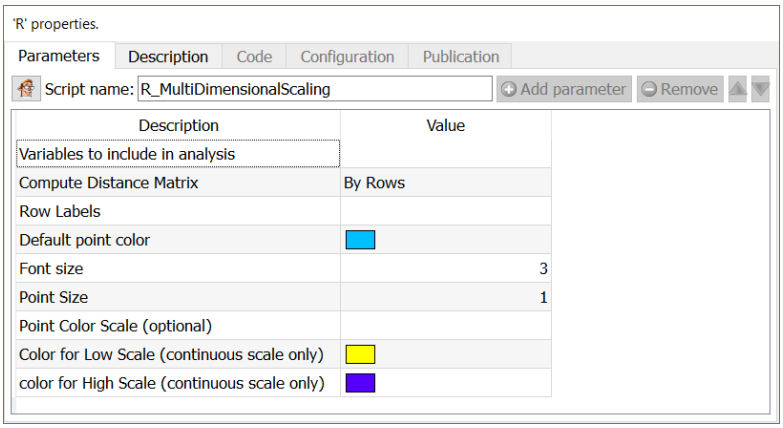
The second way require generating data for the cloud of word from a text. Let's explore a star wars review form December 21, 2018 (http://www.trustedreviews.com/opinion/star-wars-the-last-jedi-spoilers-ending-3356585):
First, we "normalize" our text data (set all to lower case, remove punctuation, remove accents), and we use the "Cloud of Word" action from "Text Mining", we then aggregate the data and send to the R Cloud of Word action:



5.11.9. Multi Dimensional Scaling (R action)

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

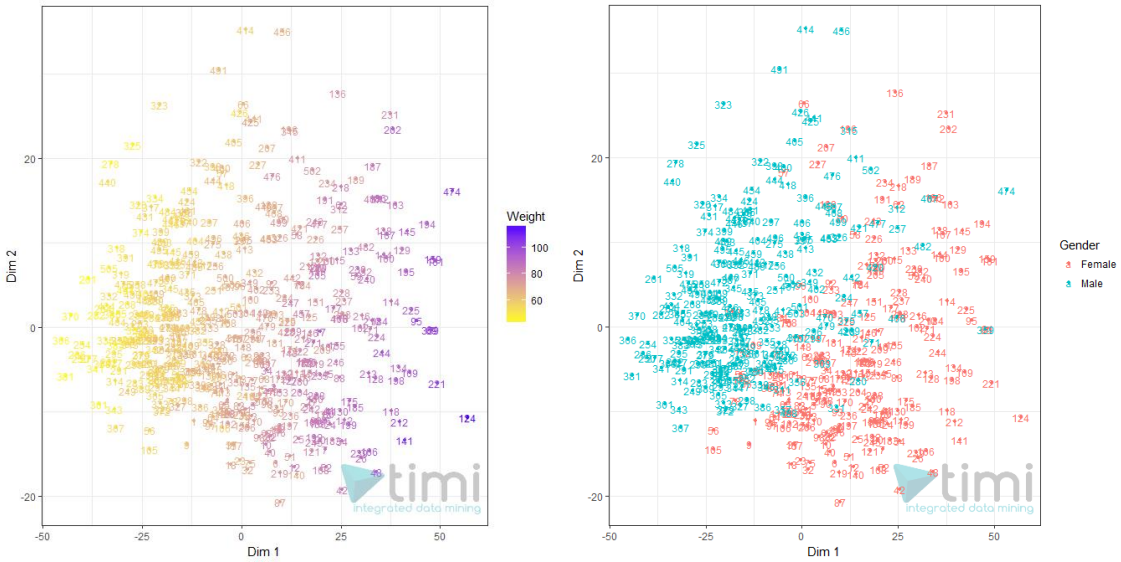
Short description:
MDS is a visual data reduction method used to better understand the proximity between records, or between variables.

Long Description:
MDS is an algorithm based on similarity (Euclidean distance). The algorithm seeks to represents in 2-3 dimensions the best approximation of distance between points.

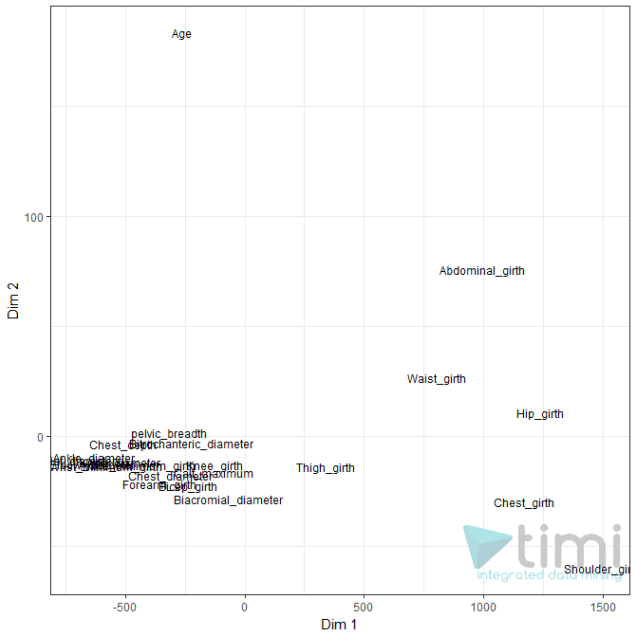
The implementation includes two modes: by row (similarity between observations) or by column (similarity between variables).

As this starts by generating a distance matrix, the algorithm quickly fills the memory when working with large database. It is, however, a good way to better understand correlation between columns (a bit like a PCA).

Variables to include in analysis: select the variables you wish to compute the distance on.
Compute Distance Matrix: Select whether you wish to study the distance between rows (observations) or columns (variables).
Row Labels: select the variable with the row lables (ID, etc.).
Default point color: select the color of the point on the chart (overruled if you used a scale variable).
Font Size: set the size of text displayed for point labels.
Point Size: set the size of points on the chart.
Point Color Scale: select a variable (nominal or continuous) to color the point and add additional information on the chart. Note that if the scale is NUMERICAL it MUST be set to float or key in Anatella, as R will consider a text vector of numbers as text (hence will treat it as categorical)
Color for Low Scale: if the color variable is a scale (ordinal or continuous), this will represent low values.
Color for High Scale: if the color variable is a scale (ordinal or continuous), this will represent High values.

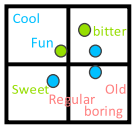


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

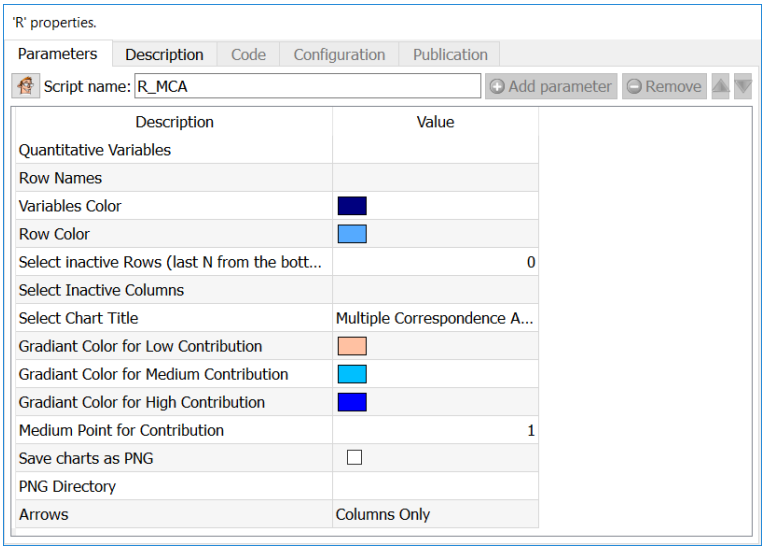


5.11.10. Correspondance Analysis (R action)

Icon:



Property window:



Short description:
Use a Chi-Square approximation to represent an aggregated matrix

Long Description:
Multiple Correspondance Analysis (MCA is a method often used in Market Research to understand the relationship between brands and attributes. It typically comes in the form:

	Brand 1	Brand 2	…	Brand K
Attribute 1				
Attribute 2				
…				
Attribute N				

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Where all the datapoints are average of the attribute for each brand, or frequency in which the attribute applies.

Multiple Correspondence Analysis is extensively discussed by Hoffman et al, and is a very popular technique in marketing research. Essentially, it is a multi dimensional scaling technique (MDS) that allows representation of multivariate data in a Euclidean space. Using this technique, one looks at the distance between objects to understand the characteristics of various brands on a market place.
While this method is often applied to plot proportions, it will also return valid results based in averages of data. Mapping analysis mostly relies on subjective interpretation of a chart, but the technique also offers metrics that allow us to estimate the validity of the map. The most important metric is the percentage of variance extracted (the amount of differences between brands and attribute correctly represented in the chart).

Scale of the variables is not that important but ***there MUST be a variance (if a row has identical values for all columns, MCA will fail)***

MCA is somewhat related to factor analysis but uses a Chi-Square approximation of the distance, which tends to misrepresent distances near the center of the chart, and often tends to merge two dimensions in one.

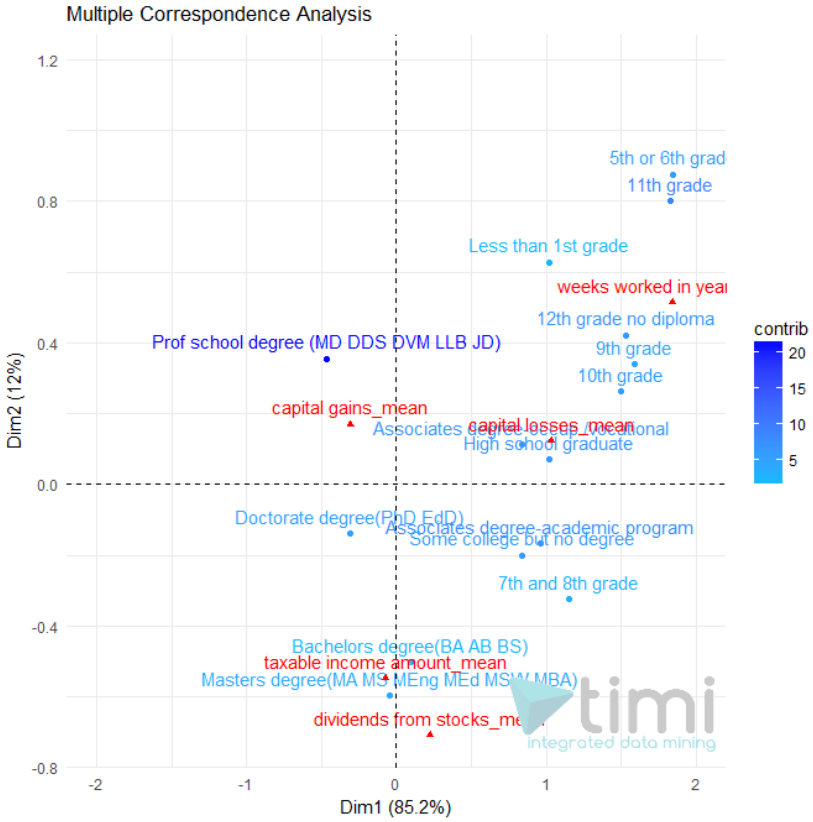
Quantitative Variables: select numerical variables to include in the plot
Row Names: Column with row labels
Variables Color: Set the color to plot column positions
Row Color: Set the color to plot row position
Select Inactive Rows: Select the number or rows from the BOTTOM that will be ignored in the computation, but still plotted
Select Inactive Columns: Select the columns that will be ignored in the computation, but still plotted
Set chart title: set the main label of the chart
Gradiant color for low contribution: set the color to represent points that are poorly represented
Gradiant color for medium contribution: set the color to represent points that are “OK” represented
Gradiant color for high contribution: set the color to represent points that are well represented
Medium point for contribution: set the contribution to set the “OK represented” value (default is 1)
Arrows: select if you want to put arrows on columns variables, row variables, or both
Map: The default plot of (M)CA is a "symmetric" plot in which both rows and columns are in principal coordinates. In this situation, it’s not possible to interpret the distance between row points and column points. To overcome this problem, the simplest way is to make an asymmetric plot. This means that, the column profiles must be presented in row space or vice-versa. The allowed options for the argument map are:

- "rowprincipal" or "colprincipal": asymmetric plots with either rows in principal coordinates and columns in standard coordinates, or vice versa. These plots preserve row metric or column metric respectively.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- "symbiplot": Both rows and columns are scaled to have variances equal to the singular values (square roots of eigenvalues), which gives a symmetric biplot but does not preserve row or column metrics.
- "rowgab" or "colgab": Asymmetric maps, proposed by Gabriel & Odoroff (1990), with rows (respectively, columns) in principal coordinates and columns (respectively, rows) in standard coordinates multiplied by the mass of the corresponding point.
- "rowgreen" or "colgreen": The so-called contribution biplots showing visually the most contributing points (Greenacre 2006b). These are similar to "rowgab" and "colgab" except that the points in standard coordinates are multiplied by the square root of the corresponding masses, giving reconstructions of the standardized residuals.

Intepretation

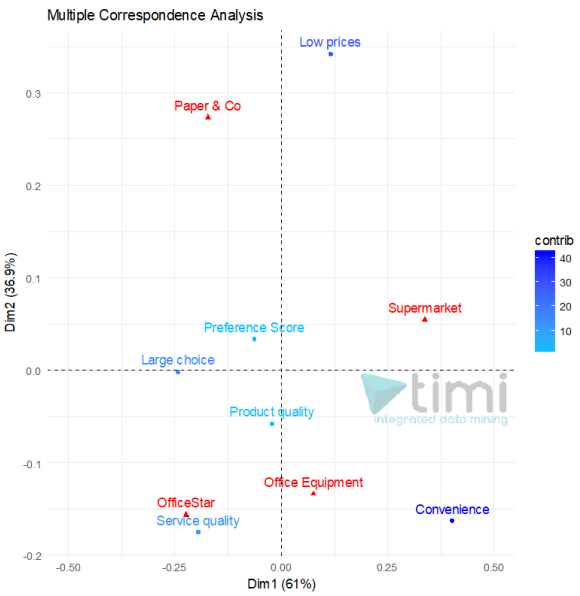


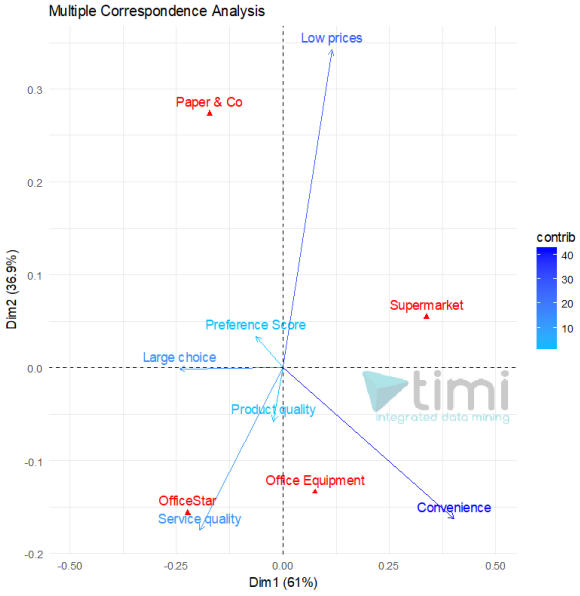
- **Taxable income Amount** is mostly explained by levels of education of Bachelor and Masters degree, and seems to occur more with high "dividend from stock".
- Those with a professional degree tend to have more capital gain, but less "taxable income".
- Those with the lowest education tend to work more weeks per year, and have the least chances to be rich.

Note that if you run a TIMi Model to predict "Taxable income amount", you will have a much better understanding of the dynamics, and some conclusions of this map will even turn out to be erroneous. Use interpretation *very* cautiously.

In the following example (courtesy of prof. A. De Bruyn. ESSEC), we see that OfficeStar is set apart by Service Quality, and convenience applies equally to OfficeEquipment and SuperMarket. Paper & Co is only set apart by Low Prices, where it is the only relevant brand.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

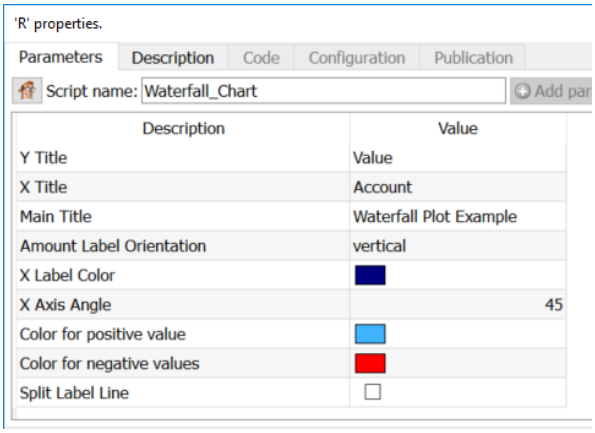




5.11.11. Waterfall Plot (action)

Icon:

Property window:



DataTable (21 rows - 2 columns) (comp

	C2	C1
1	Income 1	786.203
2	Income 2	632.865
3	Income 3	304.573
4	Income 4	275.863
5	Small Supply	-10.5837
6	Food	-13.3562
7	Events	-29.8032
8	Conferences	-43.1087
9	Local Tran...	-52.2597
10	Finance c...	-60.6157
11	Supplies	-65.215
12	Hotels	-121.835
13	Travel	-140.564
14	Marketing	-142.654
15	Raw Mate...	-160.124
16	Suppliers	-162.12
17	Contractors	-170.457
18	Salaries Low	-226.578
19	Salaries M...	-257.17
20	Salaries M...	-266.053
21	Commissi...	-293.54

Short description:
Create a waterfall Plot

Long Description:
The information used consists of two columns: category, and amount.

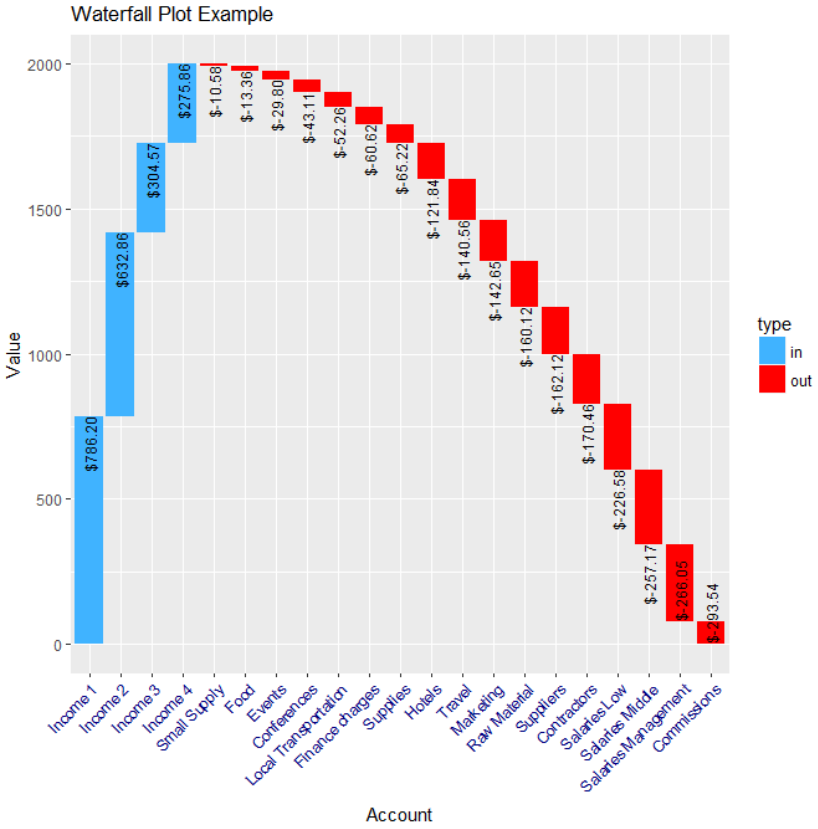
Typically, this chart is used to understand the cost structure of organizations, product categories or services.

The information must be SORTED on AMOUNT -> DECREASING, and it is a good idea to always do a bit of cleaning.

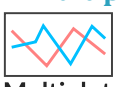
In the below example, we simply order the columns properly, clean the values to make sure everything is numerical, sort the data and plot the chart.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

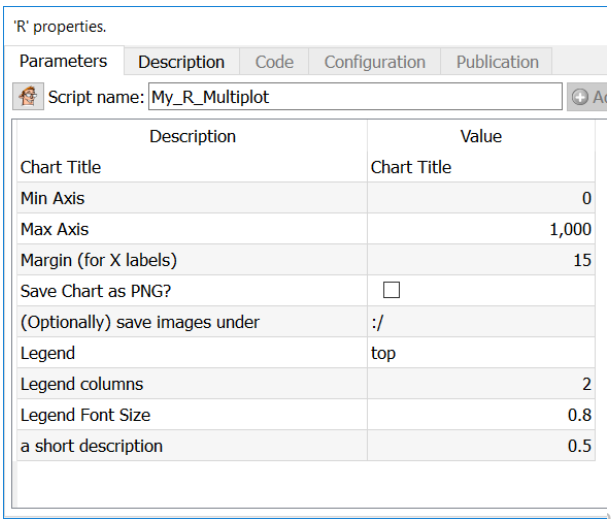
Y Title: the label of the Y Axis
X Title: the Label of the X Axis
Main title: the title of the chart
Amount Label Orientation: select how the labels are displayed in the waterfall plot
X Label Color: select the color of the labels on the X axis
Color for positive values: self explanatory
Color for negative values: self explanatory
Split Label Line: whether the labels on X include a carriage return between words, or is displayed in a single line



5.11.12. Multiplot (R action)

Icon:  Multiplot

Property window:



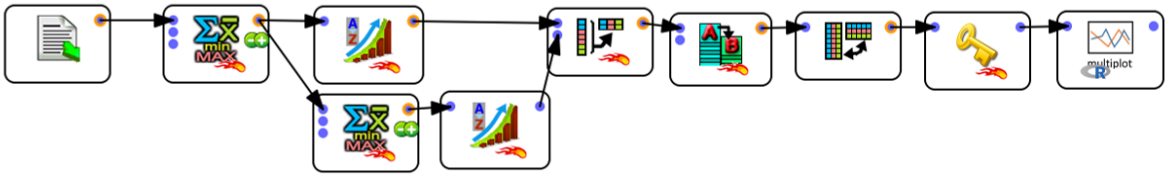
Short description:
Create a Plot with multiple series

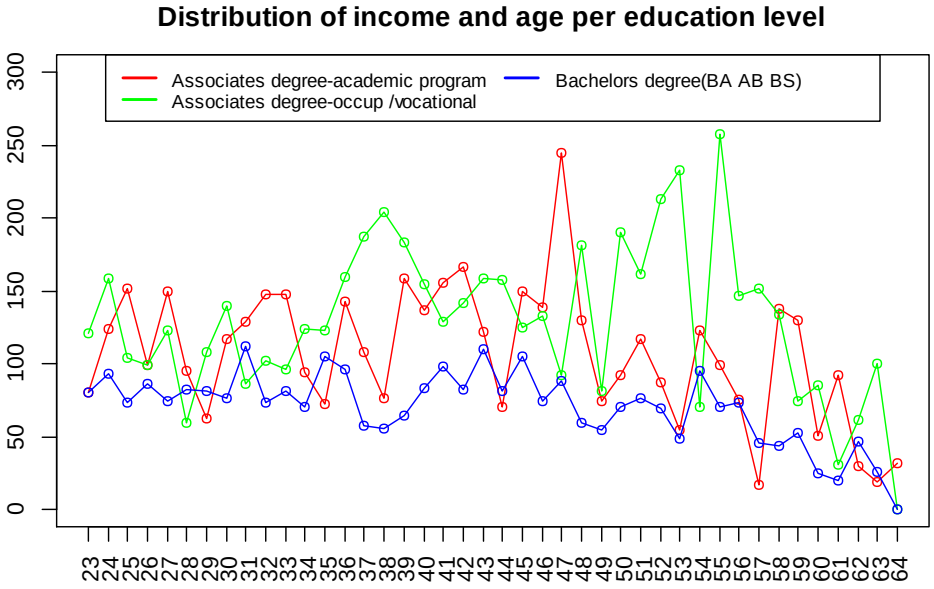
Long Description:
The information used consists of thre groups of columns:

- X axis
- Data
- Key

Main title: the title of the chart
Min Axis: the minimum value on the Y axis
Max Axis: the maximum value on the Y axis
Margin (for X labels): margin below the axis
Legend: set legend position
Legend Columns: number of columns to display the legend text
Legend font Size: size of legend characters
Save chart as PNG: select how the labels are displayed in the waterfall plot
PNG Directory: select the color of the labels on the X axis

To obtain this data, we usually require a few transformations, as illustrated below. The data must be organized as one series per column, hence a transposition.



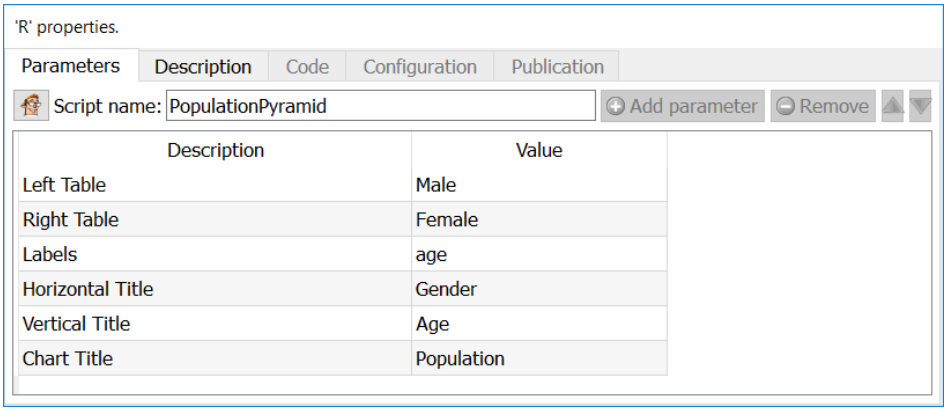


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.11.13. Population Pyramid Plot (R action)

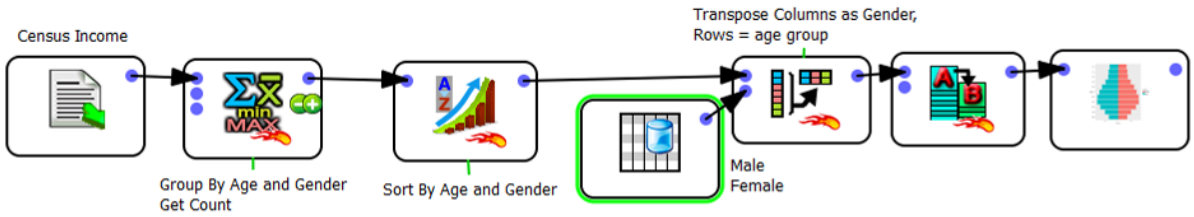
Icon:

Property window:



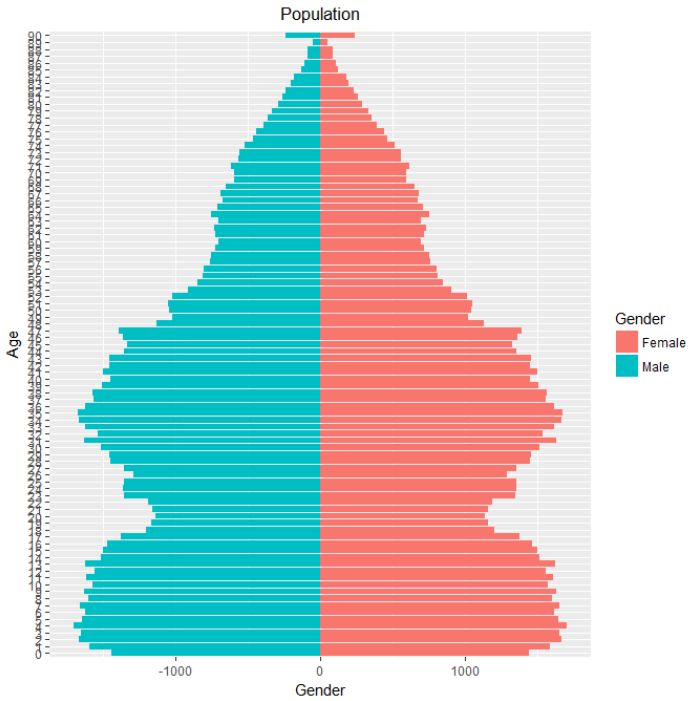
Short description:
Create a population pyramid by gender

Long Description:
Select columns in which aggregated information is available. To process vast amount of information, this plot requires the data to be pre-processed in Anatella: a population of 5.000.000 individual will quickly saturate the RAM in R, which is why the following pre-processing is required



Parameters:

Left Table: the variable to display on the left (***one Column***)
Right Table: the variable to display on the right (***one Column***)
Labels: the labels of the vertical axis (age, ***one Column***)
Horizontal Title: the title of the base (usually Gender, ***text***)
Vertical Title: the title of the vertical axis (usually Age, text)
Chart Title: the title of the plot (***text***)

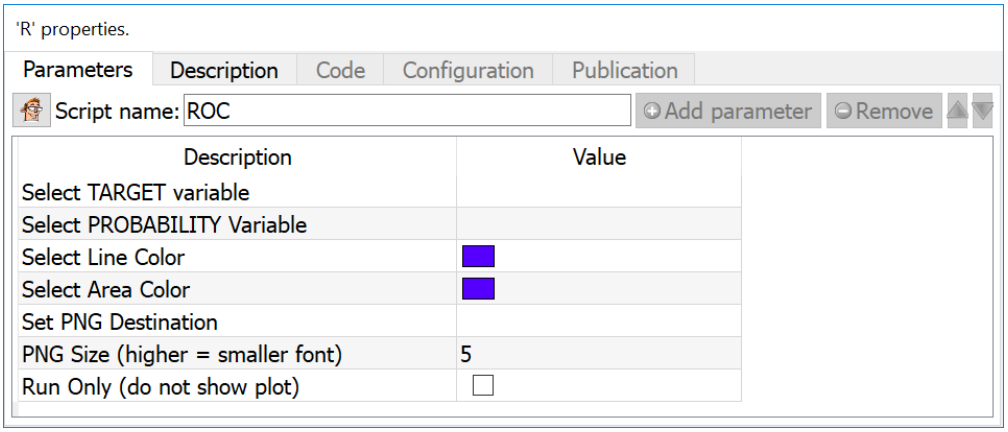


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.11.14. ROC Curve (R action)

Icon:

Property window:



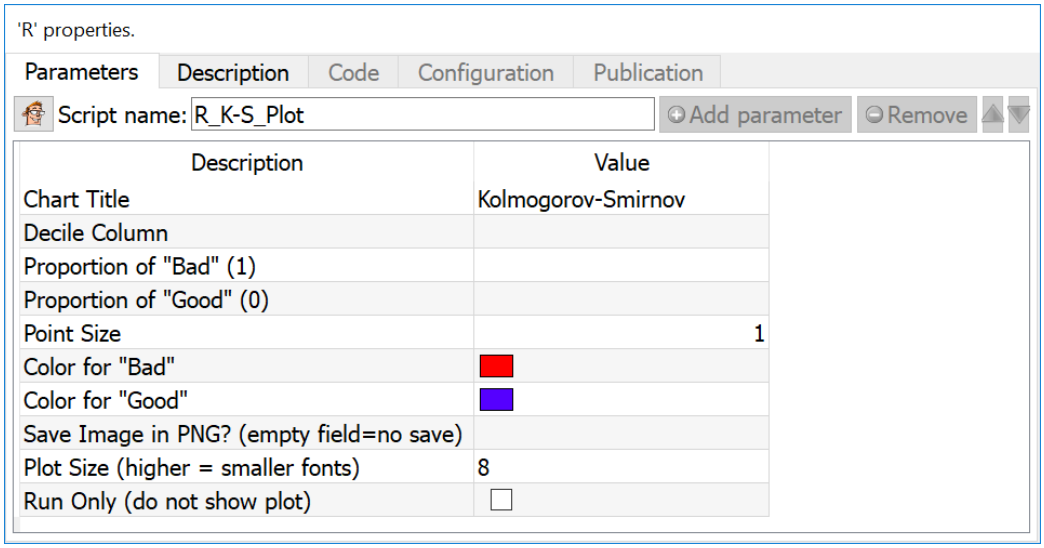
Short description:
Create a ROC curve after applying a model

Long Description:
Create a ROC curve after applying a model. Note that it is a good idea to round your score to 2 decimals prior to plotting the ROC curve if you are working with XGBoost or ElasticNet, as those algororithms tend to generate finer probability scores.

5.11.15. KS Curve (R action)

Icon:

Property window:



Short description:
Create a Kolmogorov-Smirnov curve after applying a model

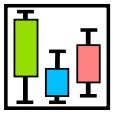
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Create a Kolmogorov-Smirnov curve after applying a model. The input data consists of an **aggregated table** (usually deciles) in which we have computed:

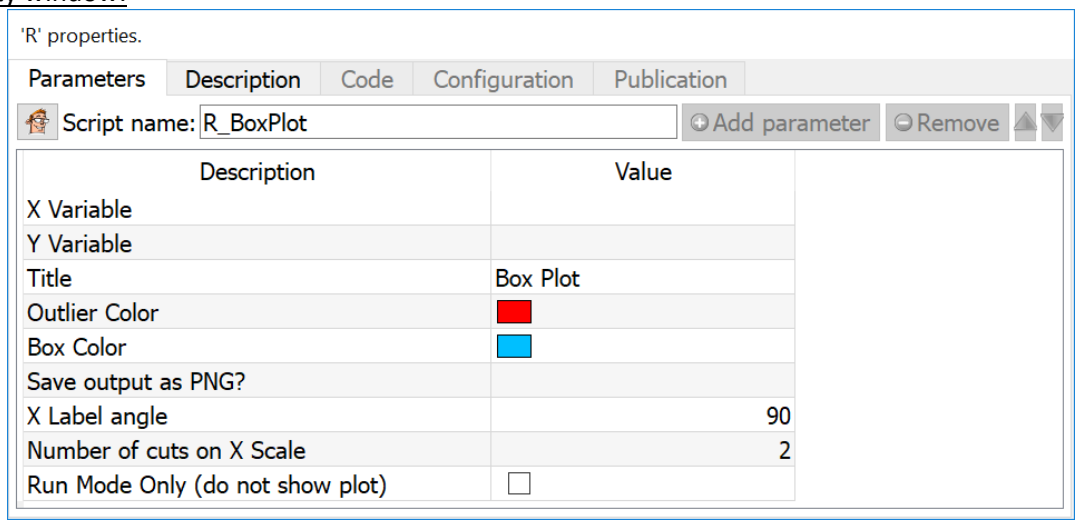
- Proportion of "bad" in each decile (usually the target)
- Proportion of "good" in each decile
- The decile

This curve is often used in credit risk analysis, but is – generally speaking – not a very robust way to evaluate a model as it assumes a normal distribution of the scores which is not appropriate if you work with actual transactional data. Some authors suggest that a logartimic transformation can improve it, but this would only be appropriate if using a logistic regression (which depends on transformed scores, and in general WoE), but not with any ML algortithm that do not use this transformation.

5.11.16. Box Plot (R action)

Icon: 

Property window:



Short description:
Create a box plot

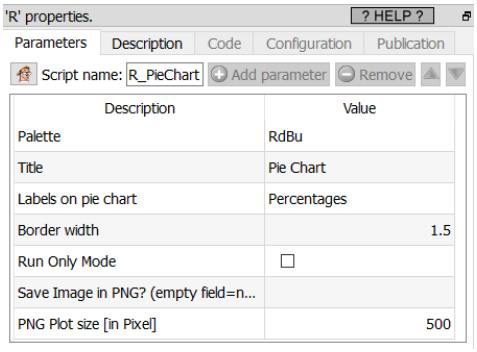
Long Description:
Create a box plot. This chart is useful to compare bi-variate distributions in which the X axis is a nominal variable (age group, product line, eduction, etc) and the Y axis is a continuous variable (price, discount, income, etc.).

Partition Type: To analyze many groups in a single operation, you can split the data based on a partition (region, race, zone, etc.) The data needs to be sorted based on this variable. It will automatically generate k plots
Var: X Variable: Select the nominal variable to split in the horizontal axis
Var Y: Select the continuous variable to observe.
Optional Target: Select a third variable to set colors based on density

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Run Mode Only: Do not display plot. This is particularly useful when working with partitions, as there is a limit to 50 plots "visible". This option can generate hundreds or thousands of plots without filling your RAM.

5.11.17. Pie Chart (R action)



Icon:

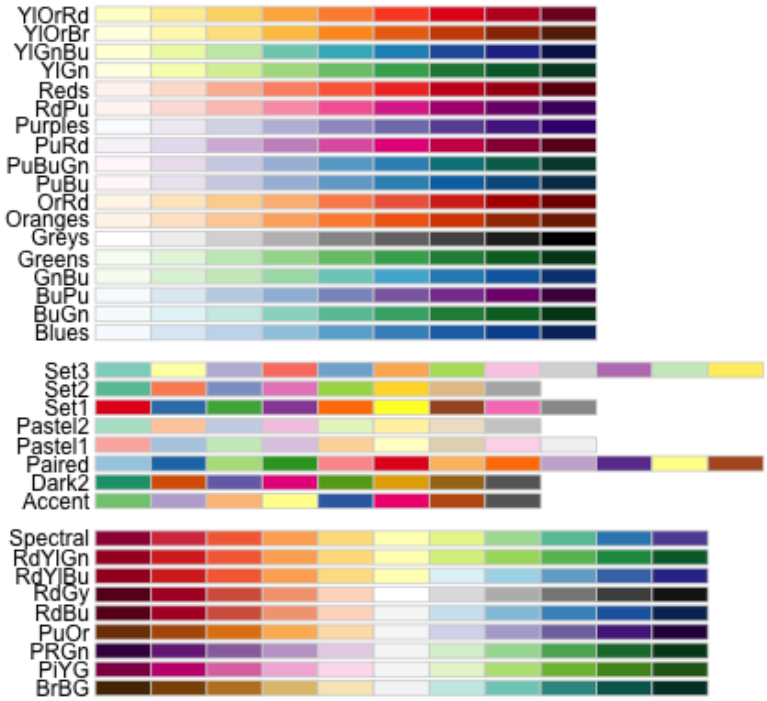
Property window:

Short description:
Create a Pie Chart

Long Description:
This plot is a popular visualization, although it should not be used in most cases as a histogram typically gives a better visualization.}
The input table is:

- first column: the different labels of the different pies.
- second column: the sizes of the different pies.

The input table can easily be created using an Aggregate Action.
The different available palettes are:



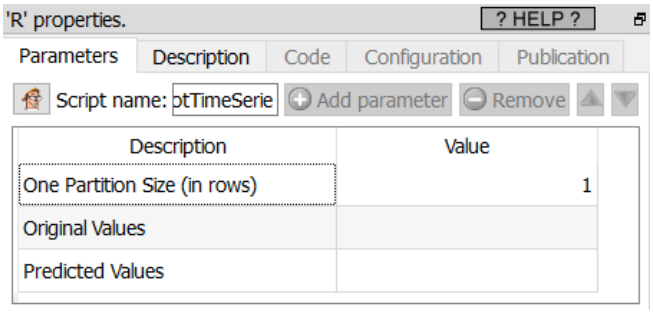
Labels of the different Pies: Select the variable with the label of the category
Size of the different Pies: select the variable with the size of the category
Labels on Pie Chart: Run Only Mode: Select "None", "Value" or "Percentage"
Run Only Mode: select this option to generate a PNG image of the plot without displaying it. If you forget to specify a filename, this will generate an error.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.11.18. Plot Time Series (R action)

Icon:

Property window:

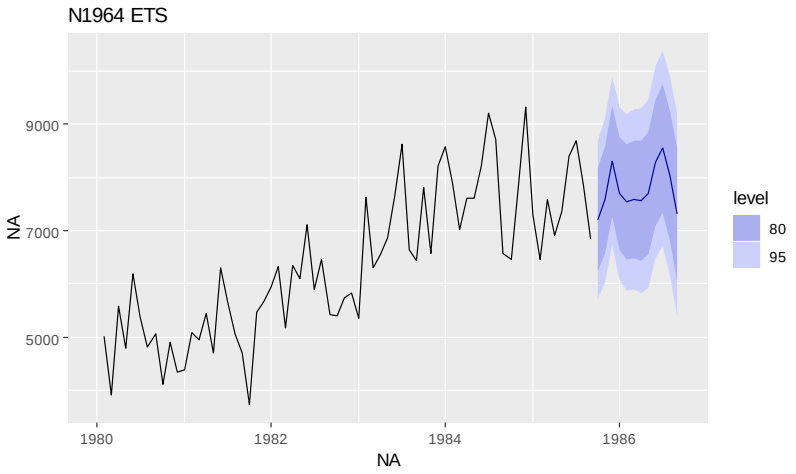


Short description:
Plot a time series

Long Description:
Plot two corresponding series:

- Past data
- Future data

The data must be stored in COLUMNS, and will generate one plot per row.
An example:

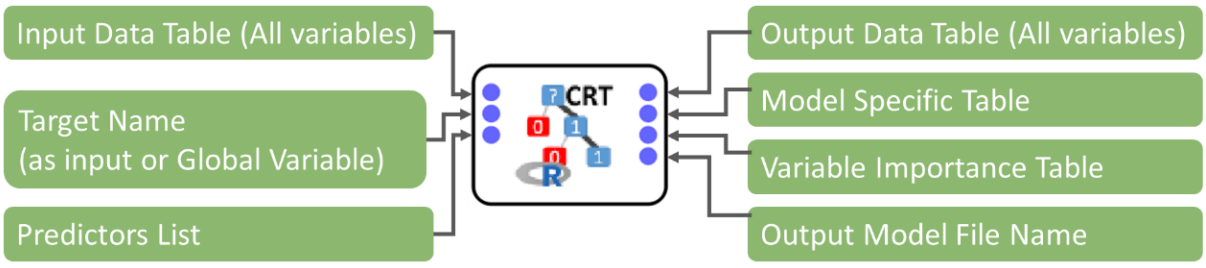


5.12. TA – R Predictive (TA=Transformations Actions)

All predictive Analytics actions have the same basic structure:

- Set Target Variables (dependent variable)
- Select Predictors variables (independent variables)
- Set options

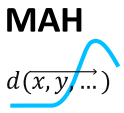
This information can be set manually by selecting variables, or by using an automated process.



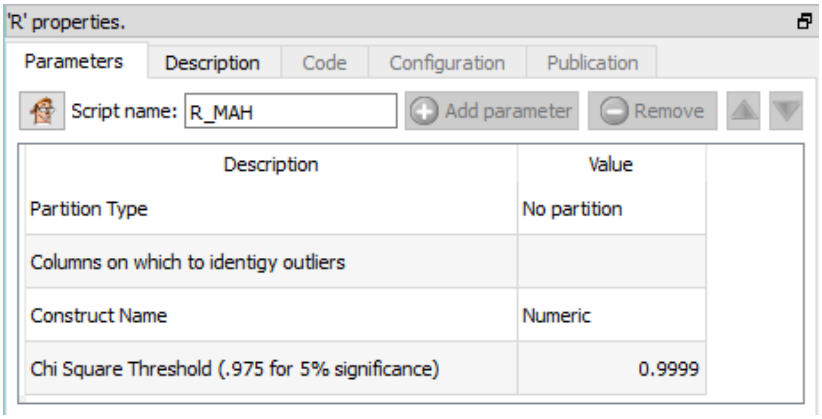
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Each node is explained in details in the following section. As the entire matrix will be sent to the R engine, it is a good idea to make a preselection of the relevant variables if you have many thousands of dimensions.

5.12.1. Multivariate Outlier Detection (R action)

Icon: 

Property window:



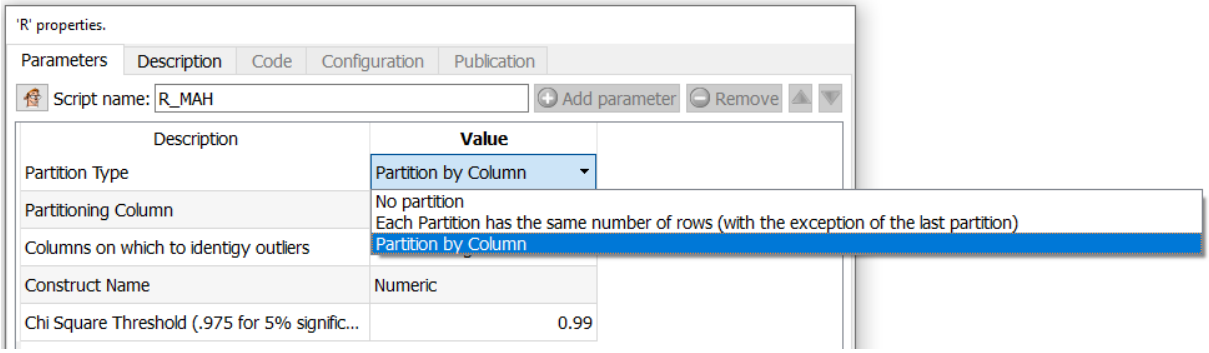
Short description:
Identifies outliers in a dataset.

Long Description:
Identifies outliers in a dataset using the Mahalanobis distance projected on a Chi-squared distribution.

The Mahalanobis distance is an absolute number starting at 0 at the center of the multivariate distribution, and the distance is weighted by the covariance matrix in order to include the density into the equation. The largest the distance, the most likely a point is an outlier.

The Chi-Squared test gives a statistical threshold to flag outliers. On sample, the value is typically 0.9999 (we reject outliers if they have less than 0.01% probability of belonging to the multivariate distribution).

Special Note on Partition:
In many situations, outliers are not that meaningful on the total population. For example, some price levels are only outliers if take per product categories, consumption is often region specific, etc. To perform such analysis, simple set a partition to your variable of interest, and Mahalanobis distance will be computed in a single operation on all different categories. Note that the **data must be sorted by the partition variable** before.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.12.2. CART – Step 1 - Create one "deep" tree (R action)

Icon:

Property window:

'R' properties.

Parameters | Description | Code | Configuration | Publication

Script name: My_R_CART_Full | Add parameter | Remove

Description	Value
predictors	age, class of worker, detailed ind...
Target	taxable income amount
Weight of the "1" category (adjust for unb...	TreeWeight
job to do	Classification (Nominal)
model name	:/CRT_R_Model_Full_weight.rMo...
Plot Scree	☑
Plot Tree (If not, the model is still generat...	☐
Plot Scree and Tree charts in a single wind...	☐
Tree Display	Beautiful
Seed	0
Maximum Depth of the tree	20
Number of Cross Validation	5
Complexity Parameter (entropy criteria to ...	0.0001
Minimum size of a node to allow a split	200
Minimum Size of a Terminal Node	100
Use surrogates in Model	Majority Rule (Recommended)
Maximum Competing splits	3
Maximum surrogate	3

Short description:
Create a tree using the CART algorithm

Long Description:
The CRT node uses Recursive Partitioning and Regresion Tree as implemented in the R rpart library, based on the 1984 book of Breiman, Friedman, Olshen and Stone. The library is the work of Terry Therneau, Beth Akinson, and Brian Ripley.

A complete introduction to the technique can be found in:
https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

This generates a CART model, and send the following output tables:
Output pin 0 : full table with prediction
Output pin 1: error table for selection of pruning parameters
Output pin 2: variable importances
Output pin 3: Saved Model Name

The goal of this analysis is to generate a large tree and identify where to prune it (see next action node). While this algorithm has gained popularity in the early stage of data mining, it is now increasingly being left unused.

The main advantage of CART is it ability to combine nominal and continuous variable to predict (typically) nominal groups. Being non-parametric, it adapts to fairly large datasets, and does not suffer from multicollinearity or non-normality of the data. When it was introduced, it was also highlighted that it yields better predictions than CHAID. Another key advantage is that it is intuitive, and offers a methodology similar to a drill down, but with some validation.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The key problems of CRT are that as trees deepen, we are almost certain to overfit the data (some argue that trees below three levels should never be used, other go up to 6 or 7 levels). Indeed, as we see a succession of conditional probabilities, whatever comes out of the model becomes irrealistic, which is why many use combinations of hundreds of small trees (XGBoost, TreeNet, Forest of Stumps, etc.) to overcome this problem. This yields better models, but they are "black boxes".

In practice, the first cut is often based on particularities of the sample, and the following cuts are dependent on the first one, making such models unstable for predictive purposes. They are, however, useful in terms of data exploration and understanding.

Parameters:

The CRT tree (Classification and Regression Tree) requires the following settings:

Predictors: list of independent variables we want to use to predict in the form of $x_1+x_2+...+x_k$

Target: the variable we want to predict (binary, continuous, or multinomial)

Weight: the tree needs to be "balanced", so it is a good idea to create a weight variable. If all categories are equal, this variable is a constant. Usually, use (1/apriori) for records with a target 1, and 1 for records with 0.

Job to do: classification or regression

Model Name: name of the model file. This is required to add pruning, or apply the model as is.

Show Plots: unselect this option to avoid showing any plots (run model without visual information)

Plot Scree and autoPrune: select whether to plot the Scree Plot to identify where overfitting begins. You can also set an automatic pruning based on minimum error or "elbow" method. Elbow will yield smaller, often more stable trees.

Plot Tree: select whether to plot the treee

Plot Scree and Tree chars in a single window: self explanatory

Tree display: simplified or fancy. The simplified one is *much* faster to display, but will not show text categories properly.

Advanced Parameters:

Seed: set a seed value to be able to change the starting point, or reproduce a previous model

Maximum depth of the tree: set how many successive splits are allowed. Note that very deep trees are unstable (overfitting). Typically, more than 10 is not worth exploring. More

Minimum size to allow a split: crt weill not cut nodes than have less observations

Minimum size of a terminal node: when splitting, CRT will attempt to respect this size. If splitting a node leads to a terminal node smaller than this criterion, the split will not happen

Use surrogates in model: this option allows you to control what happens in case of missing value.

- Display Only: an observation with a missing value for the primary split rule is not sent further down the tree.
- NO Split for all missing: means use surrogates, in order, to split subjects missing the primary
- variable; if all surrogates are missing the observation is not split.
- Majority Rule (recommended): if all surrogates are missing, then send the observation in the majority direction. This corresponds to the recommendations of Breiman et.al (1984).

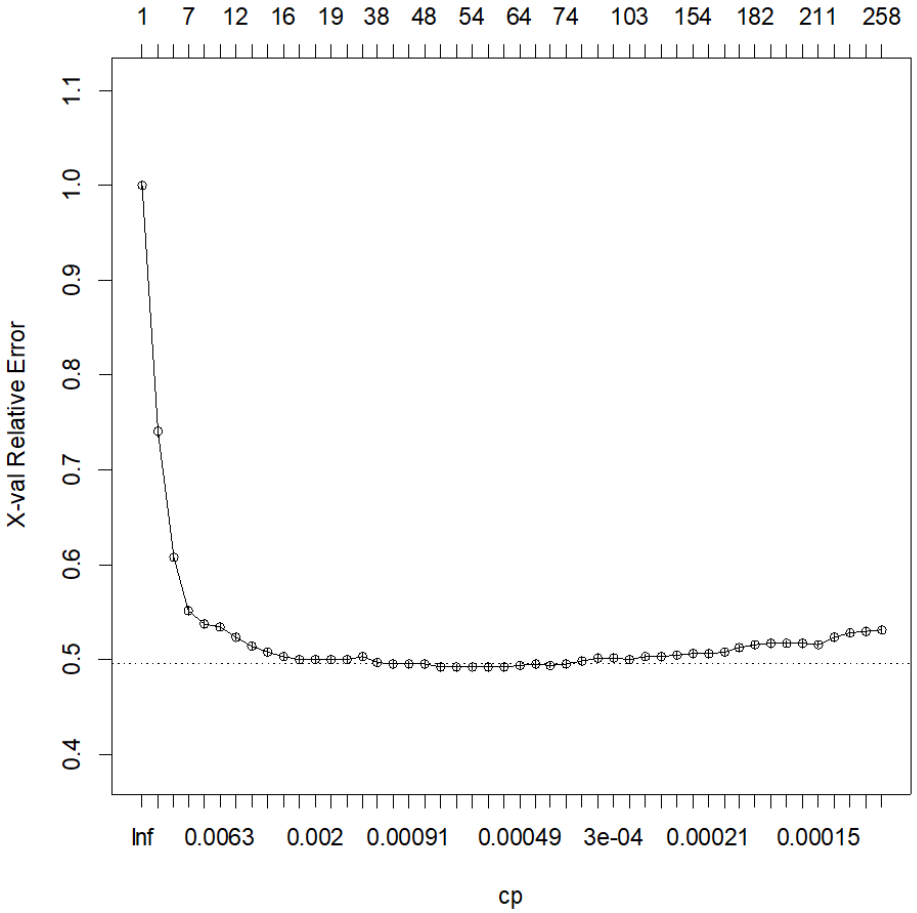
Maximum Competing Splits: the number of competitor splits retained in the output. It is useful to know not just which split was chosen, but which variable came in second, third, etc

Maximum Surrogate: the number of surrogate splits retained in the output. If this is set to zero, the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

compute time will be reduced, since approximately half of the computational time (other than setup) is used in the search for surrogate splits. However, this will yield to a model harder to put into production.

In the log, you will see the details of the analysis. The Scree plot shows the shape of the error reduction (and gives you a pruning criterion):

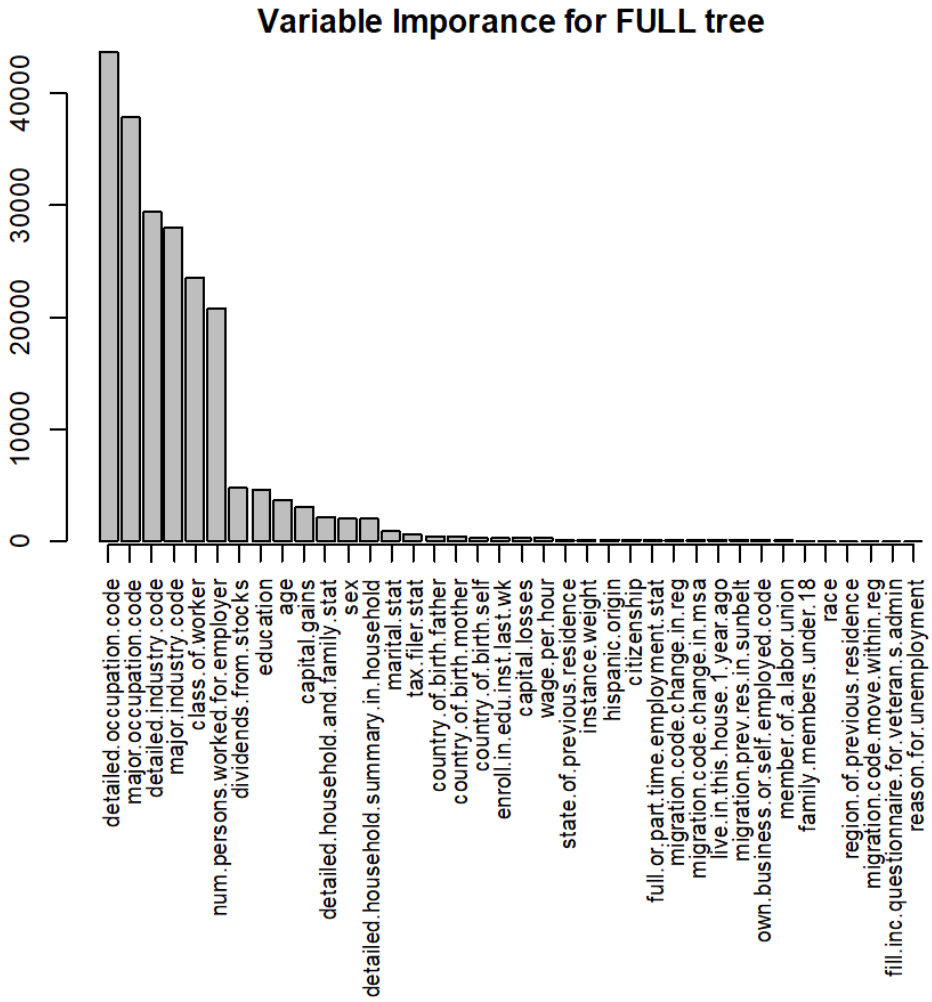


Once we observe a flat line, there is no improvement on the relative error, hence the tree is most likely overfitting by creating additional cuts. In the log, we can see details of this metric, where we see that two "good" stopping criteria could be 18, where it seems to stabilize, or 47 splits, based on the minimum error:

	CP	nsplit	rel error	xerror	xstd
1	0.207568	0	1	1	0.00527
2	0.191278	1	0.79243	0.79243	0.005003
3	0.018632	2	0.60115	0.60771	0.004611
4	0.011306	6	0.52663	0.53087	0.004396
5	0.009137	7	0.51532	0.52335	0.004373
6	0.005584	8	0.50618	0.51878	0.004359
7	0.002423	9	0.5006	0.50392	0.004312
8	0.00223	11	0.49575	0.50032	0.0043
9	0.002123	17	0.48237	0.49931	0.004297
10	**0.00203**	**18**	**0.48025**	**0.49931**	**0.004297**
11	0.001384	19	0.47822	0.49557	0.004285
12	0.001206	22	0.47407	0.49142	0.004271
13	0.001015	33	0.45828	0.48874	0.004262
14	0.000969	34	0.45727	0.48869	0.004262
15	0.0009	35	0.4563	0.48929	0.004264
16	0.000715	37	0.4545	0.48565	0.004252
17	0.000692	41	0.45159	0.48325	0.004244
18	**0.000554**	**47**	**0.44712**	**0.48205**	**0.00424**
19	0.000538	48	0.44656	0.48463	0.004249
20	0.000531	51	0.44495	0.48708	0.004257

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

A chart with variable importance is also returned:



You will also find this information in text format in the Anatella log-window.

5.12.3. CART – Step 2- Prune a "deep" tree (action)

Icon:

Property window:

'R' properties.

Parameters | Description | Code | Configuration | Publication

Script name: R_CART_Prune | Add parameter | Remove

Description	Value
Number of Split	15
Full CART Model (no pruning)	
File output for the pruned model	
Tree Display	Beautiful
Type	Separate Split Labels AND Lab...
Add shadows (Much Slower)	☐
Add Leaves on the Bottom	☐
Font size (0=automatic)	0
Split Font size	1.5

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

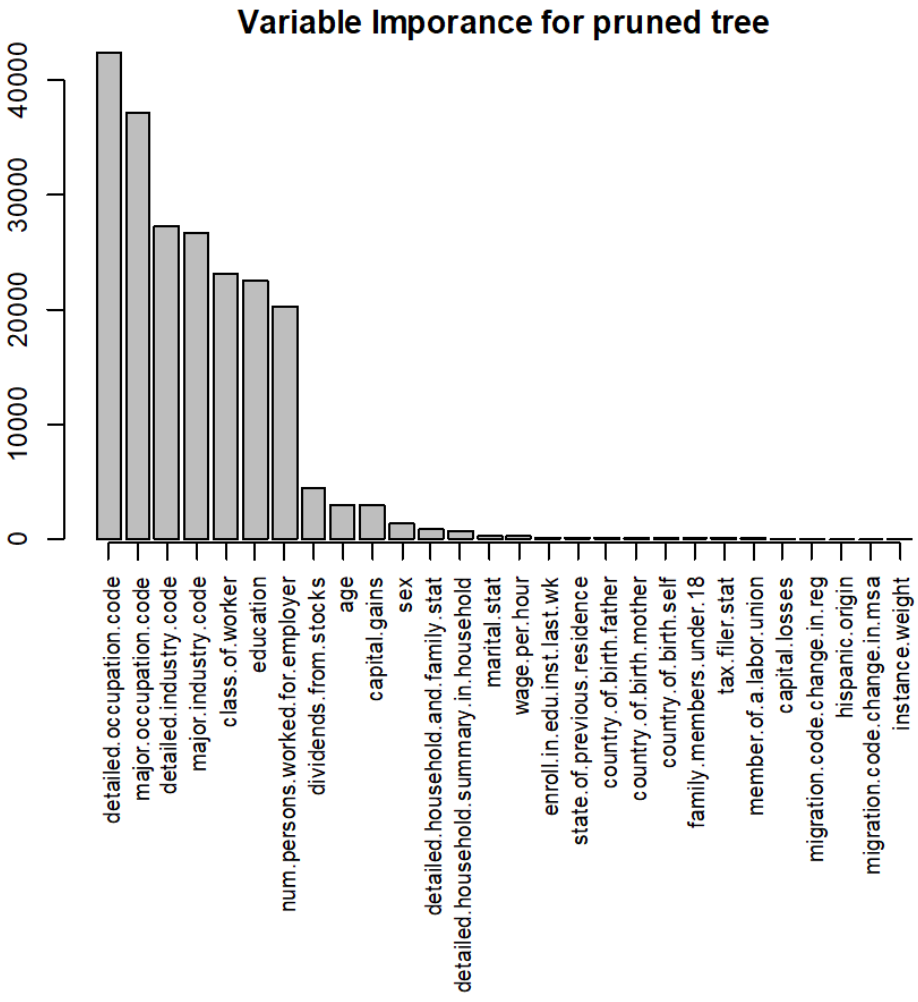
Short description:
Prune a CART model. This has been deprecated since there is now the option of managing this from the main node, and from the intereactive CART

Long Description:
This prunes a CART model, and send the following output:
pin 0 : full table with data
pin 1: variable importance in the model

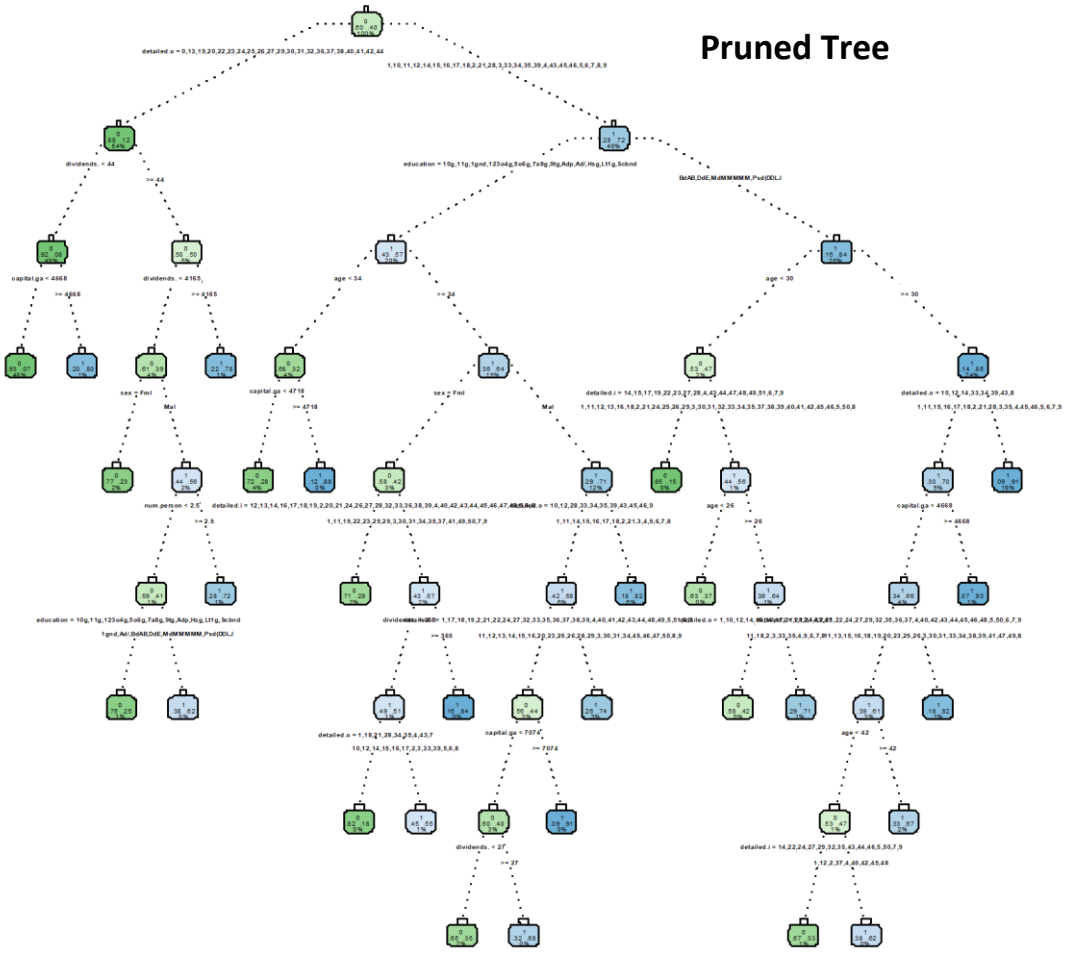
Parameters:

Number of Split: enter the number based on the previous analysis.
Full CART Model: select the file in which you saved the full model.
File output for the pruned model: Select the file where you want to save your model for future scoring.
Tree Display: simplified or Beautiful.
Type: select one of the four chart types.
Add shadows: add a shadow effect below the leaves. This is incredibly slow to display.
Add Leaves on the bottom: add all the terminal nodes on a single line in the bottom. This is very cluttered.
Font Size: Set the font size for the tree (0 is automatic).
Split font size: Set the relative font size for the split leaves.

This action displays the pruned tree and the variable importance. In the log, you will find a summary of the model, detailing all surrogate rules.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



5.12.4. Neural Network & MultiNomial Logistic Prediction (R action)

Icon:

Property window:

'R' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: R_NNET | Add parameter | Remove

Description	Value
List of Predictors	
Target	
Model Output	
Export Model to PMML	☐
PNG Directory (set to save files)	
Select Classification Model	Neural Network
Base (text label of base category)	1
Random Seed	42
Number of perceptrons - not for MNL	10
Maximum Iterations	200
Show Plots	☐
Linear Model (default is logistic) - not for MNL	☑
Include Prediction in First Output	☑
Normalize Predictors	☑

Short description:
Compute a Neural Network Model or a Multi Nomial Logistic Model

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Neural Networks are popular in datamining and analytics, mainly thanks to some of its promoters (i.e. Google) and their increased popularity in image pattern recognition. There now exists a new type of Neural Network algorithms (that is named “deep neural network”) that seems to perform reasonably well on image classification and segmentation tasks.

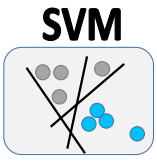
The

action described in this section is not a “deep” neural network: it’s an “old-school” Neural Network algorithm and it’s included in Anatella mainly because of completeness (and for explanatory/teaching purposes). More precisely, “old-school” Neural Network algorithms are usually not very useful because they are notoriously difficult to adjust properly to get a correct classification accuracy (although it’s sometime possible to get good results, it’s quite difficult).

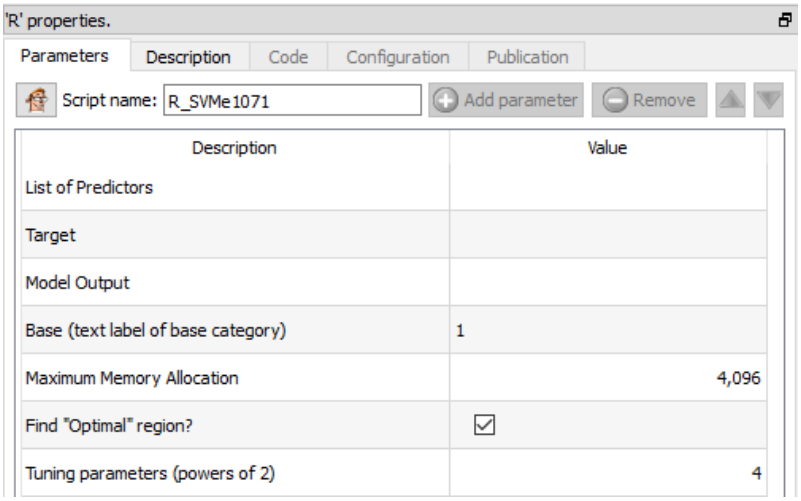
Parameters:

List of Predictors: Select independent variables
Target: Select the variable you want to predict
Model Output: Set the file name for the model results
Export to PMML: explort to a PMML file to include in other tools.
Select Classification Model: either Multinomial Logit or Neural Network
Base: set the base category
Number of perceptrons for Neural Networks: Manually set the perceptrons. This implementation uses only one layer
Maximim Iteerations: how long are you willing to wait for results. 200 is a good number.
Linear Model: Specify if the perceptrons should use linear models instead of logistics. This works only for NNET and allows estimating continuous targets, often with higher precision than simple linear models.
Show Plots: choose to show or not the visuals of the nnet or MNL model
Include Prediction in first Output: fit the model so you can better assess its quality
Normalize Predictors: make sure all variables are normalized. This usually yields better results.

5.12.5. Support Vector Machine (action)

Icon: 

Property window:



Short description:
Create a SVM model

Long Description:
Create a predictive model based on “Support Vector Machine” (SVM).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

SVM models are interesting from a purely theoretical point-of-view because they have mathematical properties that allows researcher to write many scholary articles about them.

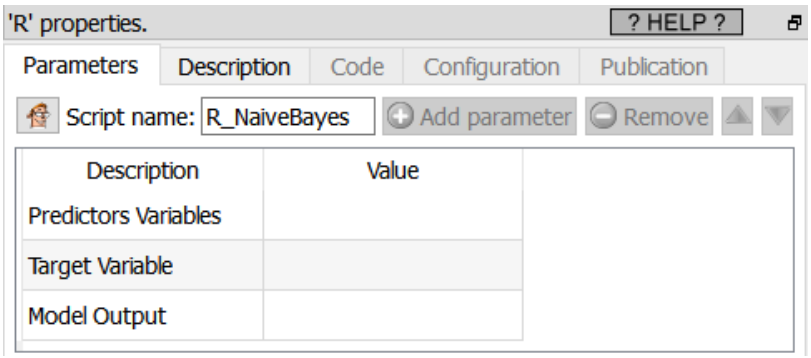
The SVM algorithm is included in Anatella mainly because of historical reasons (and for explanatory/teaching purposes). Indeed, from a pratical point-of-view, SVM are not very useful anymore because:

- The SVM algorithm do not scale well: it crashes (typically because of RAM memory issues) on most common-size datasets
- The SVM algorithm is extremely Slow (several orders of magnitude slower than any other algorithm)
- Compared to other modeling algorithms, the SVM algorithm is not very accurate (other algorithms have typically higher AUC, and higher accuracy). This fact is visible in all datamining competitions (KDD cups, Kaggle) where SVM always ranks amongst the worst algorithms.

5.12.6. Naïve Bayes (R action)

Icon:

Property window:



Short description:
Create a Naïve Bayes Model

Long Description:
Naïve Bayes simply "smoothens" the dataset based on the conditional probabilities, basically computes the average probability of an event given a set of characteristics.

The Naïve Bayes algorithm is included in Anatella mainly because of historical reasons (and for explanatory/teaching purposes). Indeed, from a pratical point-of-view, the "Naïve Bayes" algorhtm is not very useful anymore because:

- It's impractical to use because the "Naïve Bayes" algorithm assumes that all your predictor variables are 100% un-correlated, which never happens in practice. Thus, before using the

Naïve Bayes Action, you should first compute the Correlation Matrix (using the
Covariance Action from section 5.7.11) to decide which variables you'll keep inside your dataset (you must only keep un-correlated columns). This "cleaning" procedure is difficult and very time-consumming.
- Compared to other modeling algorithms, the "Naïve Bayes" algorithm is not very accurate (other algorithms have typically higher AUC, and higher accuracy). This fact is visible in all datamining competitions (KDD cups, Kaggle) where the "Naïve Bayes" algorithm always ranks amongst the worst algorithms.

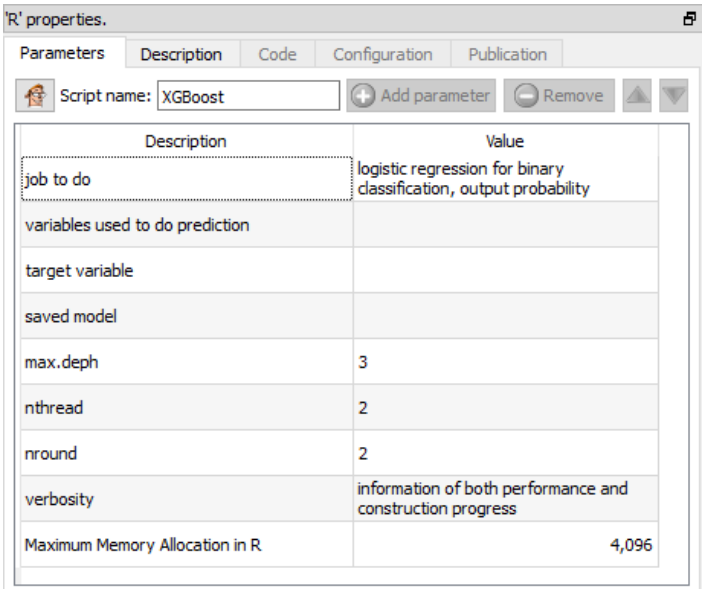
To use the Naïve Bayes action, simply select the predictors, target, and set the model output file name.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.12.7. XGBoost (R action)

Icon:

Property window:



Short description:
Use the XGBoost Library

Long Description:
Gradiant Boosting is probably the most popular algorithm in this second decade of the 21st century. The main reason is that it performed extraordinarily well in most data mining competitions. It usually ensures one of the highest accuracy in situations where the learning and test datasets are from the same time frame.

In practice, we've seen those models degrade very quickly over time (in a banking setting, for example, the accuracy dropped 10 points below LASSO in just two months), so we tend not to use it.

The general idea of gradient boosting is to make ensemble modeling on steroid. By putting together hundreds or thousands of weak models we can obain a fairly good classifier. This is done at the cost of interpetability.

Fit a gradient Boosting model. The different operating modes are:

- linear regression
- logistic regression
- logistic regression for binary classification, output probability
- Multiple classification
- Multiple Probabilities

Note about eta: xgboost automatically does the hyperparameters optimization, but you are free to set the ETA to a lower value. ETA is the step size shrinkage (a bit similar to LASSO) used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features, and it shrinks the feature weights to make the boosting process more conservative. The default value is 0.3. Lower values will take longer to compute and yield potentially overfitting models. A Value of **1** will use a "naïve gradient boosting" algorithm.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.12.8. Time Series (action)

Icon:

Property window:

Description	Value
Unit of Time	Day
Aggregate Time Serie to (Optional)	Month
Aggregation function (Optional)	Mean
STLF	☑
ETS	☑
ARIMA	☑
HOLT-WINTERS	☑
Prediction length to forecast	12
Confidence Interval	95
Time Series Color	
Arima: Information Criterion	bic
Decomp: Trend Smoothing Periodicity	0
Decomp: Season Degree	1
Decomp: Trend Degree	0
Decomp: Robust	☑
Holt Winters Alpha	0.5
Holt Winters Beta	0.1
Holt Winters Gamma (Seasonal)	0.01
Holt Winters Seasonality	Multiplicative

Short description:
Build and compare TimeSeries on sequential data

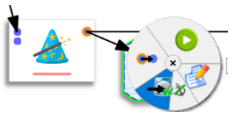
Long Description:
Use this node to compute ETS, STLF, ARIMA and Holt-Winters models.
At the very least, you need to have two series: one with dates (YYYY-MM-dd or YYYY/MM/dd or any separators, as long as the order and number of characters is respected).
Then, you need to specify the time units between observations (**Unit of Time**). Anatella will attempt to do it for you, and if it doesn't match the data, it will recode it. But it's always better to help the software.
Then, specify on which time frame you wish to run the analysis (Aggregate Time Series) to get smoother – and usually better – results. You can specify whether you want the sum or the average computed

Unit of Time	Month
Aggregate Time Serie to (Optional)	Month
Aggregation function (Optional)	Sum

Then, you can optionally uncheck the algorithms you don't want to run (because you know what you are doing... or leave them all checked and let R figure out which one works.

STLF	☑
ETS	☑
ARIMA	☑
HOLT-WINTERS	☑

When you run the algorithms, make sure you force anatella to write cache: right click on the object and set the option



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can set some advanced parameters for the decomposition, for ARIMA and for Holt Winters.

Arima: Information Criterion	bic
Decomp: Trend Smoothing Periodicity	0
Decomp: Season Degree	1
Decomp: Trend Degree	0
Decomp: Robust	☑
Holt Winters Alpha	0.5
Holt Winters Beta	0.1
Holt Winters Gamma (Seasonal)	0.01
Holt Winters Seasonality	Multiplicative

Arima: Information Criterion: By default, you should use BIC as a criterion, as it will perform better if heterogeneity is large (Aho, 2014) , but AIC an AICC are also available and should be checked as other criterion may yield better models. As a reminder, the key difference between AIC and BIC is that BIC penalizes stronger the number of parameters, and this is *often* desirable.
Consider $\widehat{\Theta}$ is the max likelihood estimates of the model parameters, $\log \ell\left(\widehat{\Theta}\right)$ is the log likelihood of the model, p is the number of parameters and n the number of observations.
AIC: Akaike Information Criteria = $-2\log \ell\left(\widehat{\Theta}\right) + 2p$
AICC: Aikake for Small Samples $AIC + \frac{2p(p+1)}{n-p-1}$
BIC: Bayesian Information Criteria. $-2\log \ell\left(\widehat{\Theta}\right) + p\log n$
Thus, when you have few observations (as often in timeseries) AICC would be the most conservative (and preferred) test to apply. When you have "a lot" of points (over 100), BIC or AIC would be appropriate.
The details of the model are available in the log:

```
Model Information:
Series: timeSerie
ARIMA (1,0,1) (2,1,0) [12]

Coefficients:
         arl     mal     sarl     sar2
      0.8020 -0.5961 -0.6803 -0.4174
s.e.  0.1671  0.2171  0.0956  0.0995

sigma^2 estimated as 974.6: log likelihood=-526.74
AIC=1063.48   AICc=1064.07   BIC=1076.89

Error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE
Training set  0.6694643 29.06258 21.78323 -0.4048295 6.963703 0.6720913
                   ACF1
Training set -0.0313262

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 1991       496.5844 456.5766 536.5921 435.3978 557.7710
Feb 1991       438.3955 397.5489 479.2421 375.9260 500.8649
```

Trend Smoothing Periodicity: this parameter will set how many periods are used to smooth the trend line. If you put a small number, you will have a very non-linear trend, if you put a large number you will get something more stable. If you leave it at 0, the default value of R for (`t.window`) is used:

```
nextodd(ceiling((1.5*period)/(1-(1.5/s.window))))
```

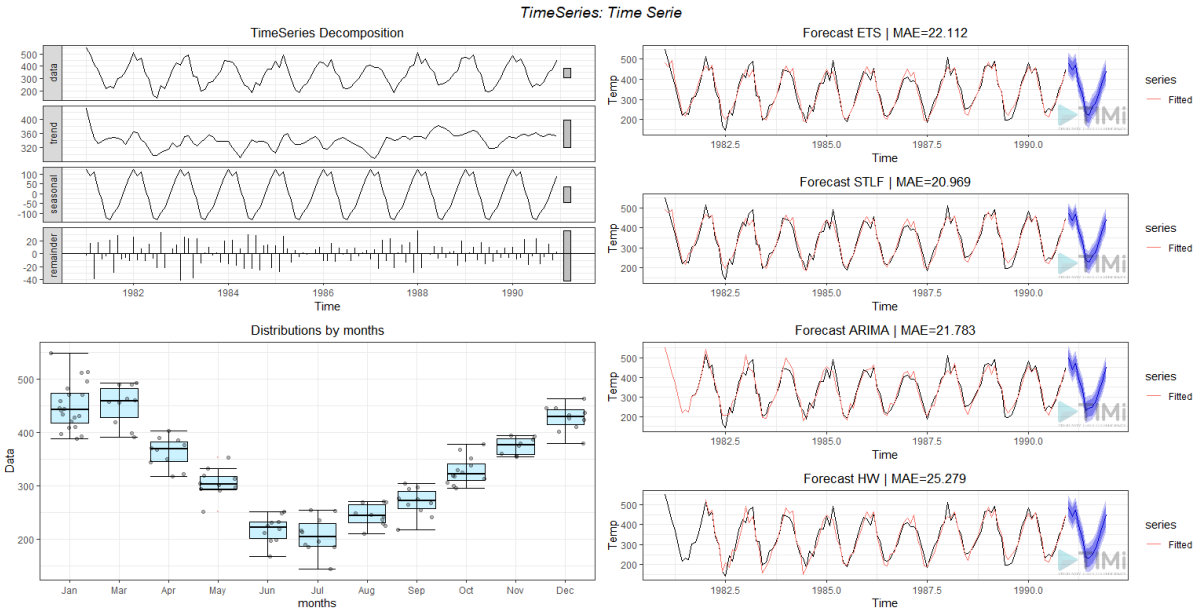
Decomp: Season Degree: (0/1) degree of the function for seasonality. Set to 0 if you don't think there is seasonality
Decomp: Trend Degree: (0/1) degree of the function for trend. Set to 0 if you don't think there is trend
Decomp Robust: Set if you want to use a robust method

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Holt Winters alpha, Beta and Gamma: Set starting values to help avoid local optima.

Outputs

If you are lucky (you have a data structure that fits the time series framework) you will get the following plot:



Here, we can observe the decomposition of trend, seasonality and error of the data, a box plot showing the distribution (error) per time unit, and the results of each algorithm.

You can pick the one with the lowest MAE, of look at the diagnostics for more KPIs:

```
Running... (21/2/22 18:08:13)
[1] "Set Time Unit to Days (minimum frequency), StartUnit== 1"
                       ME              RMSE              MAE                MPE
ETS   -0.225950033418252 27.3589584352165 22.1117188650616 -0.740521333848271
STLF  -0.386518286476155 26.0885511361616 20.9688275434863 -0.918204402968769
ARIMA  0.669464278059237 29.0625764891712 21.7832252493945 -0.404829515401513
HW      3.81109616563921 32.2023638385829 25.2788828886824  0.848436419880703
                    MAPE              MASE                ACF1
ETS    7.05346168010657 0.682226499093432  0.0640085212331007
STLF   6.72201962179413 0.646964168294057  0.0131929824344223
ARIMA   6.9637031816088 0.672091282977547 -0.0313261956920077
HW     8.13514778925323 0.779944963997742 -0.0462278167592328

Success! (finished at 21/2/22 18:08:25 after 12.04 seconds - Peak Memory Consumption~ 295 MB)
```

The same information is available in the second output pin:

DataTable (4 rows - 8 columns) (complete)

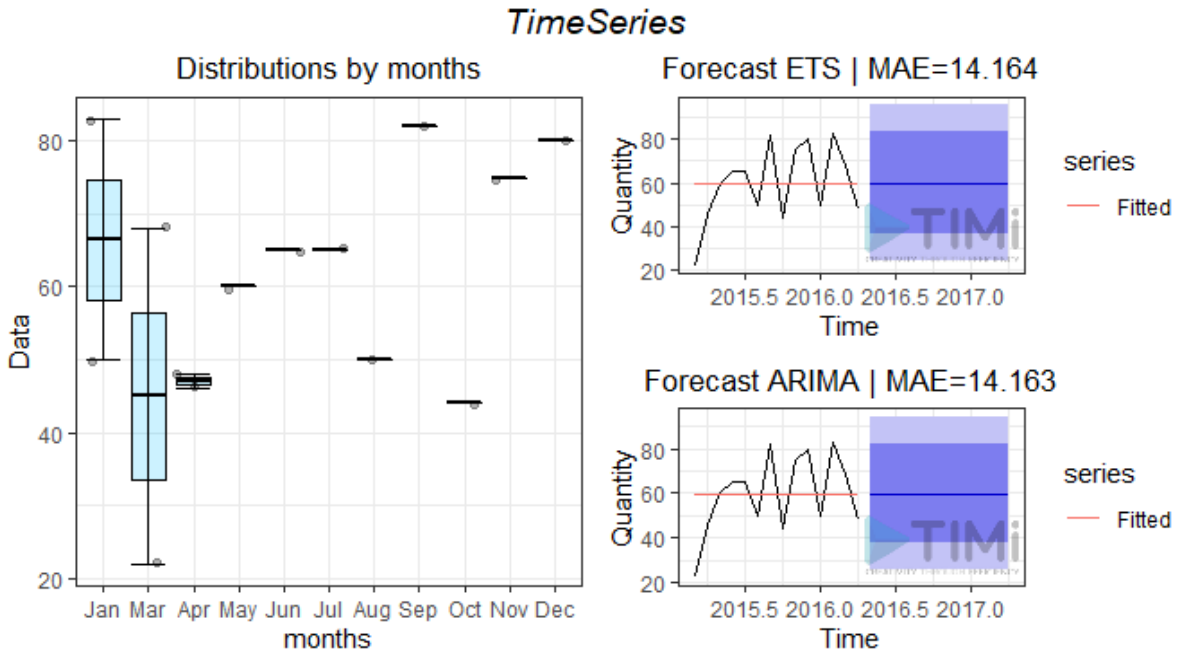
	Algorithm	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
1	ETS	-0.225950...	27.358958...	22.111718...	-0.740521...	7.0534616...	0.6822264...	0.0640085...
2	STLF	-0.386518...	26.088551...	20.968827...	-0.918204...	6.7220196...	0.6469641...	0.0131929...
3	ARIMA	0.6694642...	29.062576...	21.783225...	-0.404829...	6.9637031...	0.6720912...	-0.031326...
4	HW	3.8110961...	32.202363...	25.278882...	0.8484364...	8.1351477...	0.7799449...	-0.046227...

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The first output pin will give you the computed estimates for each observation, as well as the prediction for all successful algorithms (for those, the original observation will be set to 0)

	Date	Data	ETS	STLF	ARIMA	HoltWinters
116	1990-08-01	242.6	260.034	255.422	251.888	250.834
117	1990-09-01	275	284.095	281.239	288.183	286.114
118	1990-10-01	351.7	329.199	330.53	321.13	328.542
119	1990-11-01	379.7	389.976	388.899	392.782	394.526
120	1990-12-01	445.4	436.912	437.743	435.23	443.204
121	1991-01-01	0	477.444	471.583	496.584	481.63
122	1991-01-31	0	446.77	435.606	438.395	439.696
123	1991-03-02	0	471.539	467.222	469.915	470.435
124	1991-04-02	0	376.534	384.666	391.238	391.883
125	1991-05-02	0	321.587	325.81	333.463	325.731
126	1991-06-02	0	231.639	238.868	232.481	235
127	1991-07-02	0	219.582	226.454	240.586	234.434
128	1991-08-02	0	259.029	255.537	246.731	248.574
129	1991-09-01	0	286.098	283.958	279.479	284.592
130	1991-10-01	0	332.773	334.517	334.471	340.23
131	1991-11-01	0	389.667	388.586	376.941	389.385
132	1991-12-01	0	438.377	439.299	449.844	446.48

If we are unlucky, the time series will fail to decompose, and /or you may end up with constant estimates (when the algorithms doesn't simply fail). In this case, only the box plot and timeSeries plots will be returned. You will also see and error message in the log telling you what happened (in R dialect)




```
[1] "Error. Failed to converge STLF Algorithm.\nThe R library said:\nError in stl(ts(deseas, frequency = msts[i]),
s.window = s.window[i], : series is not periodic or has less than two periods\n"
[1] "Error. Failed to converge Holt-Winters Algorithm.\nThe R library said:\nError in decompose (ts(x[1L:wind], start
= start(x), frequency = ff), seasonal): time series has no or less than 2 periods\n"
[1] "nSeries= 2"
[1] "Warning: Failed to decompose TimeSeries. Decomposition not plotted."
<simpleError in stl(timeSerie, s.window = "periodic", t.window = idxPeriod): series is not periodic or has less than
two periods>
```

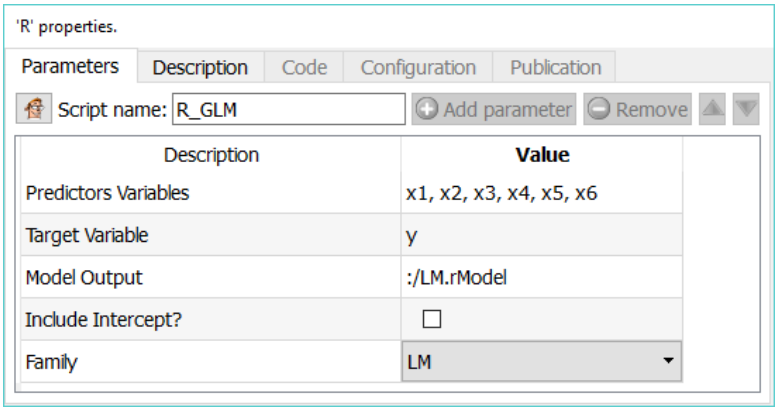

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In this case, STLF and Holt Winters failed, Arima and ETS failed without crashing, and simply return a constant prediction. The reason here is obvious: there is not enough data to find cycles!

5.12.9. GLM(action)

Icon:

Property window:

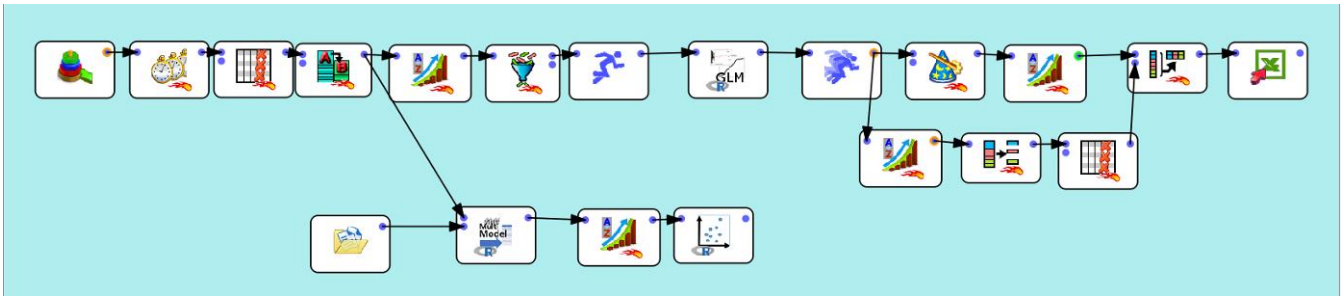


Short description:
Compute Linear Model and MANOVA

Long Description:
This node provides a combination of LM (Linear Model) and GLM (Generalized Linear Model) functionalities.

LM is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance.

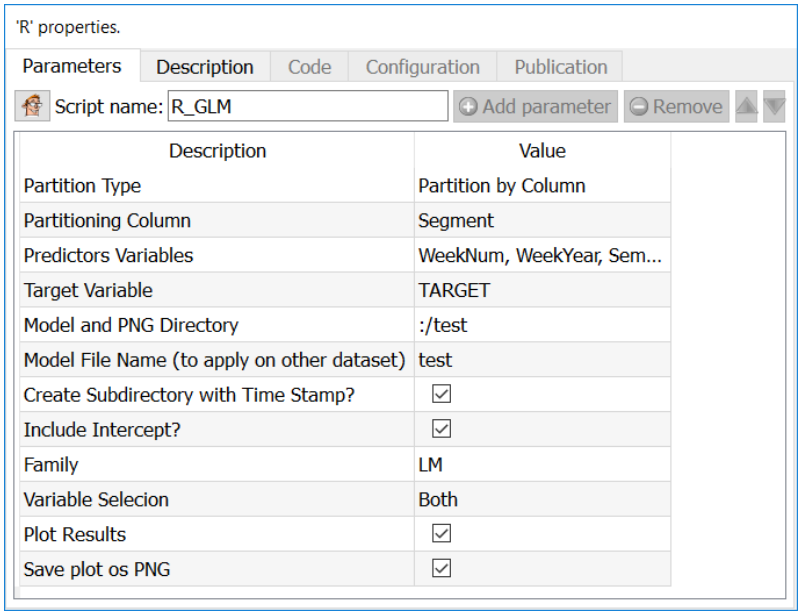
GLM can either compute a single model, or work as a “model factory” when the “partition by column” mode is selected.



In this example, we prepare data using a “Time Travel” transformation, and we have several time series (per location, brand, segment… you name it), and we want to have one predictive model for each series, in a single operation.

We simply sort the data using the partition variable, and properly set the GLM node:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Set the "**partition type**" function to "**Partition by Column**" and select the partitioning column (in our example, "**segment**"). The Model name is now overwritten, and one model will be created for each value of the partitioning variable. All other functionalities remain the same. Note that you will need to use the ModelApply

to score

Parameters:

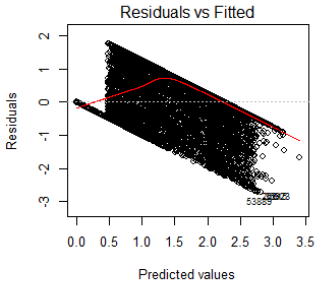
Predictors Variables: Select independent variables
Target Variable: Select the variable you want to predict
Model and PNG directory: Set directory in which you wish to save files
Model File Name: select the name of the rModel file in which the model is saved. If the extention is omitted, it will be set by default. This value is overwritten in case of a "partition by column"
Create Subdirectory with Time Stamp: in some cases, it is interesting to create a new directory with the time of execution of the model. This way, you can easily generate many models and compare them without adding additional nodes.
Include Intercept: try the regression with or without intercept
Family:

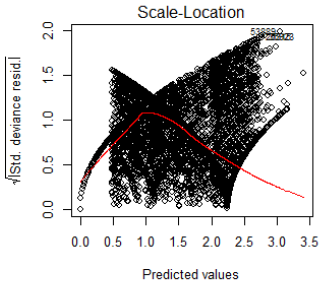
- LM (Linear Model)
- BINOMIAL
- GAUSSIAN
- GAMMA
- INVERSE GAUSSIAN
- POISSON
- QUASI

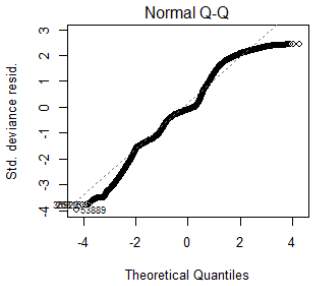
Plot results: choose whether to plot the output or not
Save plot as PNG: you can automatically save the graphical output as a PNG file.

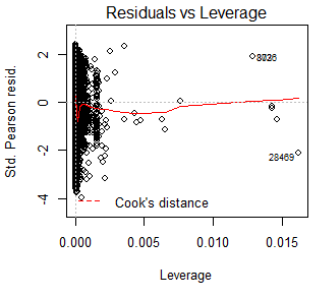
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Output









Charts:
Residual vs Fitted, Scale-Location, Normal Q-Q, Residual vs Leverage

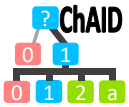
Model output in Anatella LOG

Name of the model (Family)

Formula syntax
Coefficient (x1…xn)
Degrees of freedom
Null Deviance
Residual Deviance and AIC
ANOVA

5.12.10. CHAID(action)

Icon:



Property window:

'R' properties.

Parameters | Description | Code | Configuration | Publication

Script name: CHAID | Add parameter | Remove

Description	Value
predictors	
Target	
model name	:/CHAID.rModel
Plot Charts (If not, the model is still generated)	☑
Mininimum frequency of observations in terminal nodes	0.01
Number of observations in splitted response at which no further split is desired.	1,000
Minimum number of observations in terminal nodes	800
Level of significance used for merging of predictor categories	0.05
level of significance used for the the splitting of former merged categories of the predictor	-1
Level of significance used for splitting of a node in the most significant predictor	0.05
PLOT Font Size	6
Node Color	
Terminal Classification Palette	topo.colors

Short description:
Create a CHAID Model

Long Description:
The CHAID (CHi-squared Automated Interaction Detection) algorithm is included in Anatella mainly because:

- …of historical reasons. The CHAID algorithm is a legacy of the early days of research in machine learning around the subject of "*automatic creation of classification trees*". Since the '90, the researchers in machine learning have created many newer&better algorithms to create "better" tree's than ones created with the CHAID algorithm (i.e. trees with higher AUC, and higher accuracy than the ones created with CHAID). Chronologically, the algorithms are as follow: CHAID (the oldest and the worst one), ID3, C45, C50, CART (the newest). This is obviously a non-limitative list. Nowadays, nobody uses a single tree as a classification algorithm (mainly because predictive models based on a single tree are too unreliable because they degrade too quickly over time). If you want to use some kind of "tree" algorithm, you'd

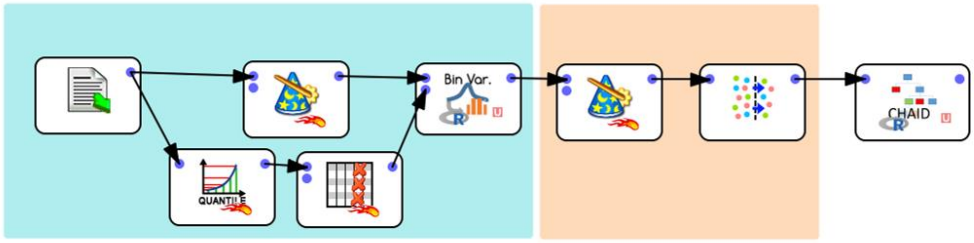
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

rather use “Forest of Trees”, or even better, “Forest of Stumps” (Stumps are “small trees” limited to, typically, maximum 3 “levels deep”). Such algorithms are available in sections 5.12.4 and 5.12.5.

- …it can be useful for explanatory/teaching purposes.
- …it’s still probably one of the most widely used tree algorithm in marketing and market research (despite his very poor results as a classification algorithm). Indeed, if your objective is an explanatory objective (i.e. you want discover and explain SMALL datasets), CHAID can maybe be a useful algorithm.

The CHAID (CHi-squared Automated Interaction Detection) algorithm is Chi-square based, meaning that it uses a statistical significance test to decide if it’s worth proceeding with an additional cut in one of the nodes. Since CHAID is based on a Chi-square significance test, it won’t work on large samples, with more than 1000 rows (since such significance tests are useless on large datasets). The way it proceeds is comparing the two groups that would be formed, and decide if there is a statistically significant difference between the two groups. As the groups become “too small”, the confidence interval increases and eventually the cuts will not be worthwhile anymore. CHAID also offers the neat functionality of doing multiple cuts per level (not just binary).

One additional limitation of CHAID (it’s a very old algorithm) is that all the variables must be CATEGORICAL (in R: Factors), so you may want to reduce the number of categories using binning functionalities (make small groups based on cut-offs) using such an Anatella graph:



Because of this limitation, CHAID requires a few transformations before we can use it:

0- Balance the dataset between categories (aim for roughly the same proportions)
1- Recode variables based on “Clever Quantile”. This first operation also requires that the data be set to numerical, and only the “Clever Quantile” results are selected from the bottom transformation
2- Transform the recoded variables to TEXT, and select a sample (because CHAID is relly not designed for populations)
3- Run CHAID

CHAID’s Parameters:

List of Predictors: Select independent variables
Target: Select the variable you want to predict
Model Output: Set the file name for the model results
Plot charts: Plot the tree
Mininimum frequency of observations in terminal nodes: percentage of the dataset in end node
Number of observations in splitted response at which no further split is desired: self explanatory
Minimum number of observations in terminal nodes: Criteria to decide whether to proceed with a cut or not.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The following options should typically not be changed:

Level of significance used for merging of predictor categories
level of significance used for the the splitting of former merged categories of the predictor
PLOT Font Size: 0 is automatic size

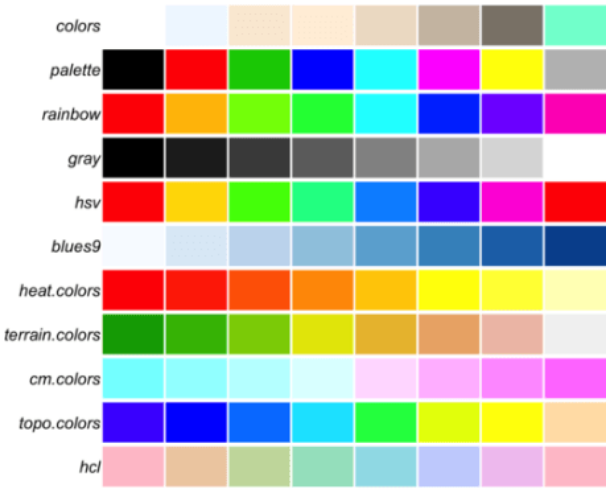
The other parameters can usually be left “as is”, refer to the R documentation for further information.

Other reasons why CHAID is seldom used:

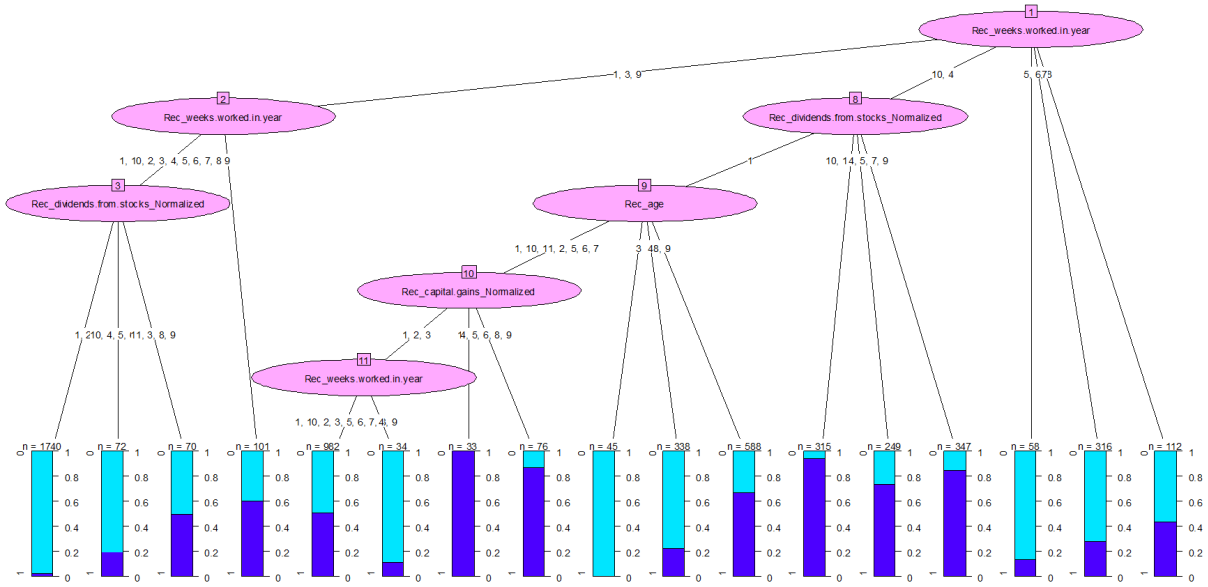
- it requires a lot of efforts to prepare data,
- it will not work well with large amount of data (datasets with more than 1000 rows),
- it does not like numbers.

Basically, it is not a good algorithm for predictive modeling on real data.

For the final node, we are using the following color options:



The CHAID tree is fairly easy to understand: At each node, we see a split with details of the criterion:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

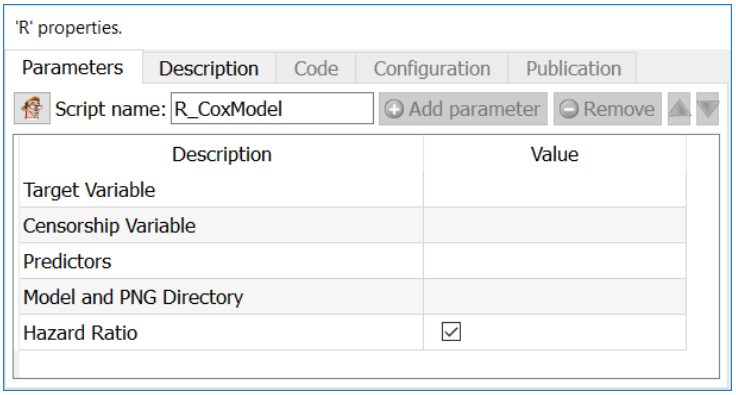
In the log window, you will also see this same information, but in text (remember to set the “fixed width” parameter in the Edit menu for a nicer display)

```
Log
Model formula:
taxable.income.amount ~ Rec_age + Rec_age_Normalized + Rec_capital.gains_Normalized +
    Rec_capital.losses_Normalized + Rec_dividends.from.stocks_Normalized +
    Rec_num.persons.worked.for.employer_Normalized + Rec_wage.per.hour +
    Rec_wage.per.hour_Normalized + Rec_weeks.worked.in.year

Fitted party:
[1] root
|   [2] Rec_weeks.worked.in.year in 1
|   |   [3] Rec_capital.gains_Normalized in 1, 2, 3, 4, 5
|   |   |   [4] Rec_dividends.from.stocks_Normalized in 1, 2, 9
|   |   |   |   [5] Rec_dividends.from.stocks_Normalized in 1, 10, 11, 2, 3, 4, 5, 6, 7, 8
|   |   |   |   |   [6] Rec_age in 1, 2, 3, 4, 6, 7: 0 (n = 1067, err = 0.0%)
|   |   |   |   |   [7] Rec_age in 10, 11, 5, 8, 9: 0 (n = 494, err = 6.7%)
|   |   |   |   [8] Rec_dividends.from.stocks_Normalized in 9: 0 (n = 27, err = 44.4%)
|   |   |   [9] Rec_dividends.from.stocks_Normalized in 10, 11, 3, 4, 5, 6, 7, 8: 0 (n = 88, err = 33.0%)
|   |   [10] Rec_capital.gains_Normalized in 10, 6, 7, 8, 9: 1 (n = 25, err = 8.0%)
|   [11] Rec_weeks.worked.in.year in 10, 9
|   |   [12] Rec_dividends.from.stocks_Normalized in 1
|   |   |   [13] Rec_age in 1, 10, 11, 2, 8, 9: 1 (n = 828, err = 34.1%)
|   |   |   [14] Rec_age in 3: 0 (n = 48, err = 6.2%)
|   |   |   [15] Rec_age in 4: 0 (n = 333, err = 20.1%)
|   |   |   [16] Rec_age in 5, 6, 7: 1 (n = 895, err = 44.5%)
|   |   [17] Rec_dividends.from.stocks_Normalized in 10, 11, 4, 5, 6, 8, 9: 1 (n = 557, err = 10.1%)
|   |   [18] Rec_dividends.from.stocks_Normalized in 2, 3, 7: 1 (n = 393, err = 22.6%)
|   [19] Rec_weeks.worked.in.year in 2, 3, 4: 0 (n = 254, err = 10.2%)
|   [20] Rec_weeks.worked.in.year in 5: 0 (n = 112, err = 18.8%)
|   [21] Rec_weeks.worked.in.year in 6, 7, 8: 0 (n = 316, err = 31.6%)

Number of inner nodes:     7
Number of terminal nodes: 14
```

5.12.11. Cox Model(action)

Icon:

Property window:



Short description:
Survival Models with COX regression

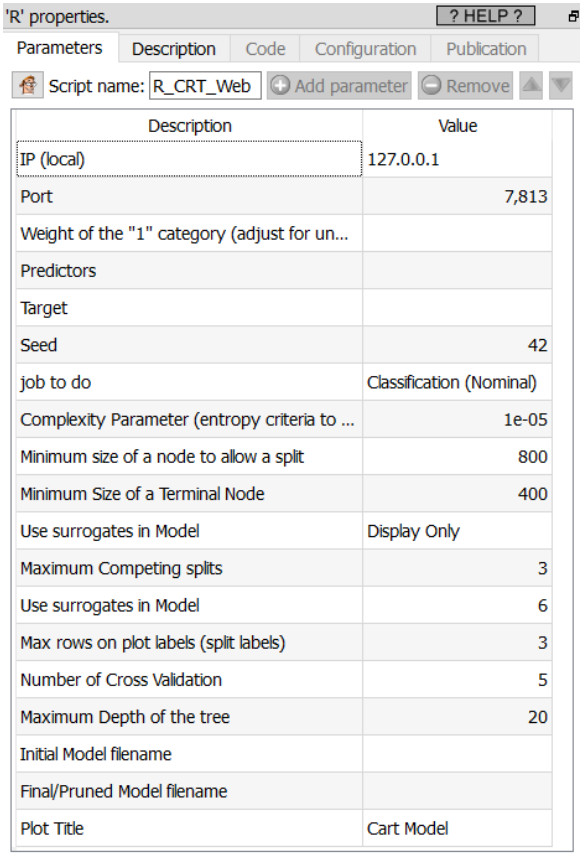
Long Description:
Survival Models with COX regression
Target variable: Select the dependant variable (one Column, Numeric)
Censorship Variable: Select the independent variables (many columns)
Predictors: Select the independent variables (many columns)
Model and PNG Directory: Leave blank if you do not want to save, set a directory if you want to save
Hazard ration: Boolean, specify whether to include a hazard ratio in the model

5.12.12. Interactive CART (action)

Icon:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Property window:



Description	Value
IP (local)	127.0.0.1
Port	7,813
Weight of the "1" category (adjust for un...	
Predictors	
Target	
Seed	42
job to do	Classification (Nominal)
Complexity Parameter (entropy criteria to ...	1e-05
Minimum size of a node to allow a split	800
Minimum Size of a Terminal Node	400
Use surrogates in Model	Display Only
Maximum Competing splits	3
Use surrogates in Model	6
Max rows on plot labels (split labels)	3
Number of Cross Validation	5
Maximum Depth of the tree	20
Initial Model filename	
Final/Pruned Model filename	
Plot Title	Cart Model

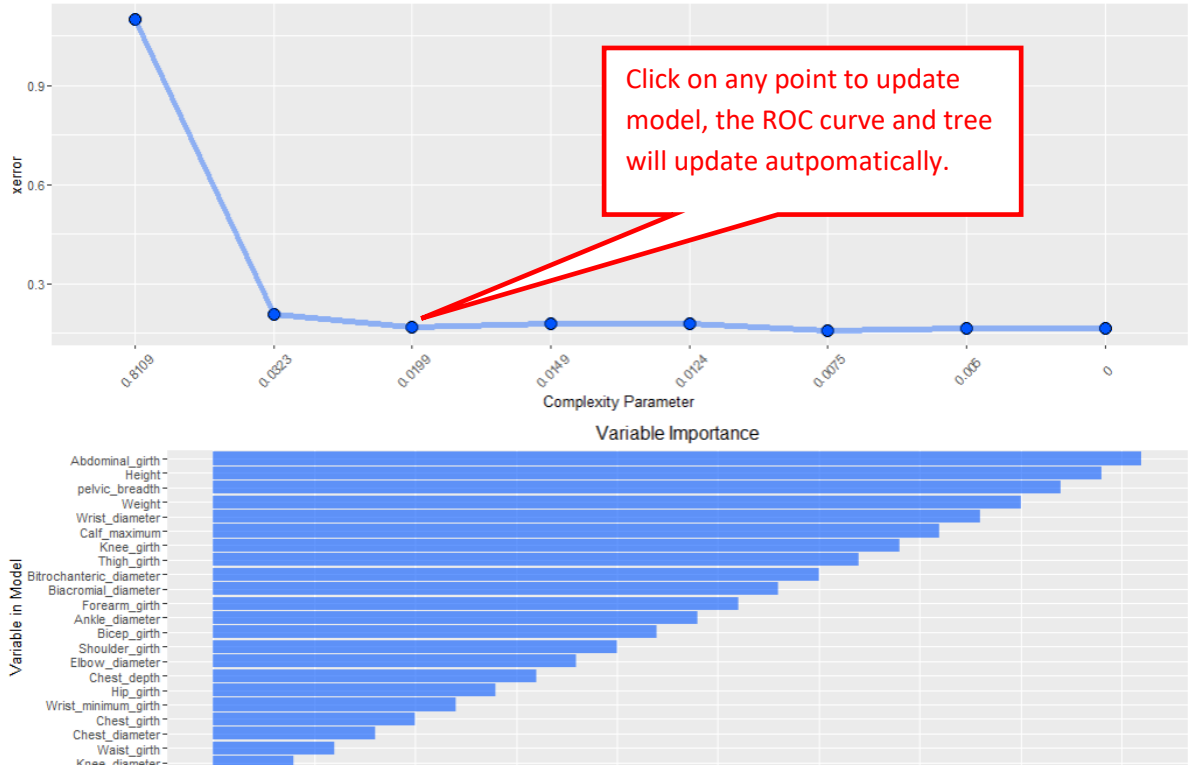
Short description:
Build a CRT model

Long Description:
CRT trees are sometimes better built “intuitively”. While there is a KPI that lets us know where to prune, there is a subjective interpretation that is better not to leave to algorithm.
This node generates a tree and give a web application in a browser in which you can choose where to cut.

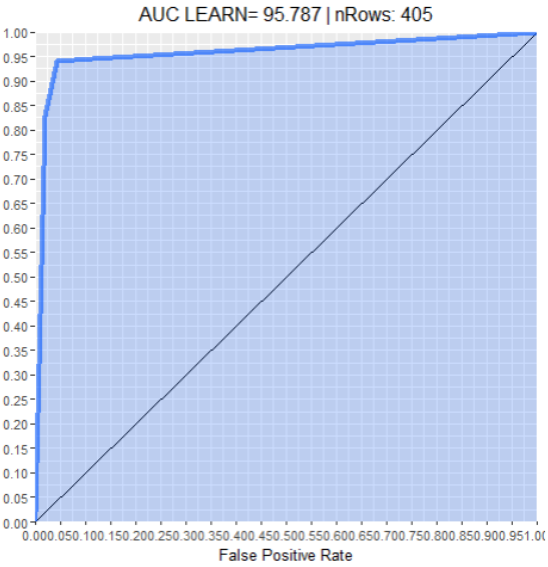
IMPORTANT:
Make sure to specify the two models: Initial and Final/Pruned models. Those to model files will be saved autonatically. For more information about the CRT model, see the sections 5.12.2. and 5.12.3.

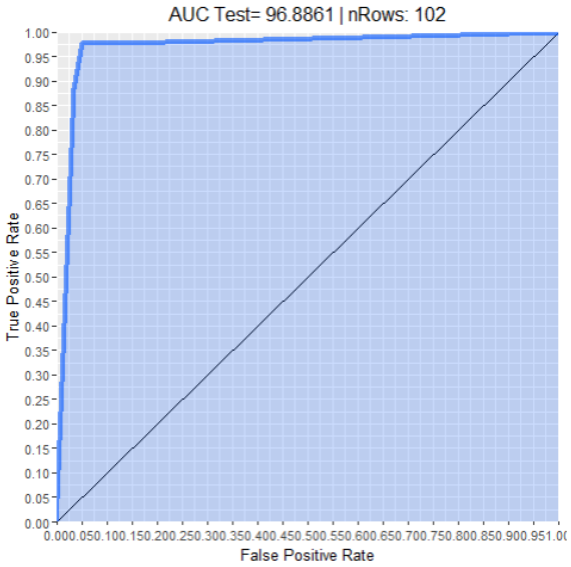
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Cart Model



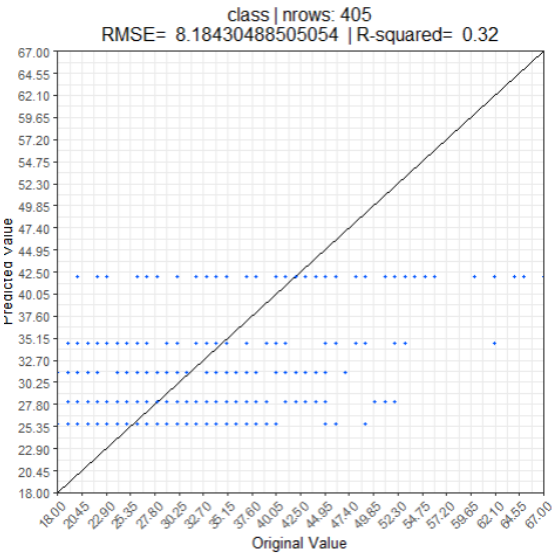
AUC automatically adapts to the new selection. Once you see the AUC is stable between learn and test, you probably have a good model.

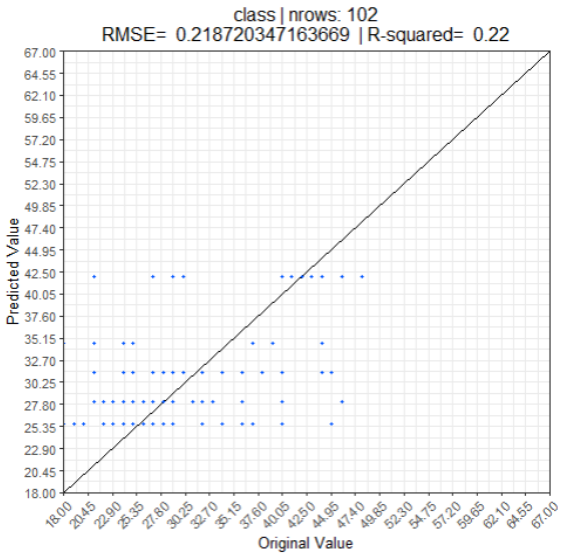




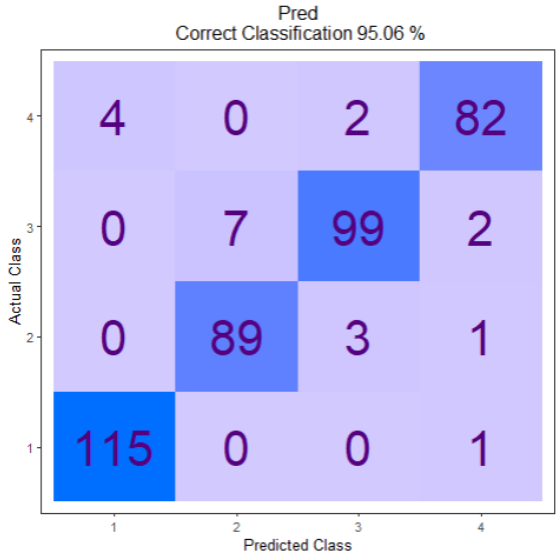
If the target is continous, instead of the AUX, the model will compute the RMSE and R^2

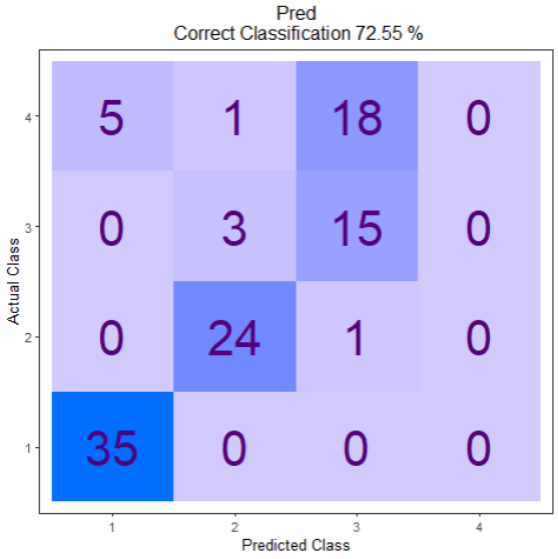
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



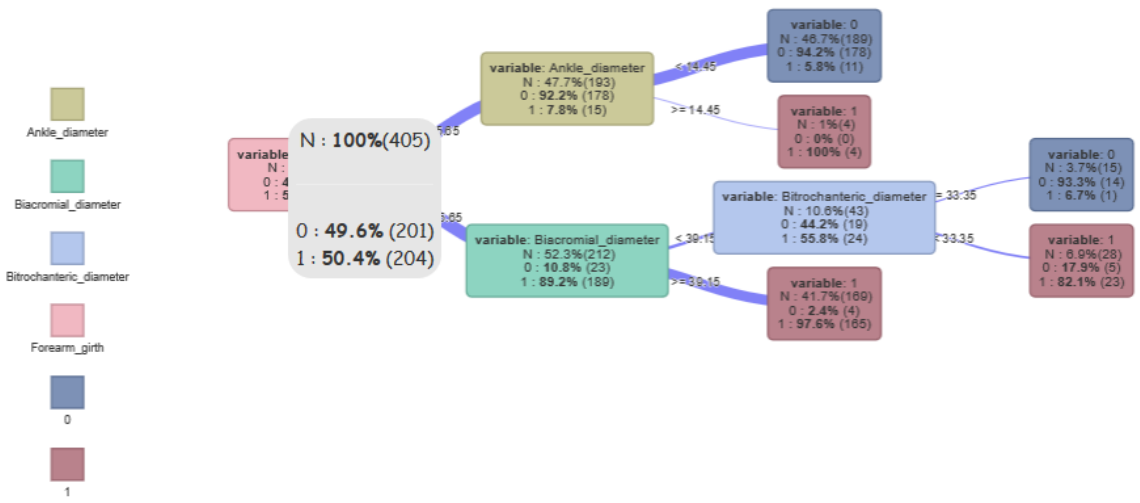


If the target is multinomial, the plots will be a classification table.





You can zoom on the tree to see more details, and on-MouseOver, more details will display about the node.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.12.13. C50 (action)

Icon:

Property window:

'R' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: R_C50 | Add parameter | Remove

Description	Value
predictors	
Target	
job to do	Classification (Nominal)
model name	
Seed	0
Trials	1
Node Colors	
Terminal Classification Palette	rainbow
Plot charts	☑
Plot Font size	4

Short description:
Build a C50 model

Long Description:
C50 models are included for completeness and educational purpose, In pretty much all situations, you will prefer to use a CRT model.

In theory, C50 sounds like an attractive model as it can have any number of branches and appears to give models with higher accuracy.

In practice, C50 are much slower and hungry in resources, and you will need to select a sample in order to build a simple model on 200.000 rows. C50 also have a bad tendency to overfit the data, thus creating accurate but unreliable models.

C5.0 algorithm is an extension of C4.5 algorithm. C5.0 is the classification algorithm which applies in big data set. C5.0 is better than C4.5 on the efficiency and the memory. C5.0 model works by splitting the sample based on the field that provides the maximum information gain. The C5.0 model can split samples on basis of the biggest information gain field. The sample subset that is get from the former split will be split afterward. The process will continue until the sample subset cannot be split and is usually according to another field. Finally, examine the lowest level split, those sample subsets that don't have remarkable contribution to the model will be rejected.

Gain is computed to estimate the gain produced by a split over an attribute
Let S be the sample:

- Ci is Class I; i = 1,2,…,m
  I(s1,s2,..,sm)= - Σ pi log2 (pi)
- Si is the no. of samples in class i
- Pi = Si /S, log2 is the binary logarithm
- Let Attribute A have v distinct values.
- Entropy = E(A) is
  Σ{(S1j+S2j+..+Smj)/S}*I(s1j,..smj) j=1
- Where Sij is samples in Class i and subset j of Attribute A.
  I(S1j,S2j,..Smj)= - Σ pij log2 (pij)
- Gain(A)=I(s1,s2,..,sm) - E(A)

Gain ratio then chooses, from among the tests with at least average gain, The Gain Ratio= P(A)

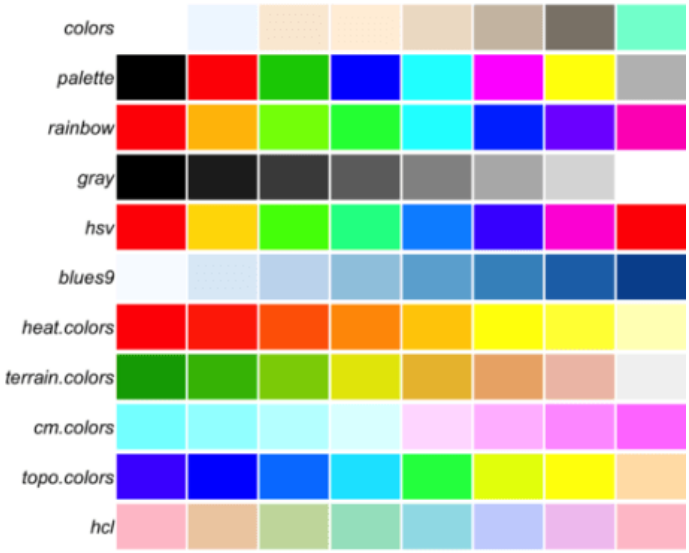
$$\sum_{i}^{t} \frac{S_i}{S} log\left(\frac{S_i}{S}\right)$$

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Gain Ratio(A)= Gain(A)/P(A)
(International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 4, June - 2012 ISSN: 2278-0181)

Parameters:

Predictors: select the variables to use as predictors. You may mix numerical and text variables (remember to set the numeric variables to KEY or FLOAT with the "cast" action)
Target: set the target variable
Job to do: classification (nominal) or Regression (continuous)
Model Name: set the file name in which to save the rModel for scoring
Seed: set a number to generate random seeds
Trials: Enable a boosting procedure, this method is more similar to AdaBoost than to more statistical approaches such as stochastic gradient boosting.
Node Color:



Plot charts: true/false
Include Prediction: Create a column with predicted classification or value.

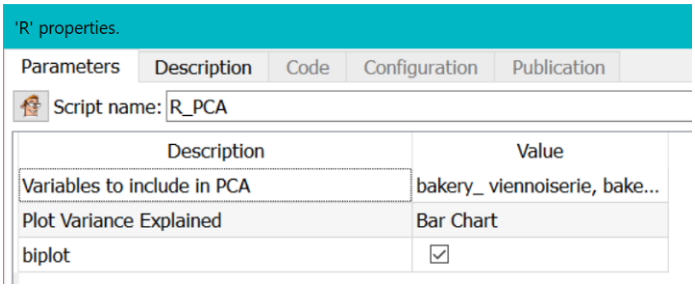
5.13. TA – R Discovery Analytics (TA=Transformations Actions)

As for the predictive analysis tools, actions in this category allow the automated selection of variables. Most actions have the following structure. As the entire matrix will be sent to the R engine, it is a good idea to make a preselection of the relevant variables if you have many thousands of dimensions.

5.13.1. PCA (R action)

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Compute the PCA.

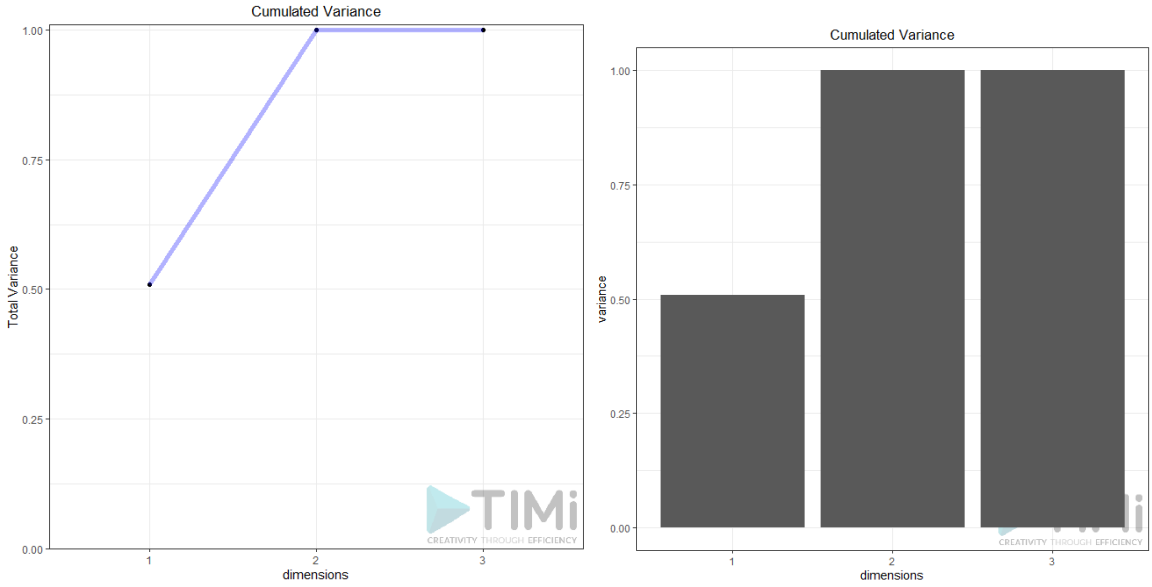
Long Description:
Principal Component Analysis is a data reduction technique that extract the linear components which are underlying a selection of variables. This action uses the prcomp command of the R Stat Library. Simply select the variables you wich to include, and the recoded columns will be added as new columns.

The tables in **input** are

- Pin 0: data table
- Pin 1: Variable Names
- Pin 2: Optional Preference Data

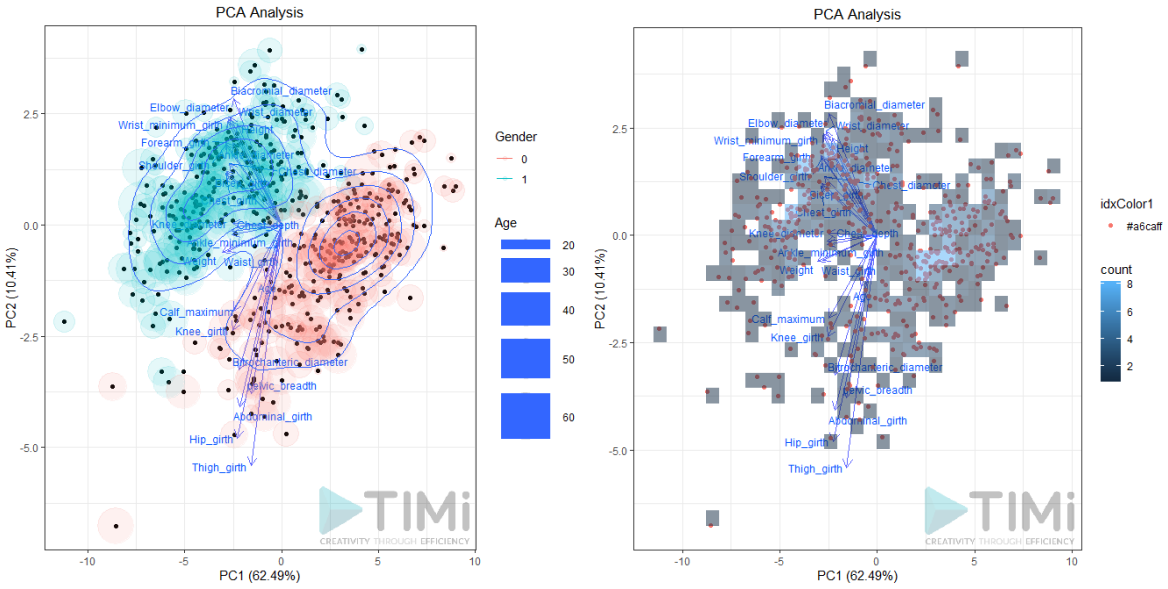
The tables in **output** are:
pin 0: intput data matrix represented in the new coordinate system composed of the eigen vectors
pin 1: eigen vectors inside the orignal coordinate system
pin 2: eigen values

Parameters:
PCA Title: Title of the chart.
Variables to include in PCA: select the list of variables. Remember that if you have a few independent variables, those will add noise in your PCA. Some manual selection is always needed.
Plot Variance Explained: None, Line Plot or Bar Plot (the example below shows both plot)



Biplot (true / false): Plots the first two dimensions. Observations are light dots, variables dark blue. The density can be represented in "2D density" (left plot) or BIN (right plot). As displayed below, you can set many options to play with point size, transparency, add label, add density, etc.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Preferences: if the second pin is connected, the PCA will be applied to this dataset. You can choose to plot it as point or as vector. In each case, you can set the transparency (between 0 and 1)

5.13.2. Hierarchical Clustering (R action)

Icon:

Property window:

'R' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: RARCHICAL | Add parameter | Remove

Description	Value
Chart Title	
Variables to include in Clusters	
Method	Ward.D2
Try All Method and plot charts?	☐
Save Image in PNG? (empty field=no save)	☐
PNG Directory	:/

Short description:
Hierarchical Clustering

Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

A classic algorithm that always works, as long as data is numerical and not too big.

Typically, hierarchical clustering is used in combination with K-Means, to find the optimum the number of segments. In general, it is not recommended to trust segments assignments made by hierarchical clustering.

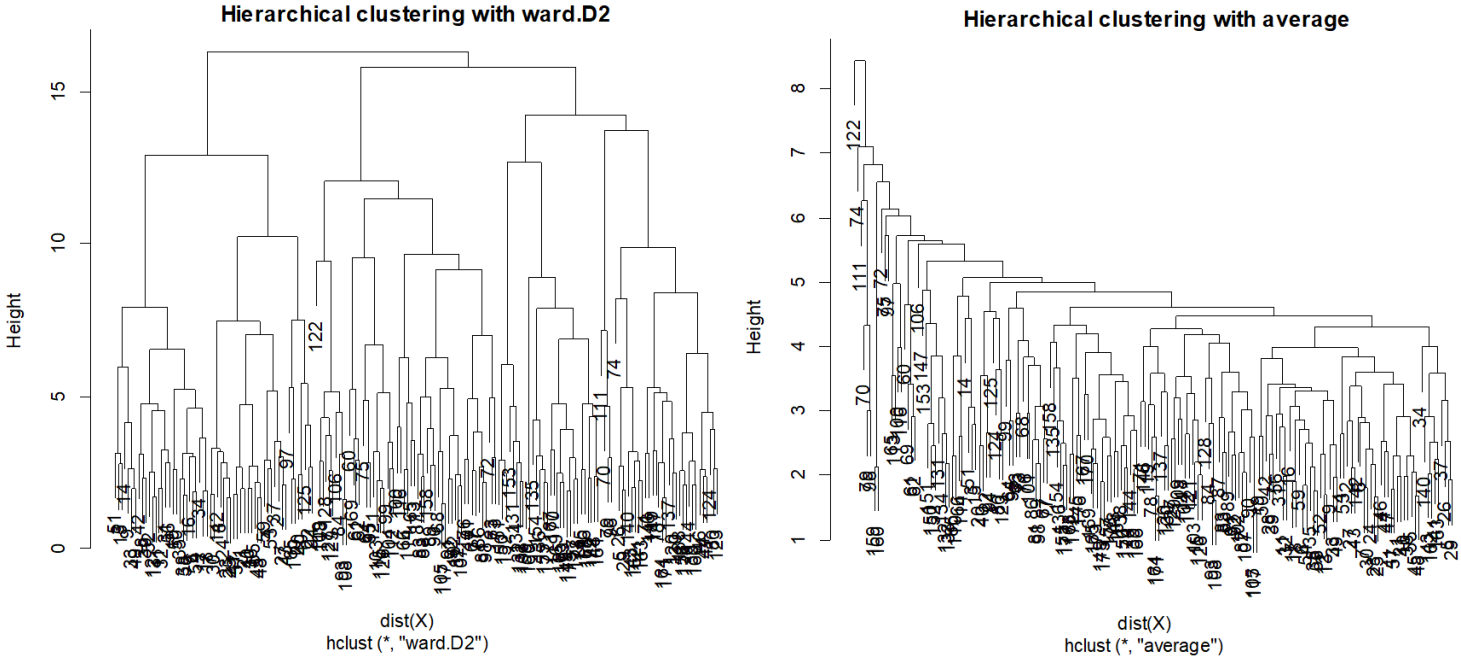
The main limitations of Hierarchical clustering are:

- it needs to start by computing the distances betweem each point, hence creates an n x n matrix in memory. This will not work with “large” database (1000 records are ok. 10.000 is often a problem. 10.000 will require a large server with a LOT of ram)
- Curse of dimensionality: if you put a lot of variables, segments will not appear (everything will look equidistant)
- it is slow

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- cluster centers are not dynamic: as we regroup, centers change and some points may become misclassified

While several methods are included in this action, it is best to used Ward.D2: This is the same distance estimation used in other popular statistical software (Stardust, SPSS, etc.), and it tends to give the clearest dendograms: compare Ward with Average Linkeage method: The latter tends to create segments of outliers and fails to provide a clear cut un terms of number of segments.



To know how many segments to retain, one must "*look for a large drop of information*", and explore the solutions of the various potentially "good" solutions.

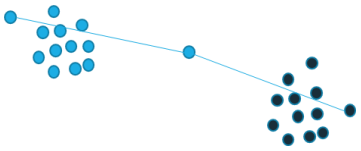
Parameters:

Chart Title: Title of the plot. It will display "Hierarchical Clustering of " **TITLE** " with METHOD"
Variables to include in Clusters: select the columns on which to compute the clusters
Method: choose one of

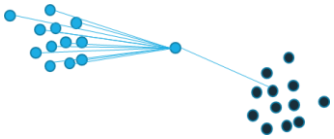
Ward.D: Ward's distance, Minimize the total variance of the clusters. Proximity between two clusters is the magnitude by which the distance in their joint cluster will be greater than the combined distance in these two clusters: SS12−(SS1+SS2)
Ward.D2: Squared Ward's distance (the most common one), we use the sum-square instead of the distance
Single: Single Linkage follows the logic of "a friend of a friend is a friend", in which points are assigned to the segment with the closest point

Complete: Complete linkage follows the logic of "the one I hate the most is my friend", points are assigned to the segments that have the least distant extreme, that is, the farthest point is the closest.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Average: Average Linkage: we take the average distance of all points for all clusters, weighted by the number of points in each cluster. You'd expect good segments, they are often not that clear.



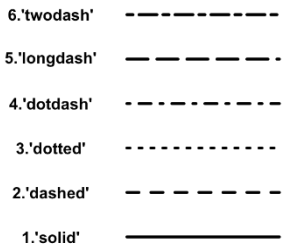
Mcquitty: average, without the weight.
Median: (WPGMC) Similarity based on the median of each cluster (similar to K-Medoid) using Euclides' distance

Centroid: UPGMC, distance to the center of each cluster using Euclides´s distance.

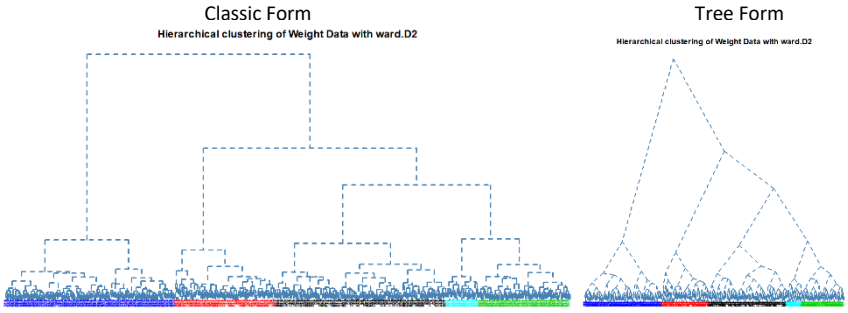
Direction: Select:

Downwards
Upwards
Left
Right

Try all methods and plot chart: run all method so we can choose which is best, visually
Save image as PNG: self explanatory
PNG Directory: specify the directory in which to save the PNG, the name will be the plot title
Row Labels: mandatory, select the variable with labels names, it can be the key.
Label Size: label size on the plot
Line Type:

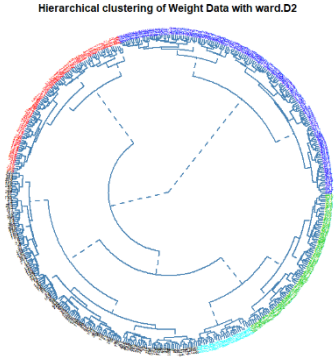


Dendogram Type:

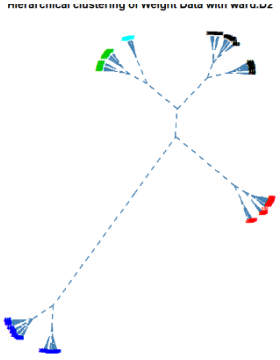


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

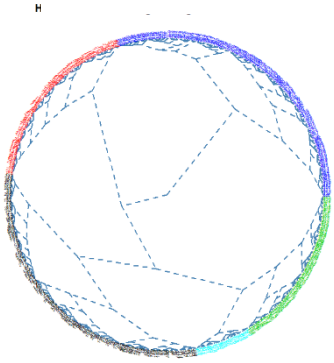
Fan (genetic) Form



Unrooted Form



Radial Form



Number of Clusters: set the number of clusters to display in color on the plot (default is 1)
Cluster Name: name of the cluster

5.13.3. K-Means Clustering (action)

Icon:

Property window:

'R' properties.

Parameters | Description | Code | Configuration | Publication

Script name: R_KMEANS | Add parameter | Remove

Description	Value
Variables to include in Cluster Solution	
Method	Hartigan-Wong
Scale Matrix before clustering?	☑
Seed	42
Number of Segments	4
Maximum number of iterations	200
Number of start	4
Cluster Name	Cluster
Include distance from center?	☑
Plot Results?	☑
Chart Title	K-Means Segments
Model Name (to apply on other dataset)	

Short description:
K-Means Clustering

Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

The K-Means algorithm is the most commonly used clustering/segment algorithm used everywhere. If you search for a good starting point to start segmenting your data, use the K-Means algorithm.
The Kmeans action will output a new column with the cluster number, and columns with the distance between each point and the center of each segment. You can easily transform this information into probability.

The K-Means algorithm has several advantages agains hierarchical clustering. To being with, it will work on large amount of data at a more reasonable speed. It also allows observations to switch group as

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

cluster centers are recomputed. These advantages made K-Means the most popular segmentation method, but the K-Means algoritm has sevelar limitations:

1. it assumes all variables are continuous, identically distributed, and belong to spherical segments of equal variance (this is probably the most limiting hypothesis)
2. it assumes that all variables are on the same scale. While this is problematic, this is easily circonveined by normalizing the data or simply selecting the option "scale matrix"
3. it assumes that all axises (i.e. all variables) are non-correlated. You can check the correlations that exists between the variables using the covariance action (see section 5.7.11). You can easily obtain non- correlated variables (i.e. perpendicular axises) by running the k-means algorithm inside the space spanned by the first "Principal Component Axises" (i.e. the first PCA's). Since is why, most of the time, you should (nearly always) use a PCA action (see section 5.12.7) before the K-Means action.
4. It does not accept any non-numerical variables
5. The number of clusters (k) must be known before hand. It is therefore often used in combination with hierarchical (ward) clustering (to find the correct value for k).

Algo: Method:

In most cases, H-W is used. However, all other algorithms are available.

The algorithms used inside this Action are described in the following papers:

- Forgy, E. W. (1965) Cluster analysis of multivariate data: efficiency vs interpretability of classifications. Biometrics 21, 768–769.
- Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. Applied Statistics 28, 100–108.
- Lloyd, S. P. (1957, 1982) Least squares quantization in PCM. Technical Note, Bell Laboratories. Published in 1982 in IEEE Transactions on Information Theory 28, 128–137.
- MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, eds L. M. Le Cam & J. Neyman, 1, pp. 281–297. Berkeley, CA: University of California Press.

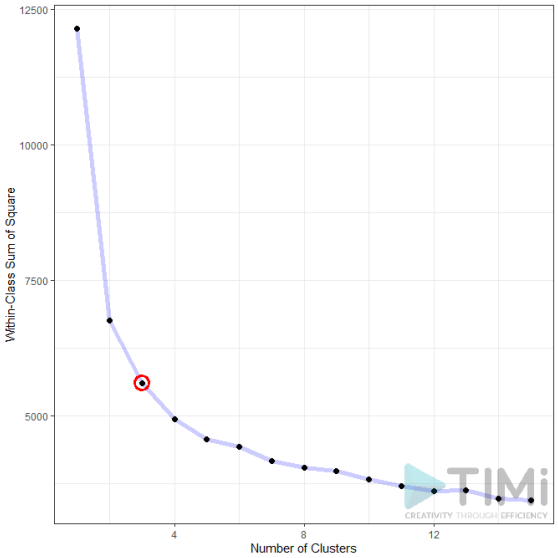
Parameters:

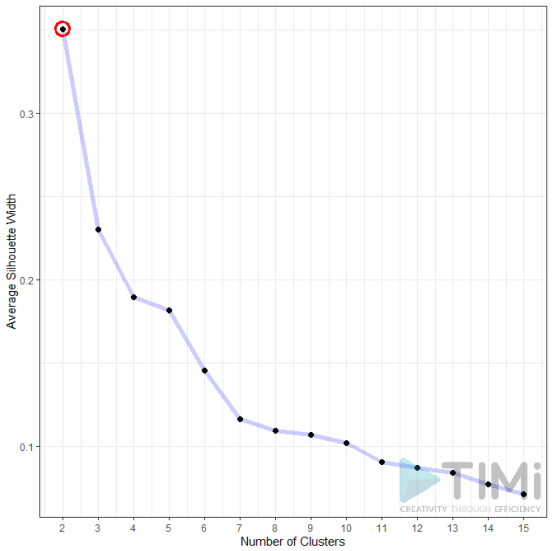
ALGO: WSS optimal Number Selection: use the elbow method or the WSS to select the optimal number of segments. While a visual selection with Hierarchical is often preferred, those two methods offer an alternative selection of the number of segment. The **Elbow** method selects the inflection point of the **Within Sum of Square** line, while the **silhouette** method is defined as follows:

> "Put a(i)= average dissimilarity between I and all other points of the cluster to which I belong. (if i is the *only* observation in its cluster, s(i)=0 without further calculations). For all *other* clusters C, put d(i,C) = average dissimilarity of i to all observations of C. The smallest of these d(i,C) is b(i)=minCd(i,C), and can be seen as the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does *not* belong. Finally,
>
> $$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$
>
> Observations with a large s(i) (almost 1) are very well clustered, a small s(i) (around 0) means that the observation lies between two clusters, and observations with a negative s(i) are probably placed in the wrong cluster."[1]

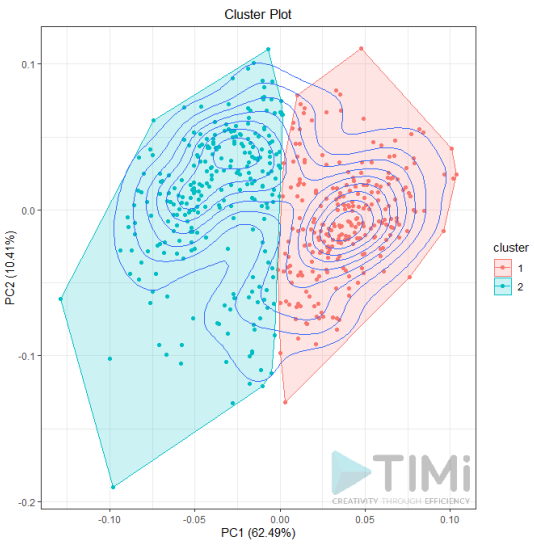
[1] https://www.rdocumentation.org/packages/cluster/versions/2.1.0/topics/silhouette

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu





ALGO: Scale Matrix before clustering: proceed with a normalization of the data to avoid dominance from varaibles on a larger scale.
ALGO: Seed: set a seed number so you can run the same analysis again, with consistent results.
ALGO: Number of segments: Select the number of segments to keep.
ALGO: Number of iteration: set stopping criteria in case of no convergence.
ALGO: Number of starts: How many random sets are chosen.
OUT: Cluster Name: name of the variable with the cluster results.
ALGO: Include distance from center: include Euclidean distance from centers as new variables.
CHT: Plot Results: Select whether to display a distribution chart
CHT: Chart title: set the title of the chart (if you selected the previous option)
CHAT: Cluster Border: Limites or Ellipse.



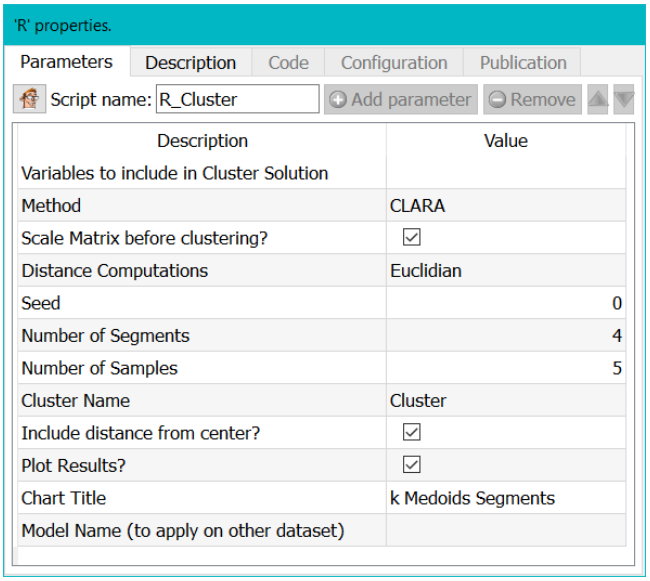
Model Name: Name of the model to use for later scoring.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.13.4. K-Medoids Clustering (action)

Icon:

Property window:



Short description:
K-Medoids Clustering

Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

K-Medoid is an alternate clustering technique that performs better than K-Means with non-spherical segments. It is, however, quite slow and impossible to apply to large dataset without sampling. K-Medoid will output a new column with the cluster number, and columns with the distance between each point and the center of each segment. You can easily transform this information into probability.

Parameters:

Method: you can use either PAM or CLARA

ALGO: automatically select number of segments: use the **silhouette** method, defined as follows:

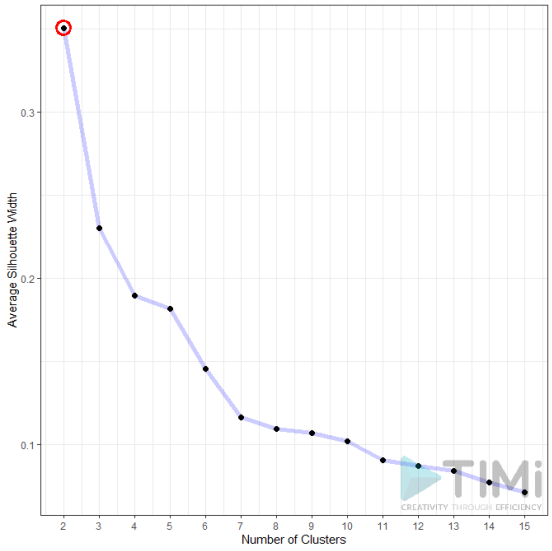
"Put a(i)= average dissimilarity between I and all other points of the cluster to which I belong. (if i is the *only* observation in its cluster, s(i)=0 without further calculations). For all *other* clusters C, put d(i,C) = average dissimilarity of i to all observations of C. The smallest of these d(i,C) is b(i)=minCd(i,C), and can be seen as the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does *not* belong. Finally,

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Observations with a large s(i) (almost 1) are very well clustered, a small s(i) (around 0) means that the observation lies between two clusters, and observations with a negative s(i) are probably placed in the wrong cluster."[2]

[2] https://www.rdocumentation.org/packages/cluster/versions/2.1.0/topics/silhouette

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

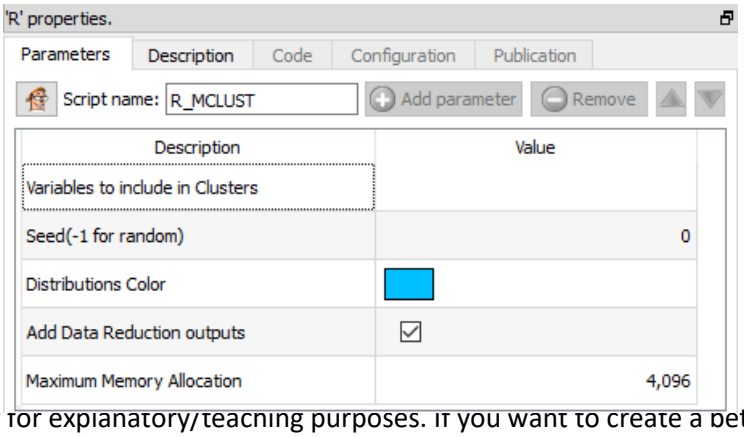


Scale Matrix before clustering: proceed with a normalization of the data to avoid dominance from varaibles on a larger scale.
Distance computation: Select whether you want to use Euclidean (sensitive to outliers) or Manhattan (absolute) distance.
Seed: set a seed number so you can run the same analysis again, with consistent results.
Number of segments: Select the number of segments to keep.
Number of samples: sumber of samples to use in the process. 1 means all the dataset will be used (may be very slow)
Cluster Name: name of the variable with the cluster results.
Include distance from center: include Euclidean distance from centers as new variables.
Plot Results: Select whether or not to display a distribution chart
Chart title: set the title of the chart (if you selected the previous option)
Model Name: Name of the model to use for later scoring.

5.13.5. Model Clustering (R action)

Icon:

Property window:



Short description:
Model Clustering
Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

This algorithm uses BIC to select optimal solution based on a bunch of hypothesis. Very cool, on SMALL dataset: computation time becomes quickly problematic when applied on a few thousand obervervations, and more than 10 variables is hard to Interpret. For this reason, we nearly always compute PCA before hand when using this technique.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is how to interpret the output inside the log window:
“EII”: spherical, equal volume
“VII”: spherical, unequal volume
“EEI”: diagonal, equal volume, equal shape
“VEI”: diagonal, varying volume, equal shape
“EVI”: diagonal, equal volume, varying shape
“VVI”: diagonal, varying volume, varying shape
“EEE”: ellipsoidal, equal volume, shape, and orientation
“EEV”: ellipsoidal, equal volume and equal shape
“VEV”: ellipsoidal, equal shape
“VVV”: ellipsoidal, varying volume, shape, and orientation

MClust gives results consistents with “Latent Class” (see next section 5.12.11). In the following example, we will use the wine dataset available in the datasets directory of Timi.

Chart 1: BIC

The BIC (Bayesian Information Criteria, or Schwartz Criteria) is an extention of Log Likelihood penalizing the number of parameters. In this particular case, it is used to assess the likelihood that a particular structure fits the data better than the others.

$$BIC=\ln(n)k - 2\ln(L).$$

Basically: the closest to 0, the betters (it can be negative or positive).
In this example, we see there is a maximal value for 4 segments, of type VEE: diagonal, varying volume, and equal shape and orientation.

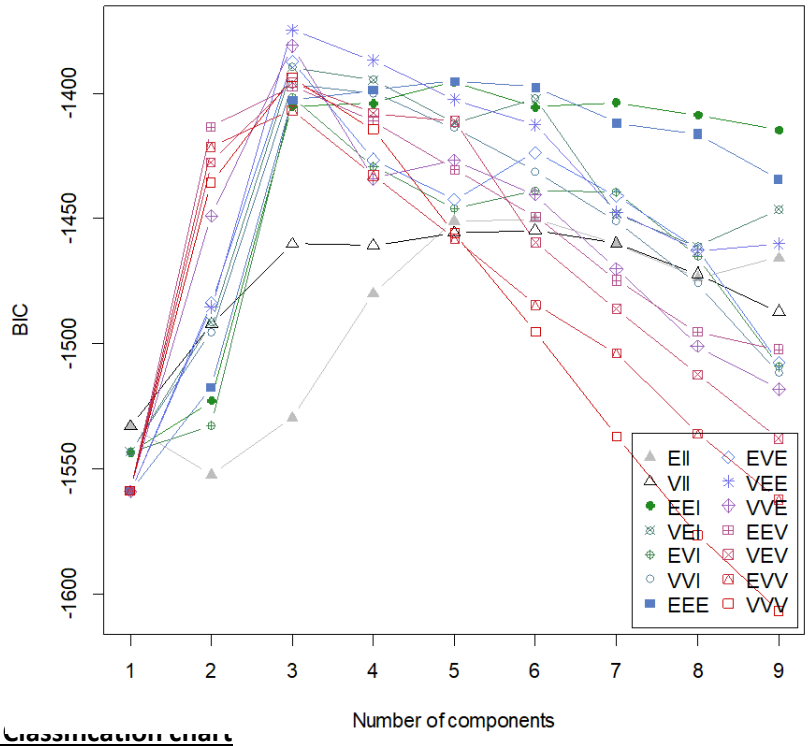


Chart 2: Classification chart

This chart shows a pairwise visualization of the various distributions identified, while plotting each individual point in a color specific to the segment assigned, as well as an estimation of the distribution.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

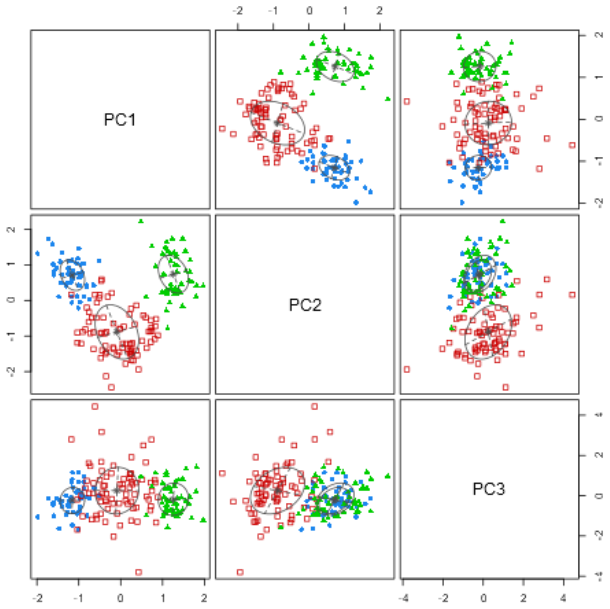


Chart 3: uncertainty

This charts complements the previous one by displaying the points for which there is a high understainty regarding the segment assignation. This helps get a feeling of the risk of mis-assignment of clusters

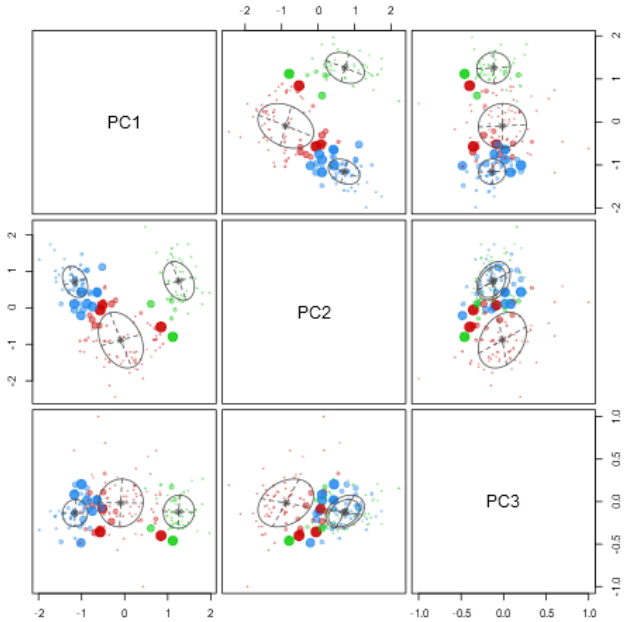
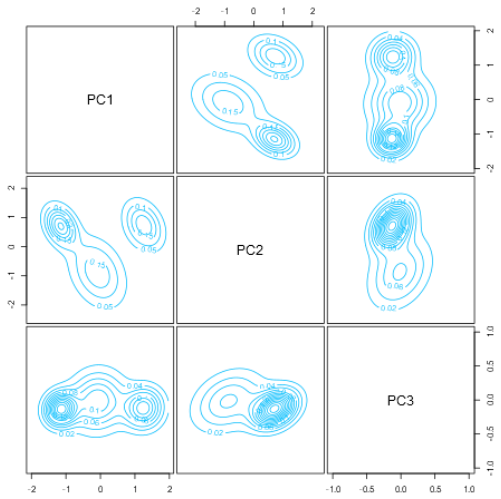


Chart 4: density

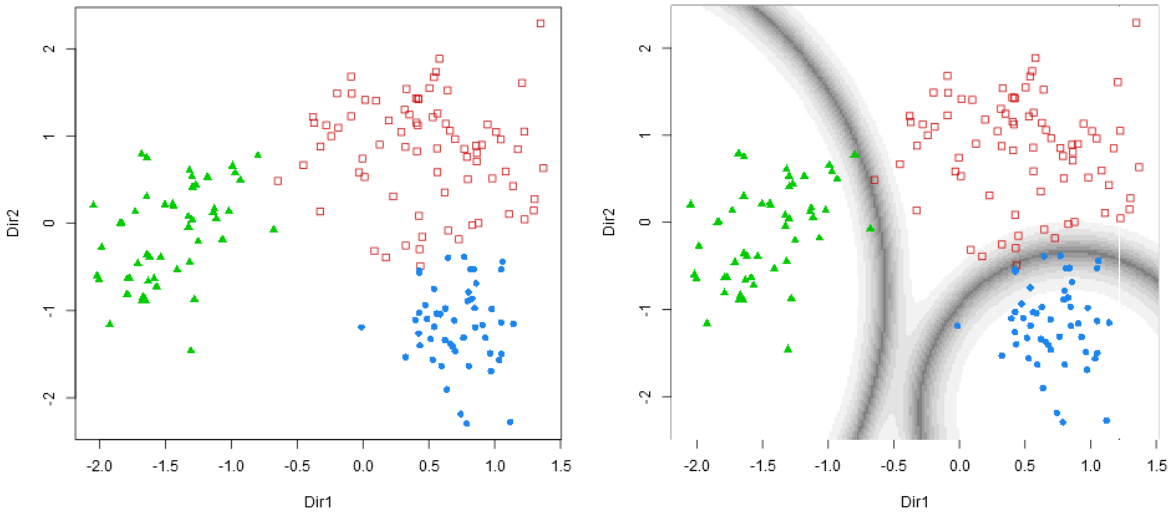
This chart displays the density of the segments and how they are positions in the multivariate space. Each line represents a boundary of the confidence we have that a particular point belongs to the distribution (p)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Charts 5-6: Scatterplot and Boundaries

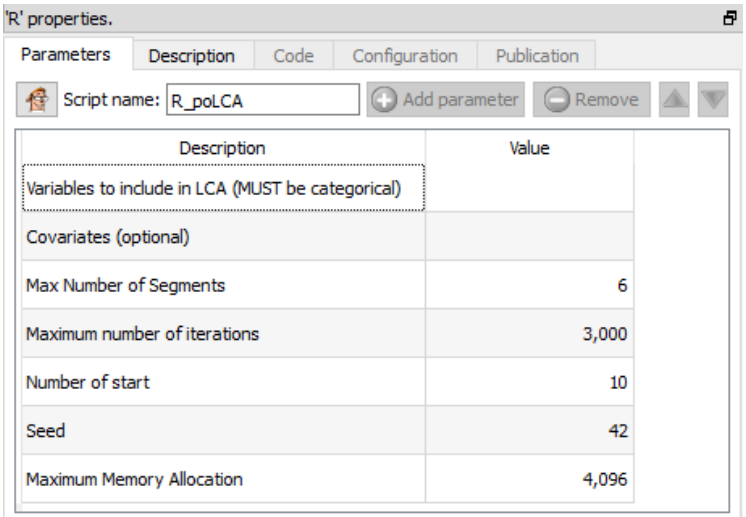
These last two plots are only displayed if the option “Add data reduction outputs” is checked. The two first principal components are displayed with the color codes corresponding to the segments. The boundaries (areas in which misclassification is to be espected) are also displayed.



5.13.6. Latent Class Analysis Clustering (R action)

Icon:

Property window:



Short description:
Latent Class Clustering

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

Variables must be ordinal, with few categories, on TINY dataset!
The code is mosty inspired from http://statistics.ohlsen-web.de/latent-class-analysis-polca/

Latent Class Clustering is a method that allows to group categorical variables more efficiently and has the advantage of providing measures of goodness of fit. For more information about the fundamentals of it, I invite you to read the works of Magidson and Vermunt who developed the method.

The key advantage is that it takes the shape of distributions into account, and mixes them (hence the other name: Finite Mixture Models). It is often presented as a probabilistic extension of K-Means, although it has a completely different objective.
While K-Means attempts to maximize external homogeneity (we want centers as far from each other as possible) and internal homogeneity (we want the observations in our segments to be as similar as possible), Latent Class looks for a latent variable that explains why local independence is not respected.

What is local independence? Let's consider a simple example: users of facebook vs users of LinkedIn. If the two variables were independent, the probability to use one facebook and MySpace should be the product of the two probabilities. Remember this concept: $P(A \cup B) = P(A).P(B)$. In this case, we should have 40%*50%=20% of the population using Facebook and MySpace.

	Facebook	Facebook	Total
Myspace	260	140	400
~~Myspace~~	240	360	600
Total	500	500	1000

Clearly, this is not the case. There has to be something that explains it. This something else is what Latent Class is looking for, an unobserved variable that cuts the population in such a way that our equation holds true.

	LC 1: College education			LC 2: HighShool education		
	Facebook	~~Facebook~~	Total	Facebook	~~Facebook~~	total
Myspace	240	60	300	20	80	100
~~Myspace~~	160	40	200	80	320	400
Total	400	100	500	100	400	500

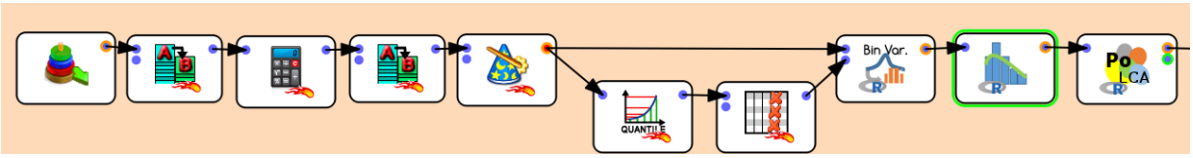
In their publications, Magidson and Vermunt concluded that LC performs as well as DISC, and better than k-means in terms of classification errors when working with known classification dataset. Besides this, Latent Class has many advantages vs K-Means:

- Probability-based classification, which performs better than distance based classification, gives goodness of fit indication (BIC is to be preferred to AIC) and includes indicators of statistically optimal number of clusters
- No need for standardization of variables (scale independent)
- No assumption of **normality** and **homoscedasticity**, it actually allows a combination of mixed scales: continuous, ordinal and nominal variables.
- Local independence (segments are not correlated).
- Covariate based profiling (no further analysis needed for discrimination).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Currently, this technique is not included in Stardust (as is does not work with hundreds of thousands of records).

There are three main software packages that include the implementation of Magidson and Vermunt: Latent Gold (the original one, and in our humble opinion, the best one), SAS, and – of course - R.



First, you need to sample. 2000 is a lot, 5000 rows is big Data, 10.000 rows is hours of computation, 20.000 is probably going to crash. Remember this is distribution based, so designed to work with samples of 20-400 records per segments.
Then, there are a few operations in data transformation that are required before you can run Latent Class

1- Rename variables (remove spaces and special characters)
2- Bin variables: you need to recode your variables to a small amount of levels (ideally 3-5 levels max)

Then, simply select the recoded variables in PoLCA, and click on the SECOND OUTPUT PIN to get a feeling of model quality

Model	og_likelihoo	df	BIC	ABIC	CAIC	elihood_rati	R2_Entropy
Model 1	-8307.65	254	16978.1	16778.3	17041.1	12964.2	0
Model 2	-7926.81	190	16585	16182.2	16712	12202.5	0.9
Model 3	-7747.72	126	16595.4	15989.6	16786.4	11844.3	0.895
Model 4	-7603.29	62	16675.1	15866.3	16930.1	11555.4	0.926
Model 5	-7462.97	-2	16763	15751.2	17082	11274.8	0.907

The most important metrics here is BIC, which should be as small as possible, but with positive degrees of freedom.

In this case, the optimal BIC is a 2-cluster solution, while R2 Entropy suggest an optimal solution at 4 segments. As there are really very little degrees of freedom for this solution, our instinct would be to stay at 2 segments… but still explore all of them,

The first output pin gives us the solution for each model

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

DataTable (317 rows - 64 columns) (complete)

	!Oz Si	ec_Taste Mil	c_Taste Strc	_Taste Very	LC Class 1	LC Class 2	LC Class 3	LC Class 4	LC Class 5
1		2	3	1	1	1	1	4	2
2		1	3	3	1	1	1	2	4
3		2	1	4	1	2	2	3	4
4		2	1	4	1	1	1	4	2
5		3	1	3	1	2	2	3	3
6		1	4	3	1	1	1	2	4
7		3	1	3	1	1	3	1	3
8		3	1	3	1	2	2	3	3

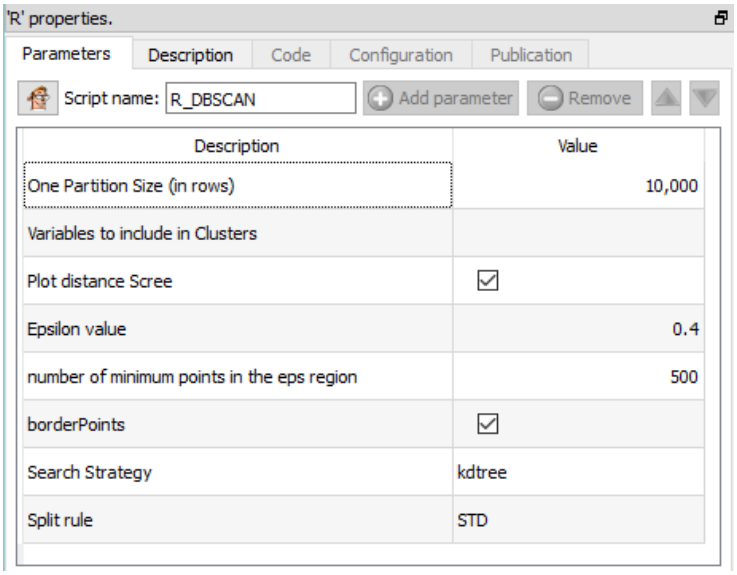
Copy | Copy with Column Names | #Row: | Go

It is therefore quite easy to simply get a few transition tables and study what information is gained by each model.

5.13.7. DBSCAN Clustering (R action)

Icon:

Property window:



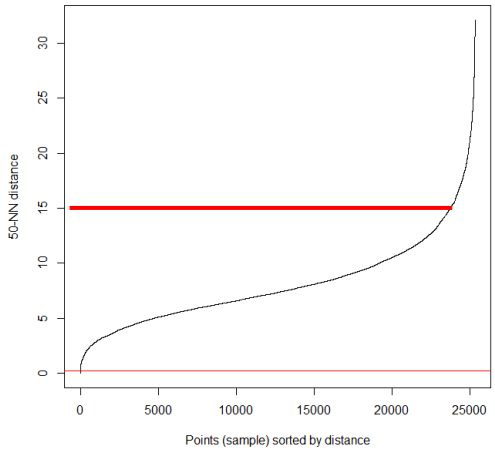
Short description:
DBSCAN Clustering

Long Description:
This Action is mainly for explanatory/teaching purposes. If you want to create a better segmentation, you should use Stardust.

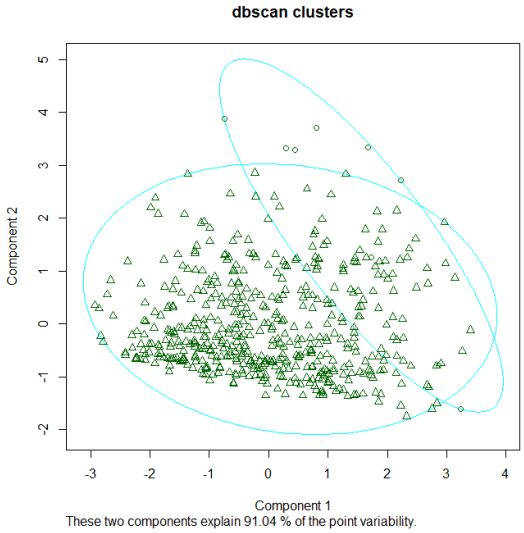
The DBScan algorithm is often used in geographic settings, to understand segments about mobility, for example.

It is performed in two steps. First, run to plot the distance plot, and draw and imaginary line between the last point of inflection, and the vertical axis. This will tell you roughly what the value of Epsilon should be (see red line, in this case, about 15)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

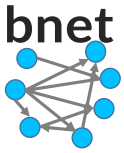


You will then be able to get segment assignment and a plot:

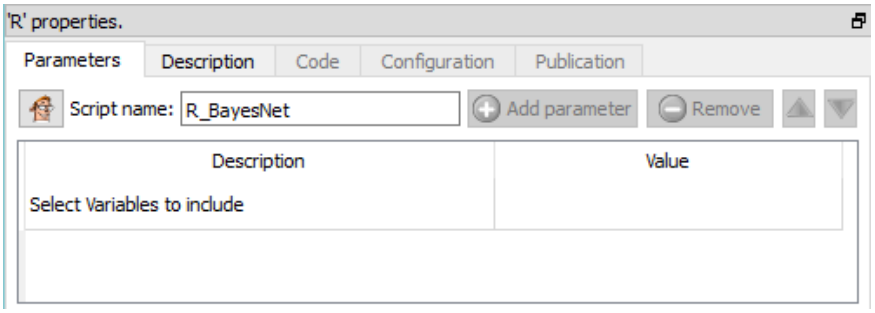


5.13.8. Bayesian Network (action)

Icon:



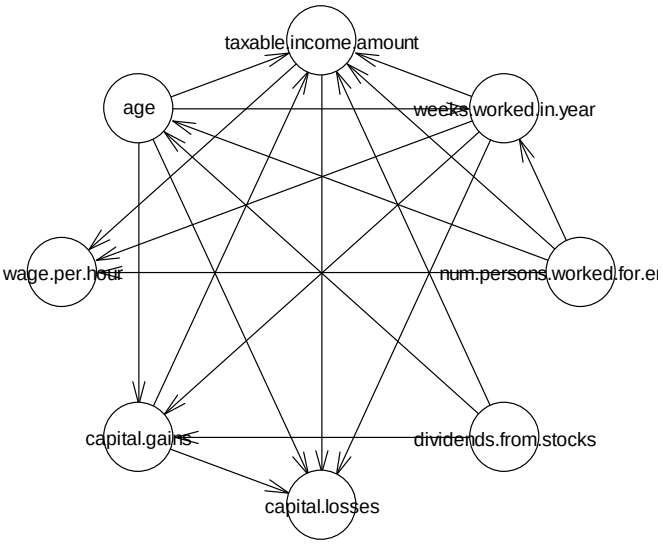
Property window:



Short description:
Built and Display a Naïve Bayes Network

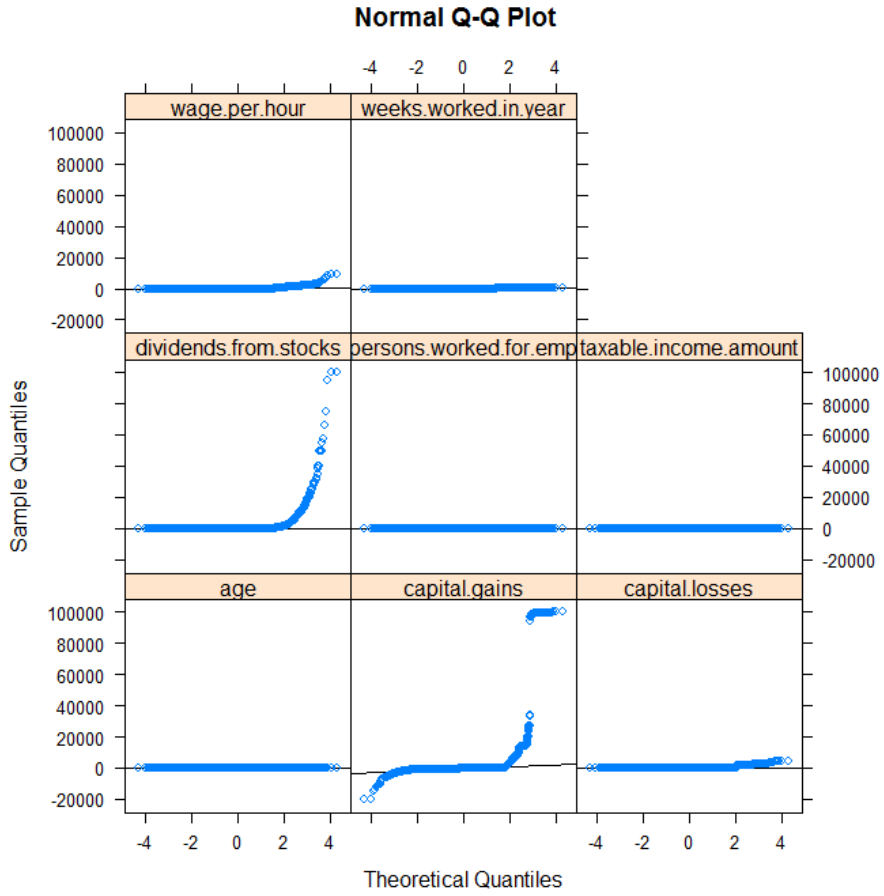
Long Description:
This is useful to see the Causality Network. Bayesian network uses the bnlearn library to extract causality between a set of variables, such as:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



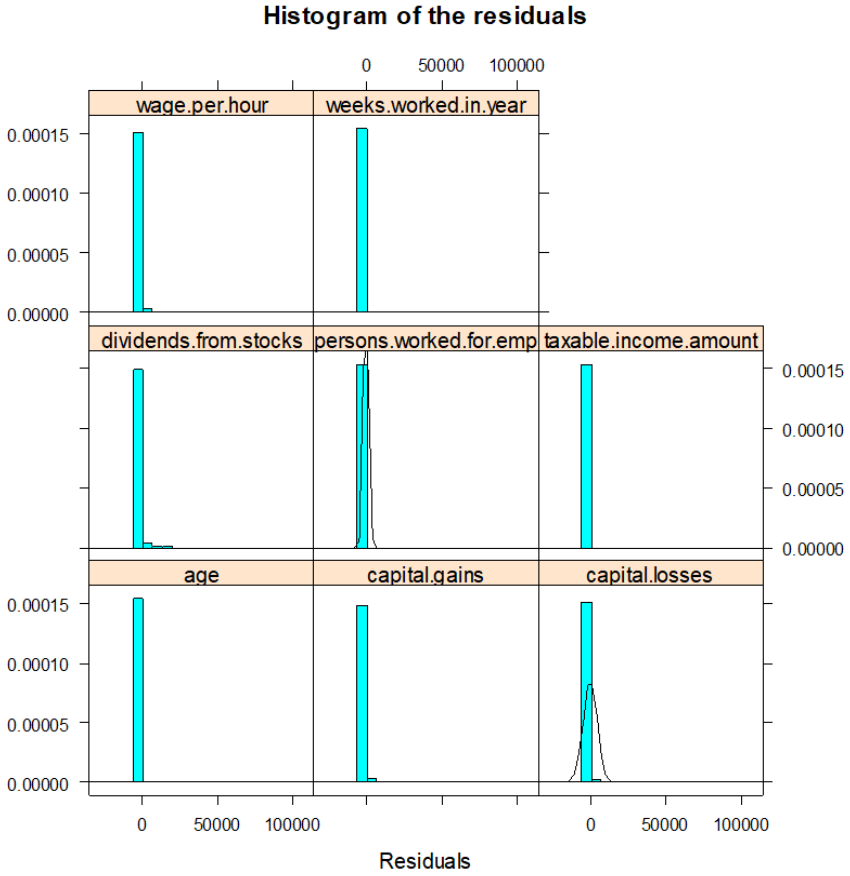
Three additional plots can be generated:

1. Normal QQ Plots

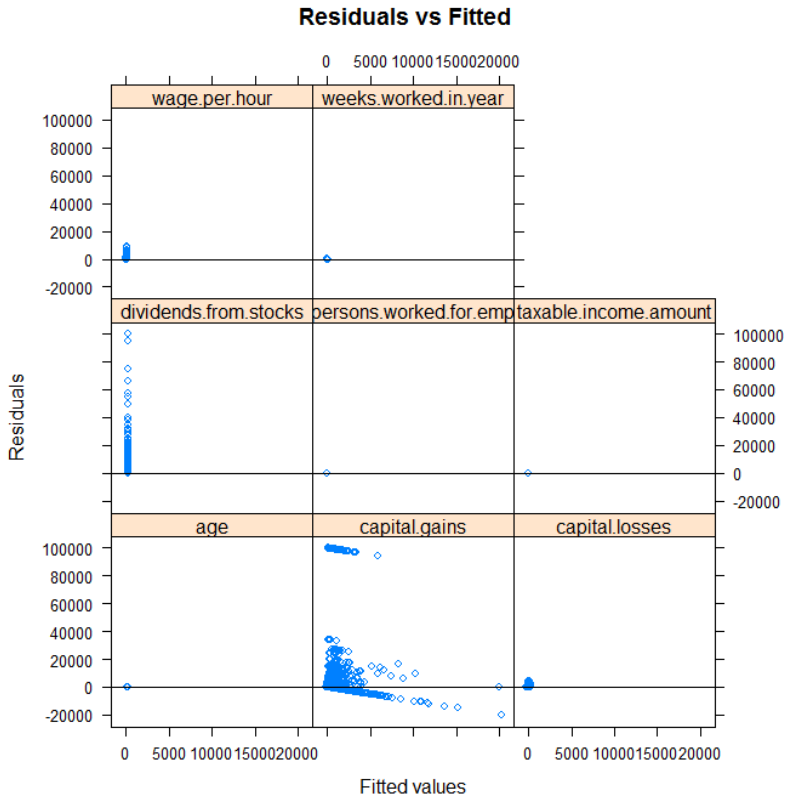


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Variable Histograms



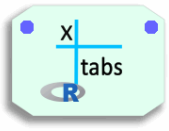
3. X/Y Plots



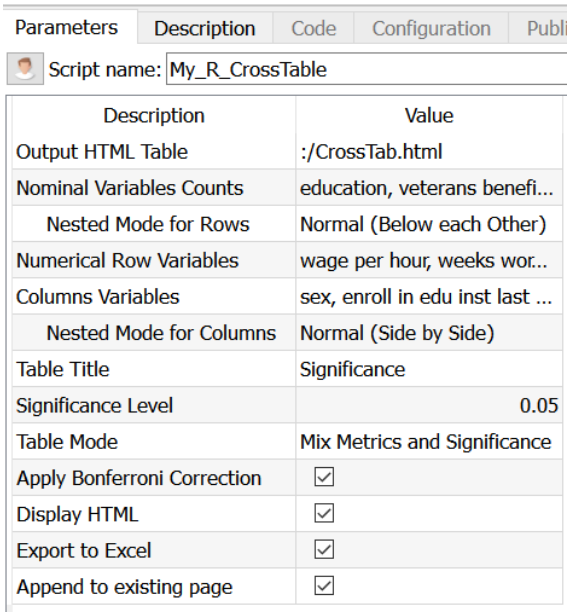
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.13.9. Cross Tabulation (action)

Icon:



Property window:



Description	Value
Output HTML Table	:/CrossTab.html
Nominal Variables Counts	education, veterans benefi...
Nested Mode for Rows	Normal (Below each Other)
Numerical Row Variables	wage per hour, weeks wor...
Columns Variables	sex, enroll in edu inst last ...
Nested Mode for Columns	Normal (Side by Side)
Table Title	Significance
Significance Level	0.05
Table Mode	Mix Metrics and Significance
Apply Bonferroni Correction	☑
Display HTML	☑
Export to Excel	☑
Append to existing page	☑

Short description:
Build a cross tab table for frequencies or means, with or without significance tests. This is equivalent to the Custom Tables from SPSS.

Long Description:
Use this node to generate a table as the following, in which you can control easily which variables are nested:

Significance

	#Total	sex		enroll in edu inst last wk		
		Female	Male	College or university	High school	Not in universe
		A	B	A	B	C
education						
10th grade	3.8	3.9 > B	3.6 < A	0.4 < B C	34.2 > A C	2.8 > A < B
11th grade	3.4	3.5	3.4	0.8 < B C	32.2 > A C	2.5 > A < B
12th grade no diploma	1.1	1.0 < B	1.2 > A	1.0 < B	6.8 > A C	0.9 < B
1st 2nd 3rd or 4th grade	0.9	0.9	0.9		0.1 < C	1.0 > B
5th or 6th grade	1.6	1.6	1.7	0.0 < B C	0.3 > A < C	1.7 > A B
7th and 8th grade	4.0	4.0	4.0	0.1 < B C	3.7 > A	4.1 > A
9th grade	3.1	3.2	3.1	0.2 < B C	19.8 > A C	2.6 > A < B
Associates degree-academic program	2.2	2.5 > B	1.9 < A	3.1 > B C	0.0 < A C	2.2 > B < A
Associates degree-occup /vocational	2.7	3.0 > B	2.4 < A	1.8 > B < C	0.0 < A C	2.8 > A B
Bachelors degree(BA AB BS)	10.0	9.5 < B	10.4 > A	5.7 > B < C	0.1 < A C	10.4 > A B
Children	23.8	22.5 < B	25.2 > A		0.0 < C	25.4 > B
Doctorate degree(PhD EdD)	0.6	0.3 < B	0.9 > A	0.0 < C		0.7 > A
High school graduate	24.3	25.9 > B	22.5 < A	13.4 > B < C	1.6 < A C	25.4 > A B
Less than 1st grade	0.4	0.4	0.4		0.1 < C	0.4 > B
Masters degree(MA MS MEng MEd MSW MBA)	3.3	3.0 < B	3.5 > A	0.4 < C		3.5 > A
Prof school degree (MD DDS DVM LLB JD)	0.9	0.5 < B	1.3 > A	0.0 < C		1.0 > A
Some college but no degree	13.9	14.4 > B	13.5 < A	73.0 > B C	1.1 < A C	12.6 > B < A
veterans benefits code						
0	23.8	22.4 < B	25.2 > A			25.4
1	1.0	0.4 < B	1.6 > A	0.5 > B < C	0.0 < A C	1.0 > A B
2	75.2	77.2 > B	73.2 < A	99.5 > C < B	100.0 > A C	73.6 < A B
wage per hour						
Mean	55.4	49.3 < B	62.1 > A	67.5 > B C	36.0 < A C	55.8 > B < A
weeks worked in year						
Mean	23.2	20.6 < B	26.0 > A	26.3 > B C	11.6 < A C	23.5 > B < A

The output can be any combination of

- An anatella table
- An HTML Table
- An Excel file

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Parameters:

Output HTML Table: set name of the html file containing the table (can be existing, with the Append option)
Nominal Row Variable : select the nominal vbariables to compare
Nested Mode for Rows: select if you want to set dependencies between your nominal variables
Numerical Row Variables: select the variables on which you wish to compare means
Column Variables: select the variables on which you wish to perform the "Group By"
Nested mode for Columns:
Table Title: name of the table in the HTML file, and the TAB in the Excel file
Significance LeveL default is 0.05
Table Mode:

- **Mix Metrics and Significance**: generates a single table in which the significance is marked next to the numeric metric
- **Separate Significance**: generate two tables, one with only the metric, and one combined
- **No Significance**: only show the metric

Apply Bonferroni Correction: self explanatory
Display HTML: display the HTML file on completion
Export to Excel: Export table to an excel file, in the same directory as the HTML File
Append to existing file: add the table to the current HTML File, and the Excel as a new Tab

Examples:

Basic Table

	#Total	sex		enroll in edu inst last wk		
		Female	Male	College or university	High school	Not in universe
veterans benefits code						
#number of cases	199523	103984	95539	5688	6892	186943
#row %	100	52.1	47.9	2.9	3.5	93.7
0	23.8	22.4	25.2			25.4
1	1.0	0.4	1.6	0.5	0.0	1.0
2	75.2	77.2	73.2	99.5	100.0	73.6
education						
#number of cases	199523	103984	95539	5688	6892	186943
#row %	100	52.1	47.9	2.9	3.5	93.7
10th grade	3.8	3.9	3.6	0.4	34.2	2.8
11th grade	3.4	3.5	3.4	0.8	32.2	2.5
12th grade no diploma	1.1	1.0	1.2	1.0	6.8	0.9
1st 2nd 3rd or 4th grade	0.9	0.9	0.9		0.1	1.0
5th or 6th grade	1.6	1.6	1.7	0.0	0.3	1.7
7th and 8th grade	4.0	4.0	4.0	0.1	3.7	4.1
9th grade	3.1	3.2	3.1	0.2	19.8	2.6
Associates degree-academic program	2.2	2.5	1.9	3.1	0.0	2.2
Associates degree-occup /vocational	2.7	3.0	2.4	1.8	0.0	2.8
Bachelors degree(BA AB BS)	10.0	9.5	10.4	5.7	0.1	10.4
Children	23.8	22.5	25.2		0.0	25.4
Doctorate degree(PhD EdD)	0.6	0.3	0.9	0.0		0.7
High school graduate	24.3	25.9	22.5	13.4	1.6	25.4
Less than 1st grade	0.4	0.4	0.4		0.1	0.4
Masters degree(MA MS MEng MEd MSW MBA)	3.3	3.0	3.5	0.4		3.5
Prof school degree (MD DDS DVM LLB JD)	0.9	0.5	1.3	0.0		1.0
Some college but no degree	13.9	14.4	13.5	73.0	1.1	12.6
wage per hour						
Mean	55.4	49.3	62.1	67.5	36.0	55.8
Std. dev.	274.9	246.9	302.3	218.4	154.1	279.8
Unw. valid N	199523.0	103984.0	95539.0	5688.0	6892.0	186943.0
weeks worked in year						
Mean	23.2	20.6	26.0	26.3	11.6	23.5
Std. dev.	24.4	23.9	24.6	20.9	17.2	24.6
Unw. valid N	199523.0	103984.0	95539.0	5688.0	6892.0	186943.0

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Mix Metrics

	#Total	sex		enroll in edu inst last wk		
		Female	Male	College or university	High school	Not in universe
		A	B	A	B	C
veterans benefits code						
0	23.8	22.4 < B	25.2 > A			25.4
1	1.0	0.4 < B	1.6 > A	0.5 > B < C	0.0 < A C	1.0 > A B
2	75.2	77.2 > B	73.2 < A	99.5 > C < B	100.0 > A C	73.6 < A B
education						
10th grade	3.8	3.9 > B	3.6 < A	0.4 < B C	34.2 > A C	2.8 > A < B
11th grade	3.4	3.5	3.4	0.8 < B C	32.2 > A C	2.5 > A < B
12th grade no diploma	1.1	1.0 < B	1.2 > A	1.0 < B	6.8 > A C	0.9 < B
1st 2nd 3rd or 4th grade	0.9	0.9	0.9		0.1 < C	1.0 > B
5th or 6th grade	1.6	1.6	1.7	0.0 < B C	0.3 > A < C	1.7 > A B
7th and 8th grade	4.0	4.0	4.0	0.1 < B C	3.7 > A	4.1 > A
9th grade	3.1	3.2	3.1	0.2 < B C	19.8 > A C	2.6 > A < B
Associates degree-academic program	2.2	2.5 > B	1.9 < A	3.1 > B C	0.0 < A C	2.2 > B < A
Associates degree-occup /vocational	2.7	3.0 > B	2.4 < A	1.8 > B < C	0.0 < A C	2.8 > A B
Bachelors degree(BA AB BS)	10.0	9.5 < B	10.4 > A	5.7 > B < C	0.1 < A C	10.4 > A B
Children	23.8	22.5 < B	25.2 > A		0.0 < C	25.4 > B
Doctorate degree(PhD EdD)	0.6	0.3 < B	0.9 > A	0.0 < C		0.7 > A
High school graduate	24.3	25.9 > B	22.5 < A	13.4 > B < C	1.6 < A C	25.4 > A B
Less than 1st grade	0.4	0.4	0.4		0.1 < C	0.4 > B
Masters degree(MA MS MEng MEd MSW MBA)	3.3	3.0 < B	3.5 > A	0.4 < C		3.5 > A
Prof school degree (MD DDS DVM LLB JD)	0.9	0.5 < B	1.3 > A	0.0 < C		1.0 > A
Some college but no degree	13.9	14.4 > B	13.5 < A	73.0 > B C	1.1 < A C	12.6 > B < A
wage per hour						
Mean	55.4	49.3 < B	62.1 > A	67.5 > B C	36.0 < A C	55.8 > B < A
weeks worked in year						
Mean	23.2	20.6 < B	26.0 > A	26.3 > B C	11.6 < A C	23.5 > B < A

Nested Table

	#Total	sex		enroll in edu inst last wk	
		Female		Not in universe	
		sex	enroll in edu inst last wk	sex	enroll in edu inst last wk
		Male	Not in universe	Male	Not in universe
		A	B	A	B
education					
10th grade	3.8	3.8	3.8	3.8	3.8
11th grade	3.4	3.4	3.4	3.4	3.4
12th grade no diploma	1.1	1.1	1.1	1.1	1.1
1st 2nd 3rd or 4th grade	0.9	0.9	0.9	0.9	0.9
5th or 6th grade	1.6	1.6	1.6	1.6	1.6
7th and 8th grade	4.0	4.0	4.0	4.0	4.0
9th grade	3.1	3.1	3.1	3.1	3.1
Associates degree-academic program	2.2	2.2	2.2	2.2	2.2
Associates degree-occup /vocational	2.7	2.7	2.7	2.7	2.7
Bachelors degree(BA AB BS)	10.0	10.0	10.0	10.0	10.0
Children	23.8	23.8	23.8	23.8	23.8
Doctorate degree(PhD EdD)	0.6	0.6	0.6	0.6	0.6
High school graduate	24.3	24.3	24.3	24.3	24.3
Less than 1st grade	0.4	0.4	0.4	0.4	0.4
Masters degree(MA MS MEng MEd MSW MBA)	3.3	3.3	3.3	3.3	3.3
Prof school degree (MD DDS DVM LLB JD)	0.9	0.9	0.9	0.9	0.9
Some college but no degree	13.9	13.9	13.9	13.9	13.9
veterans benefits code					
0	23.8	23.8	23.8	23.8	23.8
1	1.0	1.0	1.0	1.0	1.0
2	75.2	75.2	75.2	75.2	75.2
wage per hour					
Mean	55.4	55.4	55.4	55.4	55.4
weeks worked in year					
Mean	23.2	23.2	23.2	23.2	23.2

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Nested Table Separate Significance

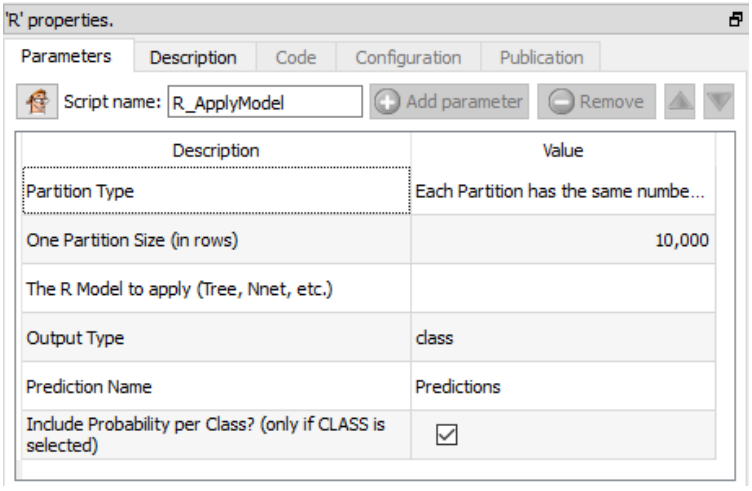
	#Total	sex		enroll in edu inst last wk		sex		enroll in edu inst last wk	
		Female		Not in universe		Female		Not in universe	
		sex	enroll in edu inst last wk	sex	enroll in edu inst last wk	sex	enroll in edu inst last wk	sex	enroll in edu inst last wk
		Male	Not in universe	Male	Not in universe	Male	Not in universe	Male	Not in universe
						A	B	A	B
veterans benefits code									
#number of cases	199523	199523	199523	199523	199523				
#row %	100	100	100	100	100				
0	23.7611703913834	23.8	23.8	23.8	23.8				
1	0.994371576209259	1.0	1.0	1.0	1.0				
2	75.2444580324073	75.2	75.2	75.2	75.2				
education									
#number of cases	199523	199523	199523	199523	199523				
#row %	100	100	100	100	100				
10th grade	3.78753326684142	3.8	3.8	3.8	3.8				
11th grade	3.4462192328704	3.4	3.4	3.4	3.4				
12th grade no diploma	1.06554131603875	1.1	1.1	1.1	1.1				
1st 2nd 3rd or 4th grade	0.901650436290553	0.9	0.9	0.9	0.9				
5th or 6th grade	1.64241716493838	1.6	1.6	1.6	1.6				
7th and 8th grade	4.01307117475178	4.0	4.0	4.0	4.0				
9th grade	3.12244703618129	3.1	3.1	3.1	3.1				
Associates degree-academic program	2.18671531602873	2.2	2.2	2.2	2.2				
Associates degree-occup /vocational	2.68540469018609	2.7	2.7	2.7	2.7				
Bachelors degree(BA AB BS)	9.95624564586539	10.0	10.0	10.0	10.0				
Children	23.7676859309453	23.8	23.8	23.8	23.8				
Doctorate degree(PhD EdD)	0.633009728201761	0.6	0.6	0.6	0.6				
High school graduate	24.2613633515936	24.3	24.3	24.3	24.3				
Less than 1st grade	0.410478992396867	0.4	0.4	0.4	0.4				
Masters degree(MA MS MEng MEd MSW MBA)	3.2783187903149	3.3	3.3	3.3	3.3				
Prof school degree (MD DDS DVM LLB JD)	0.898643264185081	0.9	0.9	0.9	0.9				
Some college but no degree	13.9432546623698	13.9	13.9	13.9	13.9				
veterans benefits code									
0	23.8					23.8	23.8	23.8	23.8
1	1.0					1.0	1.0	1.0	1.0
2	75.2					75.2	75.2	75.2	75.2
education									
10th grade	3.8					3.8	3.8	3.8	3.8
11th grade	3.4					3.4	3.4	3.4	3.4
12th grade no diploma	1.1					1.1	1.1	1.1	1.1
1st 2nd 3rd or 4th grade	0.9					0.9	0.9	0.9	0.9
5th or 6th grade	1.6					1.6	1.6	1.6	1.6
7th and 8th grade	4.0					4.0	4.0	4.0	4.0
9th grade	3.1					3.1	3.1	3.1	3.1
Associates degree-academic program	2.2					2.2	2.2	2.2	2.2
Associates degree-occup /vocational	2.7					2.7	2.7	2.7	2.7
Bachelors degree(BA AB BS)	10.0					10.0	10.0	10.0	10.0
Children	23.8					23.8	23.8	23.8	23.8
Doctorate degree(PhD EdD)	0.6					0.6	0.6	0.6	0.6
High school graduate	24.3					24.3	24.3	24.3	24.3
Less than 1st grade	0.4					0.4	0.4	0.4	0.4
Masters degree(MA MS MEng MEd MSW MBA)	3.3					3.3	3.3	3.3	3.3
Prof school degree (MD DDS DVM LLB JD)	0.9					0.9	0.9	0.9	0.9
Some college but no degree	13.9					13.9	13.9	13.9	13.9
wage per hour									
Mean	55.4269081759998	55.4	55.4	55.4	55.4				
Std. dev.	274.896453902842	274.9	274.9	274.9	274.9				
Unw. valid N	199523	199523.0	199523.0	199523.0	199523.0				
weeks worked in year									
Mean	23.1748971296542	23.2	23.2	23.2	23.2				
Std. dev.	24.4114881675042	24.4	24.4	24.4	24.4				
Unw. valid N	199523	199523.0	199523.0	199523.0	199523.0				
wage per hour									
Mean	55.4					55.4	55.4	55.4	55.4
weeks worked in year									
Mean	23.2					23.2	23.2	23.2	23.2

5.14. TA – R Scoring (TA=Transformations Actions)

5.14.1. Apply a R-Based predictive model (R action)

Icon:

Property window:

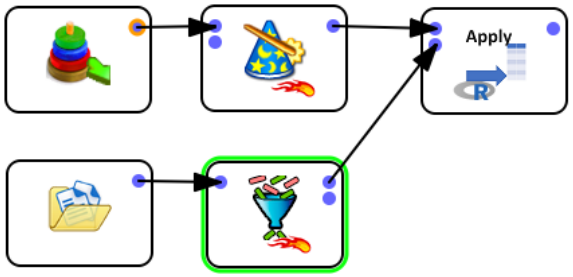


Short description:
Apply a R-Based model

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action works with any partition type. This means that, using this Action, you can score any dataset size, including the datasets that are larger than your central RAM memory. This allow to by-pass the limitations on the RAM memory that all other tools that are using R as back-end exhibits. In this latest release, the objective of the model is saved in the file, and it is possible to open many models at the same time. In a single operation, you can score dozens of models of different types.
Parameters:

- Partition: set partition or partition variable if you have a few 100 thousands or millions of rows.
- Key Variable: Unique identifyer of each row
- R Model(s) to apply: select the models manually. This is not necessary if you use the input pin, which is recommended, as in the example below, in which we simply list *.rModel, filter those that we want to use, and apply on the new dataset.

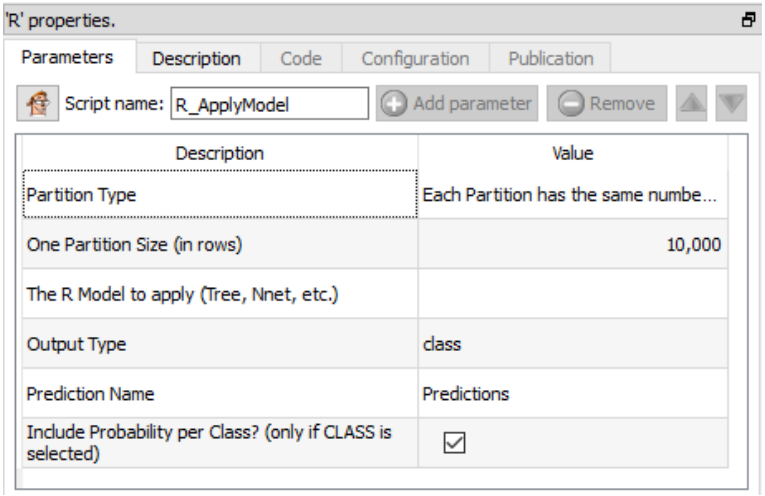


- Include probability per class: Some algorithms allow returning probability additional to class. Select this option to return it.

5.14.2. Apply a R-Based PCA model (action)

Icon:

Property window:



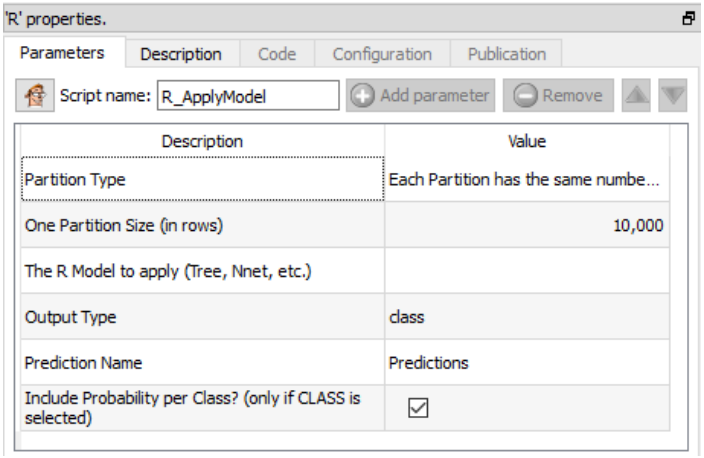
Short description:
Apply a R-Based PCA model

Long Description:
This Action works with any partition type. This means that, using this Action, you can score any dataset size, including the datasets that are larger than your central RAM memory. This allow to by-pass the limitations on the RAM memory that all other tools that are using R as back-end exhibits.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.14.3. Apply a R-Based Segmentation model (action)

Icon:



Property window:

Short description:
Apply a R-Based Segmentation model

Long Description:
This Action works with any partition type. This means that, using this Action, you can score any dataset size, including the datasets that are larger than your central RAM memory. This allow to by-pass the limitations on the RAM memory that all other tools that are using R as back-end exhibits.

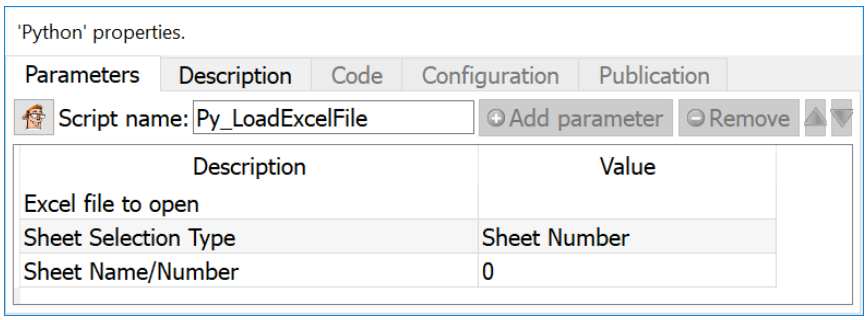
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.15. TA - Python Various (TA=Transformations Actions)

5.15.1. PyLoad Excel File (Python action)

Icon:

Property window:



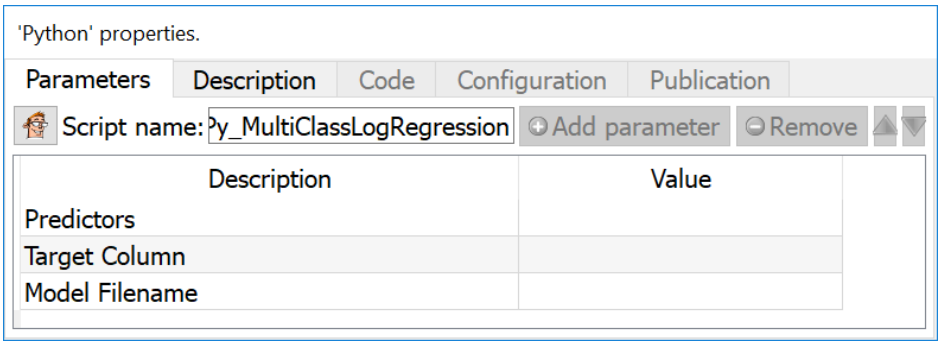
Short description:
Open old excel files

Long Description:
Load excel files of a pre-Excel 2000 format. While it is a good idea to automatically convert those files to a later format using the Upgrade to XLSX action (see section 5.21.11), sometimes you may just want to open a single file. When this happens, and you have Python Anaconda installed, then this node is your friend.

5.15.2. Multiclass Logistic Regression (Python action)

Icon:

Property window:



Short description:
Creates a multinomial logit predictive model

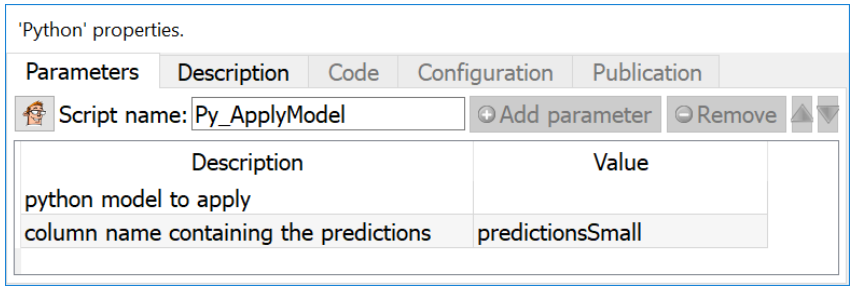
Long Description:
When you have a large number of variables (a few hundreds), all R libraries dealing with MNL modeling tend to fall apart as they first discretize every variable and need to run a full model before step-wise approaches are used.

In this very particular case, Python's library is better implemented. This node will fit the model.

5.15.3. PyApply Model (Python action)

Icon:

Property window:



Short description:
Apply a Python model created with Anatella

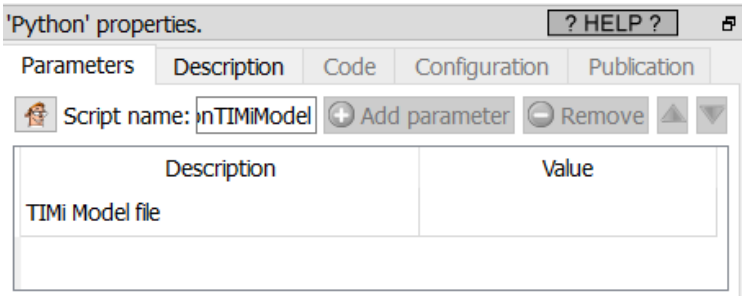
Long Description:
Apply a python model to score a new dataset. You can create this model using the MultiClassLogRegression action described in the previous section (5.15.2).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.15.4. PyApply Timi Model (Python action)

Icon:

Property window:



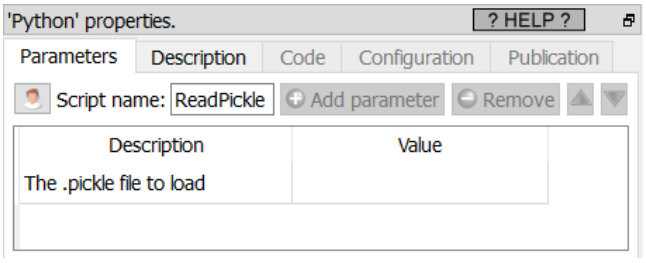
Short description:
Apply a Timi model in Python code

Long Description:
Apply a timi model in Python code. This is roughly 40 times slower than the regular apply model in TIMi from the Data Mining section. But if you really want to use python to score your models, go ahead, slow data mining is also good. Step 1: be slower, Step 2: make profit.

5.15.5. Read Pickle (Python action)

Icon:

Property window:



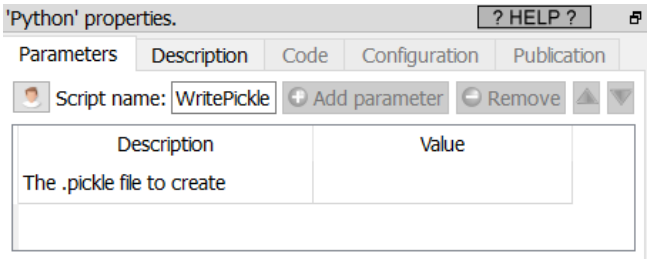
Short description:
Load a .pickle file

Long Description: self-explanatory

5.15.6. Write Pickle (Python action)

Icon:

Property window:



Short description:
Write a table to a .pickle file

Long Description: self-explanatory

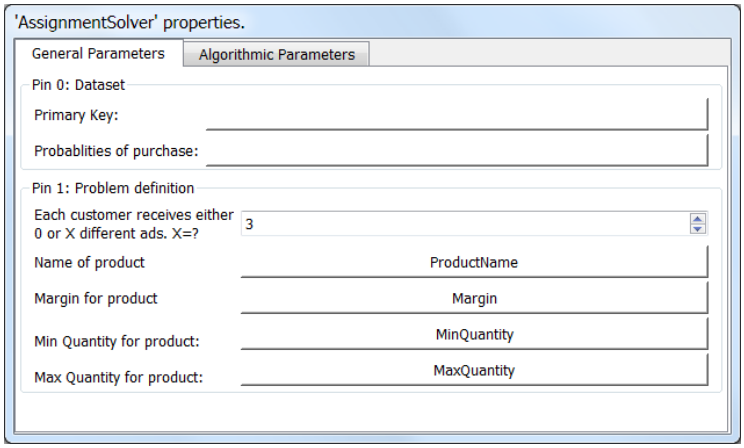
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

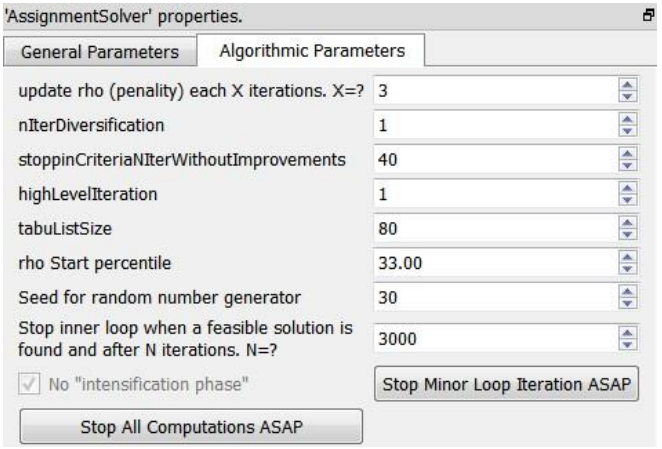
5.16. TA – Operational Research (TA=Transformations Actions)

5.16.1. Assignment Solver (High-Speed action)

Icon:

Property window:





Short description:
Solve the constrained assignement problem.

Long Description:
The constrained assignment problem is a very classical problem that occurs very often in direct marketing. It's also very common in retail. The typical usage is the following:

- The setting: You have a limited number of "promotional coupons" for 100 different products. Each of these coupons is offering a reduction on a specific product. Each of these coupons is in limited quantity because they are given by the corresponding product supplier in limited quantity. You can think about these 100 products as the "*highlighted products of the week*".
- The objective: Use the promotional coupons to attract some customers into your store to have the opportunity to sell them some more products.
- How? Each of your customers will receive 5 coupons (out of a total number of 100 different coupons associated to 100 different products). To redeem the coupons your customer need to come into your store. We'll select the 5 coupons that are the "*most-likely to be redeemed*". These 5 coupons are different for each customer. These 5 coupons are selected using predictive modeling:
  Each of the coupons corresponds to a specific product. We'll build 100 predictive models that compute the probability of purchase for each of the 100 products. We then "apply" these 100 models to get, for each cutomers, 100 purchase probabilities. In the simple case (when there are no constraints), the 5 coupons (that a specific customer receives) correspond to the top-5 probabilities out of the 100 probabilities that are computed for each customer (using our 100 predictive models).
- The difficulty: There are constraints: You cannot simply assign to a customer the products with the highest purchase-probabilities (and totally disregard the limitation on the quantity of each product). When such a constraint (on the quantity of each product) exists, the assignment of

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

a specific product to a specific customer must be computed using a more complex mechanism: This is the objective of the AssignmentSolver Action.

The classical methodology used to optimally assign coupons to your customers is:

- For each different product (or "coupon"), create one predictive model.
- Apply all the predictive models on the customer database to obtain, "*For each customer, the probablity of purchase of each product*".
- Assign coupons to each customers so that:
  - The Expect Return (ER=ROI) of the campaign is maximized
    (i.e. The customers are receiving the coupons related to the products that they are the most likely to buy).
  - The constrained are respected: These are:
    - The amount of coupons of each product is limited to a given threshold.
    - Each customer receives exactly ***n*** coupons.
      (i.e. one folder contains exactly ***n*** coupons for ***n*** different products).

This procedure assigns a different set of products (or "coupons") to each customer. Potentially, there are as many different folders as there are customers (To print these folders you need a "digital printer" because this is the only technology that allows you to print a different folder for each customer).

There exist two types of assignment problems:

1. **Mono-product assignment problems:** You need to select ONE product per customer, taking into account the constraints.

   These type of problems can easily be solved with any number of *highlighted* products and any number of customers: For example:

   For each different customer, propose 1 product amongst 9 different products.

   The Mono-product solver uses the concept of "meta-customer". A "meta-customer" is a mini-segment of customers computed using Stardust. Typically, we'll have 300 to 1500 differents segments. This type of assignment problem is easily solved using any LP solver (Anatella has a very high-quality multi-threaded LP solver).
2. **Multi-product assignment problems:** You need to select ***n>1*** products per customer, taking into account the constraints (n is the number of coupons inside a folder).

   These type of problems are extremely difficult to solve because they belong to the class of problem known as "NP-hard" problems. For example:

   For each different customer, propose 3 products amongst 9 different products.
   For each different customer, propose 8 products amongst 450 different products.

   Because these problems are "NP-hard", it's not possible to solve them exactly. The AssignmentSolver Action uses advanced meta-heuristics to find a "good-enough" solution in a reasonable amount of time.

About "NP-Hard" problems.
Typically, to solve a "normal" numerical problem (i.e. not a NP-hard problem), a good idea is to "split" the problem into simpler problems, then solve each simpler problem separately and thereafter combine all these intermediate results into the "final" solution.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This type of approach is not possible with "NP-Hard" problems: They are not decomposable. The only way to exactly solve "NP-Hard" problems is to enumerate all possible solutions and to give, as the "final" solution, the best admissible solution found during the enumeration.

Let's now evaluate how much time is required to enumerate all possible solutions to the multi-product assignment problem. Let's assume that, inside our assignment problem, there are *c* customers and *p* products. One solution of the assignment problem is a matrix *X*. Each element X_{ij} of the solution-matrix *X* is, either:

"X_{ij}=1": the customer *i* received a coupon about product *j*.
"X_{ij}=0": the customer *i* did not receive a coupon about product *j*.

We need to enumerate all the different solution-matrices *X*.
There are $2^{c \times p}$ different solution-matrices *X*.
For a normal-size problem, we have: c=1 million customers, p=100 products, the number of different solutions: $2^{100 \text{ million}}$= a very very large number.
Even on the best possible hardware, it will take several times the age of the universe to enumerate all the solutions.

Since it's not possible to solve exactly the multi-product assignement problem (because it's not possible to enumerate all different solutions), we'll only try to find a "good enough" solution: We'll enumerate only a (very) limited number of good "candidate solutions". These "candidate solutions" are generated using a meta-heuristic named "tabu-search". The solution that is returned by the AssignmentSolver Action is the best admissible solution found during this "limited" enumeration.

"Admissible solution" means a solution that respects the constraints (i.e. The constraints are: We have some limitations on the quantity of each coupon/product).
"Best solution" means a solution that has the highest ROI/the highest ER.

The quality of the returned solution mainly depends on the running time: If you run the "tabu-search" procedure for a very-long time, you'll be able to test a large number of different "candidate solutions" and you'll be more likely to find a very good solution (To summarize: the longer you compute, the better the "returned" solution).

The different parameters of the AssignmentSolver Action control are:

1. the running-time (i.e. these are parameters used as "*stopping criterias*")
2. the way the "candidate solutions" are generated using the "tabu-search" procedure.

The solution returned by the AssignmentSolver Action is a table that contains the primary keys of each customers and the solution-matrix *X*. Each element X_{ij} of the solution-matrix *X* is, either:

"X_{ij}=1": the customer *i* received a coupon about product *j*.
"X_{ij}=0": the customer *i* did not receive a coupon about product *j*.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

More precisely, inside the solution table, we have:

1. Each row is a customer.
2. The columns are:
   - The primary key of the customer.
   - The different products assigned to the customer: These are *p* different columns (*p*=number of highlighted products) that contains either "0" (i.e. the product is not assigned to this customer) or "1" (i.e. the product is assigned to the current customer).

In mathematical terms, the best solution is the solution that cumulates the greatest purchase probabilities for all the assigned products to all the customers: It's:

Best solution $X^*_{i,j}$ is the one that solves: $\max_{X_{i,j}} \sum_{i,j} PurchaseProbability_{i,j} \times X_{i,j}$

With *i=1,…,c* : all the customers (there are *c* different customers in total)

j=1,…,p : all the "*highlighted products of the week*" (there are *p* different highlighted products in total).

$PurchaseProbability_{i,j}$: this is typically computed using 100 predictive models (one model for each different highlighted product).

Furthermore, this solution must respect these 2 constraints:

1. The number of coupons per product is limited (this is a user-specified business constraint):
$$\sum_i x_{i,j} < ProductMax_j, \forall j$$

2. Each customer receives **5** ads (this is thus a **multi**-product assignment problem and not a **mono**-product assignment problem because 5>1)(one folder contains 5 coupons):
$$\sum_j x_{i,j} = 5, \forall i$$

Let's now assume that you get a small margin each time a customer uses a coupon. The "*expected profit*" when a customer *i* purchases a product *j* is thus: $ProductMargin_j.PurchaseProbability_{i,j}$

The table returned by the AssignmentSolver Action is the solution X_{ij} to the following optimization problem:

Maximum Expected Return: $ER = \max_{x_{i,j}, y_i} \sum_{i,j} ProductMargin_j.PurchaseProbability_{i,j}.x_{i,j}$

contraints: $\sum_j x_{i,j} = 5y_i \quad , \forall i$

$\sum_i x_{i,j} < ProductMax_j, \forall j$

$i = 1,...,nCustomers$

$j = 1,....,nProducts$

$x_{i,j} = \{0,1\}$: customer i receives one ad about product j

$y_i = \{0,1\}$: customer i receives a folder with some ads

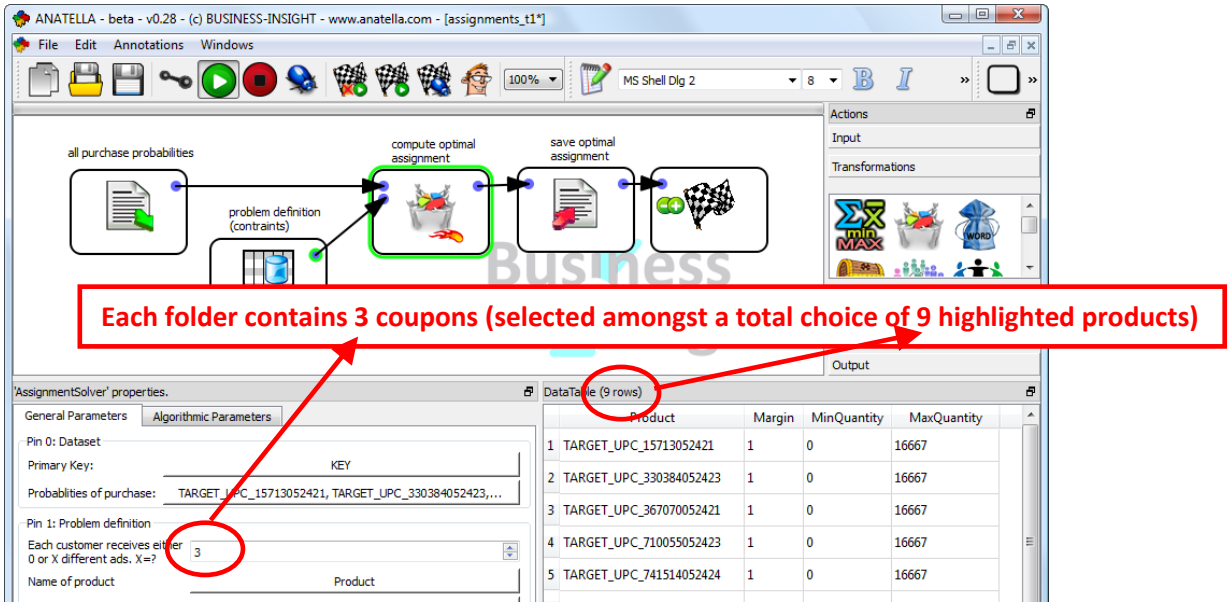
Computed using TIMi predictive models.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To use the AssignmentSolver Action, you need to provide these 2 input tables:

1. Probability Table: On input pin 0: one row= one customer: This table contains:
   - The primary key of each customer
   - The $PurchaseProbability_{i,j}$ for all customers and all products. These probabilities are typically computed using TIMi.
2. Problem-DefinitionTable: On input pin 1: one row=one highlighted product: This table contains 3 columns:
   - The name of the product: This name must match the corresponding column name inside the Probability Table.
   - The $ProductMargin_j$ (margin on each of the product)
   - The $ProductMax_j$ (the constraint: i.e. the maximum available quantity of coupons related to this product)

Here is an example:



The AssignmentSolver Action uses a "tabu-search" procedure to generate many good "candidates solutions". At the end, we'll return the best, admissible candidate solution (i.e.: The one with the highest ER="Expected Return" that satisfies all the contraints). The "tabu-search" procedure is an iterative procedure: it starts from a known candidate-solution and "changes" it slightly to create a new candidate solution. For example, this small "change" can be: Swapping 2 products between 2 different customers. The "tabu-search" procedure always tries to make a change that improves the "*quality of the candidate solution*". To avoid entering inside the "infeasible space" (i.e. to avoid entering the space composed of solutions that do not satify the constraints), the "*quality of the candidate solution*" is defined as:

$$QualityOfCandidateSolution = ExpectedReturn - \rho\ SumConstraintsViolations$$

$$SumConstraintsViolations = \left|\sum_{i,j} x_{i,j} - ProductMax_j\right| + \left|\sum_{i,j} x_{i,j} - 5y_i\right|$$

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If the ρ parameter is very **small**, the "tabu-search" procedure will allow "entering" the **infeasible** space.
If the ρ parameter is very **large**, the "tabu-search" procedure will be "pushed" towards the **feasible** space.

If there are many violated contraints inside the current solution, the assignement solver automatically increases the ρ parameter to "go back" towards the feasible space. Allowing the assignement solver to enter slightly inside the infeasible space is an important feature to avoid getting stuck "inside a local optimum" along the way toward the "best" solution.

It can happen that the next best candidate solution is a solution that we already tested previously. This is annoying because it can cause "infinite loops" inside the assignement solver (e.g. the solver always "circle around" testing always the same set of candidate solutions). To avoid such phenomenon, the solver keeps a list of "tabu" solutions (thus the name: "tabu-search"). A candidate solution is marked as "tabu" when it has been already "explored" at a previous iteration of the solver. The solver is then forbidden to "move to" a "tabu" solution. The "tabu-list" of forbidden solutions prevents the solver to "circle around". The tabu-list is also a very effective mechanism to "get out" of local optimum (to be able to find the "global" optimum).

The size of the "tabu-list" is limited using a FIFO methodology (FIFO=first in, first out). A large tabu-list allows you to be less sensitive to local optimums (and this is good) but it also increases significantly the running time.

The quality of the solution depends on the quality of the candidates generated by the tabu-search algorithm. To generate these candidates, at each iteration of the "tabu-search" procedure, we apply a small "change" to the current solution. These "incremental changes" must have several properties:

1. It should be easy (i.e. fast) to compute if a change has a positive or a negative impact on the "*quality of the solution*".
2. These changes should allow you to avoid getting "stuck" on sub-optimal solutions. More precisely: They should allow you to quickly explore ***all*** the different part of the feasible space (and not a small, limited part of the feasible space). This means that these "changes" must be very varied in nature.

Unfortunatly the 2 properties hereabove are antagonist objectives. More precisely: you can have:

1. **"simple" changes:** These are quick to compute but they do NOT "cover" correctly the complete feasible space (and thus you might miss the optimal, best solution using only these "simple changes").
2. **"complex" changes:** These are slow to compute but that allow you to explore "in-depth" all the feasible space.

Here are the differents steps that are executed inside the multiproduct assignement solver to compute the (near) optimal assignement:

1. **STEP 1.** Generate a first "candidate solution" using very simple heuristics. The quality of this very first solution is very weak (i.e. The ER is very low). The algorithms used in this phase are very simple and could be implemented in nearly any scripting language (including SAS).
2. **STEP2.** Fast solver:
   a) Use a very fast "tabu-search" procedure that only involves "simple changes" to quickly improve the solution found at the previous step. The "tabu-search" procedure is allowed to enter inside the infeasible space (the ρ parameter is **small**)
   b) Use a simple heuristic to find the closest feasible solution to the solution found in step 2.a).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

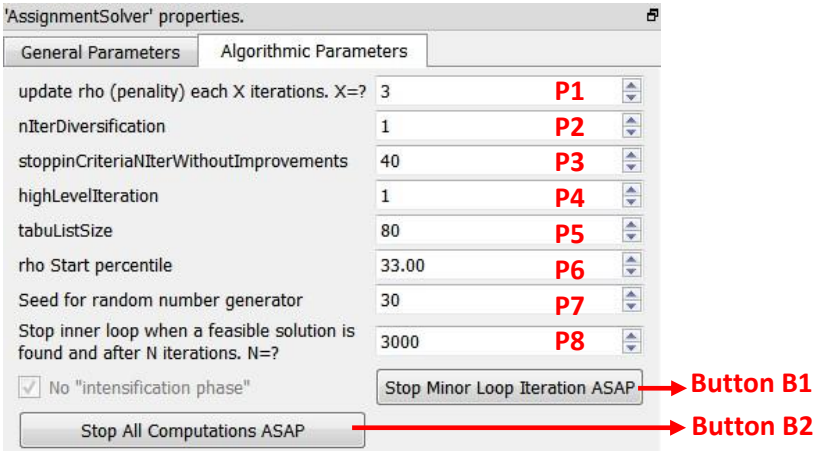
c) Use a very fast "tabu-search" procedure that only involves "simple changes" to quickly improve the solution found at step 2.b). The "tabu-search" procedure is *NOT* allowed to enter inside the infeasible space.

3. **STEP 3.** In-depth solver:
   a) Use a slow "tabu-search" procedure that involves "complex changes". Dynamically adjust the ρ parameter to allow to temporarily enter inside the infeasible space. If the ExpectedReturn (ER) of the solution does not improve for a large number of iterations, we break and go to step 3.b).
   b) The solution produced at step 3.a) is the best solution that we can obtain using:
      1. The (limited) set of "complex changes" that are implemented inside the assignment solver.
      2. A given starting point

      Let's scramble the current best solution using a large number of completely random perturbations. We now have a low-quality (and possibly infeasible) solution. Go back to step 3.a) and use this low-quality solution as starting point for the "tabu-search" procedure. This new starting point might (maybe) allow us to reach a better solution.

The "tabu-search" algorithms used during the steps 2&3 are carefully tuned C++ implementation. The extremely high speed of the C++ solver included inside Anatella allows us to explore a larger number of different candidate solutions than any other implementation, leading to a better solution (more ROI) at the end.

The parameters that control the solver are:



As usual with all meta-heuristics, the more time we explore the search space, the better the quality of the final, returned solution. Thus, it's important to avoid losing time exploring un-interesting regions of the search space.

1. Parameter P1: "*Update rho each X iterations. X=?*".

During step 3.a), the ρ parameter is dynamically adjusted every X iterations.

1.1. If the parameter P1 is too large we can inadvertently enter "deeply" inside the infeasible space. We'll then lose time simply "moving back to" (i.e. restoring) feasibility.

1.2. If the parameter P1 is too small, we can get "stuck" in a local optimum and find a really poor solution.

2. Parameter P2: "nIterDiversivication"

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

During step 3.b), we scramble the best know solution to obtain a new starting point for further optimization. The parameter P2 controls the amount of "scrambling" that is applied.

2.1. Too little "scrambling" and we'll re-explore again the same set of solutions that we already explored (and thus we won't find a better solution).

2.2. Too strong "scrambling" and we'll move really far away from the "region of interest" (i.e. the region of the search space that is the most likely to contain the optimal solution). We'll then lose a large amount of computing time just to "go back" to the "region of interest" (and this computing-time could be used more efficiently searching inside "region of interest" for the best solution).

3. Parameter P3: "stoppingCriteriaNIterWithoutImprovements"

During step 3.a), we'll run a slow "tabu-search" procedure that stops when the ExpectedReturn (ER) of the solution does not improve for P3 iterations.

3.1. If the parameter P3 is too small, we can abort prematurely the "tabu-search" procedure and potentially "miss" a very good solution.

3.2. The final solution found by one run of the "tabu-search" procedure (in step 3.a)) depends on the "starting point" (i.e. the candidate-solution that was used to initialize the "tabu-search" procedure). Some "starting points" generate better final solution than others. The objective of the "high-level" loop (that is composed of steps 3.a) and 3.b)) is to "test" different "starting points" (to see which one generates the best solution at the end). If the parameter P3 is too large, we might lose time investigating the outcome of an uninteresting "starting point" (Maybe it's better to "investigate" very roughly many different starting points than to "investigate" very deeply the outcome of a small number of starting points).

4. Parameter P4: "highLevelIteration"

The parameter P4 is the number of iterations inside the "high-level" loop (that is composed of steps 3.a) and 3.b)). The objective of this "high-level" loop is to "test" different "starting points".

If the parameter P4 is zero, then only the "fast solver" is used.

Proceed with caution when increasing P4: This increases a lot the running time of the multiproduct assignement solver.

5. Parameter P5: "tabu-list size"

A large tabu-list allows you to be less sensitive to local optimums (and this is good) but it also increases significantly the running time.

6. Parameter P6: "rho start percentile"

The parameter P6 controls the initial value of the ρ parameter inside the (slow) "tabu-search" procedure that runs in step 3.a).

6.1. If the parameter P6 is too small we'll start the tabu-search from a candidate-solution that is "deeply" inside the infeasible space. We'll then lose time simply "moving back to" (i.e. restoring) feasibility.

6.2. If the parameter P6 is too small, we can get "stuck" in a local optimum and find a really poor solution.

7. Parameter P7: "Seed for random number generator"

The "tabu-search" procedure (in steps 2.a) and 3.a)) has a strong random component.
The "diversification" procedure (in step 3.b)) has an even greater random component.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameter P7 defines the "seed" used inside the pseudo-random number generator that is used inside the multiproduct assignement solver. For more information about pseudo-random number generator, see:
http://en.wikipedia.org/wiki/Pseudorandom_number_generator

8. Parameter P8: "Stop inner loop when a feasible solution is found and after N iterations.N=?"

Ideally, the slow "tabu-search" procedure of step 3.a) uses the parameter P3 as the only stopping criteria. However, it can happen that the slow "tabu-search" procedure manages to continuously improve (i.e. at nearly each iteration) the ER of the "*best found solution so far*". These improvements are totally negligible but they still prevent the stopping criteria based on parameter P3 to work properly. The parameter P8 is useful when such rare circumstance occurs. The parameter P8 places a hard upper bound on the maximum number of iterations performed inside the slow "tabu-search" procedure. The effects of a very small (or very large) value for the parameter P8 are the exact same effects as for the parameter P3.

The multiproduct assignement solver has also 2 buttons that are only active when the solver is running:

1. Button B1: "Stop minor loop iteration ASAP"

When you click the B1 button the loop inside the "tabu-search" in step 3.a) stops immediately. This is interesting when you notice (looking at the numbers inside log window) that the current "starting point" is not interesting: it won't produce a better solution (i.e. a solution with a higher ER than the "*best solution known so far*"). In such situation, click the Button B1: the solver will directly "jump to" step 3.b).

2. Button B2: "Stop All Computations ASAP".

The parameter P4 is the number of iterations inside the "high-level" loop (that is composed of steps 3.a) and 3.b)). If the parameter P4 is very high, the solver will run for a very, very long time (testing many different "starting points"). It can happen that you need the final assignement ***now*** (and you don't want to wait for still another lengthy "high-level" iteration). In such situation, click the button B2: The multiproduct assignement solver will nearly **instantaneously** output the "*best solution known so far*".

Large Assignment Problems

To solve large assignment problems, set to zero the parameter P4 ("highLevelIteration"):

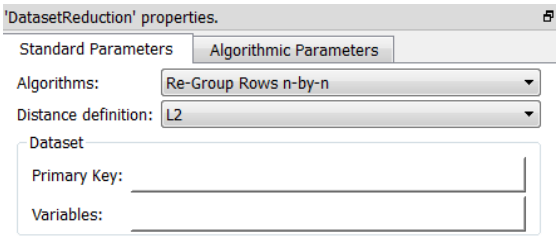
This will only run the "fast solver" (and completely skip the "In-depth solver"). The "fast solver" has also a very small RAM memory consumption (especially compared to the RAM memory consumption of the "In-depth solver"). This allows you to solve really large assignment problems with a computer that has a very limited RAM memory.

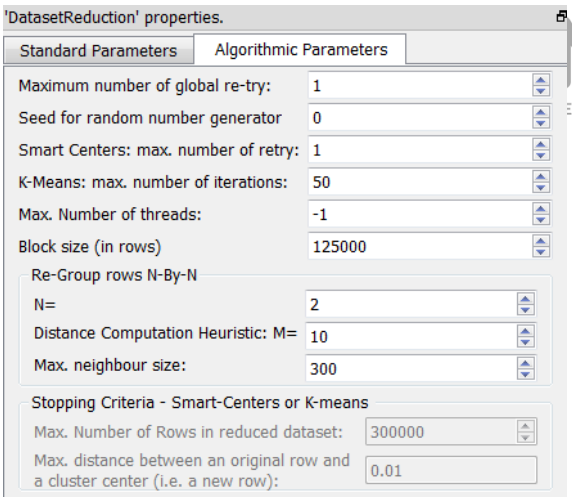
5.16.2. Dataset Reduction (High-Speed action)

Icon:

Property window:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu





Long Description:
Reduce the dataset by applying a KNN segmentation.

When dealing with transactional data, sampling (even stratified sampling) will force us to lose precious information when we create predictive models. One way to avoid this problem (while keeping the dataset reasonable) is to regroup transactions of customers that look very similar in terms of purchase behaviour.

The stratified sampling is appropriate when we want to keep representativity of a sample for inferrence purpose, but there is absolutely nothing that suggest that two people of the same age, region and ehtnic origin have the same tastes or purchase behavior. On the other hand, it doesn't make a large difference that on Fridays, Daniel buys 6 beer, 2 Camembert and 1 bread, while Frank buys 2 beers, 2 Brie and 1 bread. They essentially have the same purchase behaviour.
This technique will reduce the dataset by identifying transactions that are seemingly the same.

Parameters:

Algorithms:
- Re-group rows n by n
- Smart Centers only
- K-Means only
- Smart Centers + K-Means

Distance Definition:
- L2
- L1
- 0.9* L_Infinity + 0.1 * L1
- 0.95 * L_Infinity + 0.05 * L1

Primary key: unique transaction identifier
Variables: set of columns that are relevant to define the distances between rows.

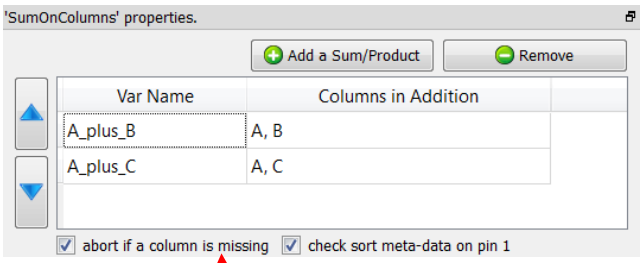
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.17. TA – Time Series (TA=Transformations Actions)

5.17.1. Aggregate on Columns (High-Speed action)

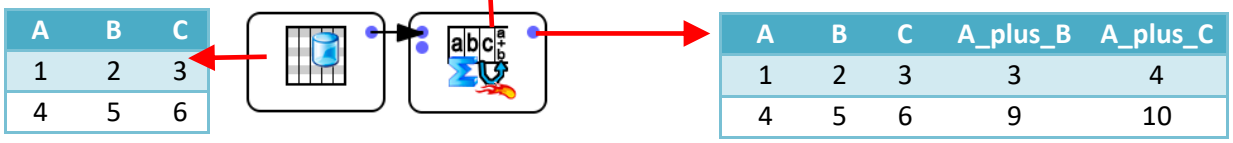
Icon:

Property window:



Short description:
Sum some columns

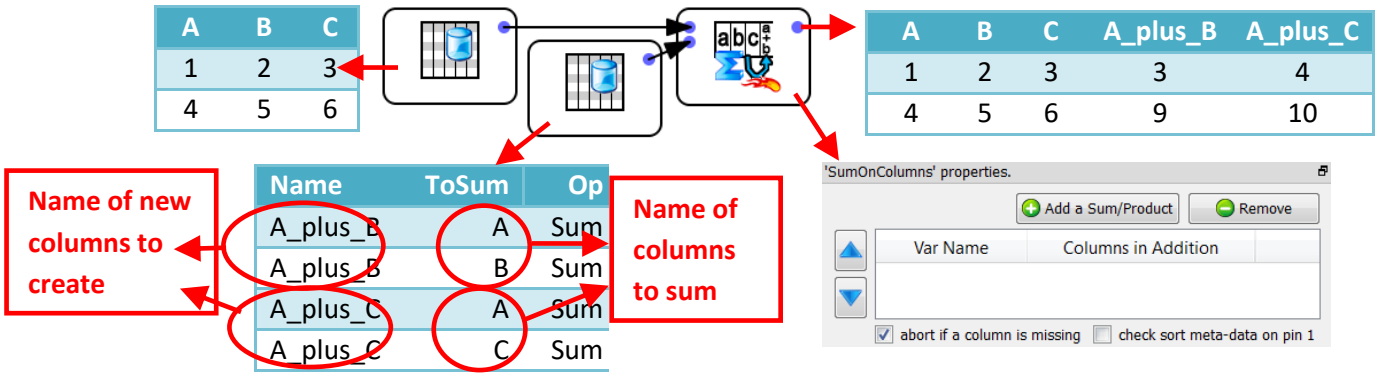
Long Description:
This Action creates some new columns that are the sum of the selected columns.
For example:



A	B	C
1	2	3
4	5	6

A	B	C	A_plus_B	A_plus_C
1	2	3	3	4
4	5	6	9	10

The table connected to the pin 0 of the AggregateOnColumns Action is the table on which to compute all the aggregation. The table connected to the pin 1 of the AggregateOnColumns Action is optional. This optional 3-columns-table defines some additional aggregations to compute. For example, the above Anatella graph is equivalent to:



A	B	C
1	2	3
4	5	6

A	B	C	A_plus_B	A_plus_C
1	2	3	3	4
4	5	6	9	10

Name	ToSum	Op
A_plus_B	A	Sum
A_plus_B	B	Sum
A_plus_C	A	Sum
A_plus_C	C	Sum

The third column on the second input pin (i.e. the column "Op" in the example above) may contain the following strings (case insensitive): imin, imax, smin, smax, first (first non-null), last (last non-null), mean, product, geometricMean, stdDev, countDistinct, countNonEmpty, countNonNull, countNonZero, mostCommonModality, countMostCommonModality.

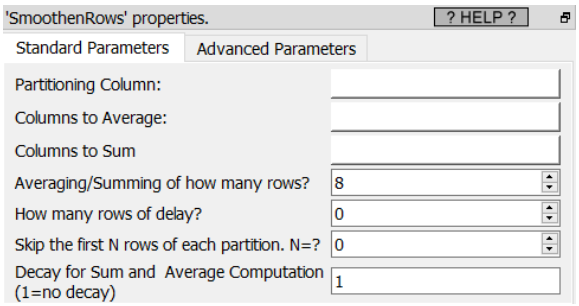
The AggregateOnColumns Action can run inside a N-Way multithreaded section (see section 5.3.2. about multithreading).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.17.2. Smoothen rows (High-Speed action)

Icon:

Property window:



Short description:
smoothen n rows

Long Description:
This operator is used to create datasets to do time-series prediction.
Let's assume that you have the following original dataset:

ORIGINAL TABLE		
ValueA	ValueB	TargetToPrediction
ValueA at time t=1	ValueB at time t=1	Target at time t=1
ValueA at time t=2	ValueB at time t=2	Target at time t=2
ValueA at time t=3	ValueB at time t=3	Target at time t=3
ValueA at time t=4	ValueB at time t=4	Target at time t=4
ValueA at time t=5	ValueB at time t=5	Target at time t=5
...	...	...

You could directly use the above dataset to create a predictive model that predict the target value at time T in function of the measured value A & B that occurs at the same time T. ...but you won't get any good predictive model because, most of the time, the target value at time T can only be predicted using the A & B values at time T, T-1, T-2, T-3, ...

TIMIcreates predictive models that,are predicting the Target on a given ROW based on the value contained in the other columns on the SAME ROW, I.e. You need a dataset that contains on each row all the required values to do the prediction for that row.

For example, for time-series prediction, if the target value at time T depends on:

- the sum of the A values at time T-2,T-3,T-4
- the sum of the B values at time T-2,T-3,T-4

(i.e. the latency of the process to predict is '2' and you must sum '3' different time period), you should have:

TRANSFORMED TABLE to obtain a suitable learning dataset for time-serie analysis		
...	...	...
(ValueA at T-2)+(ValueA at T-3)+(ValueA at T-4)	(ValueB at T-2)+(ValueB at T-3)+(ValueB at T-4)	Target at T
...	...	...

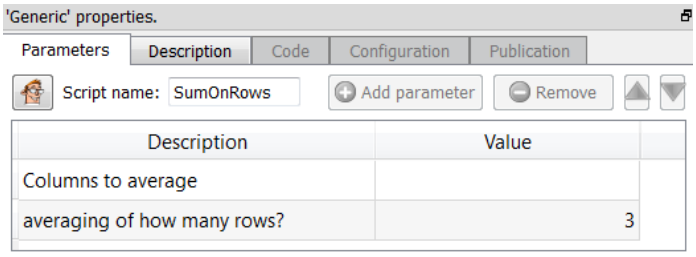
NOTE:
You can also obtain the same transformation using the 3 following operators: "Time Travel", "Sum on Columns", "Filter Column".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.17.3. Sum on Rows

Icon:

Property window:



Short description:
Sum n rows

Long Description:
This Action can be summarized as a "group by" 'n' consecutive rows Action.

Let's assume that:

- You want to create a chart in excel that represents the average SALES amount per week.
- you have an original table that contains, for each day, the SALES amount:

ORIGINAL TABLE	
DATE	DAY SALES
June 1, 2010	1500 $
June 2, 2010	1500 $
June 3, 2010	1500 $
June 4, 2010	5000 $
June 5, 2010	1500 $
June 6, 2010	1500 $
June 7, 2010	1500 $
June 8, 2010	2000 $
June 9, 2010	2000 $
June 10, 2010	2000 $
June 11, 2010	2000 $
June 12, 2010	2000 $
June 13, 2010	2000 $
June 14, 2010	2000 $

i.e. You want to obtain the following table, to be able to create your chart:

TRANSFORMED TABLE	
DATE	AVERAGE DAY SALES FOR THIS WEEK
June 1, 2010	2000 $
June 8, 2010	2000 $

The objective of this operator is to obain the final "TRANSFORMED TABLE" based on the "ORIGINAL TABLE".

NOTE:
This operator is mainly useful when you want to reduce the number of rows of a table to obtain a "synthetized" version that is more suitable for visualization. In normal situation, this operator should not be used in conjunction with a predictive analysis because the "TRANSFORMED TABLE" contains a lot less

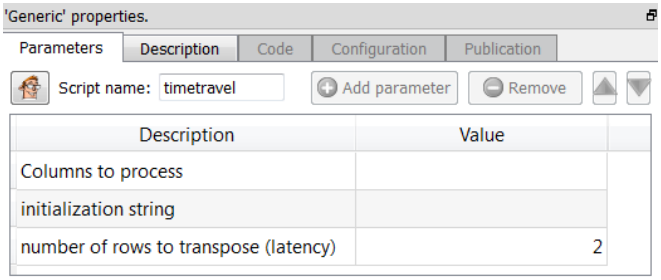
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

information compared to the "ORIGINAL TABLE" and will usually generate less accurate predictive models.

5.17.4. Time Travel

Icon:

Property window:



Short description:
Timetravel

Long Description:
This operator is used to create datasets to do time-series prediction.
Let's assume that you have the following original dataset:

```
+-----------------------------------------------------------------+
|                          ORIGINAL TABLE                         |
+--------------------+--------------------+---------------------+
|              EventA|              EventB|  TargetToPrediction|
+--------------------+--------------------+---------------------+
| EventA at time t=1 | EventB at time t=1 | Target at time t=1  |
| EventA at time t=2 | EventB at time t=2 | Target at time t=2  |
| EventA at time t=3 | EventB at time t=3 | Target at time t=3  |
| EventA at time t=4 | EventB at time t=4 | Target at time t=4  |
| EventA at time t=5 | EventB at time t=5 | Target at time t=5  |
|                ...|                ...|                 ...|
+--------------------+--------------------+---------------------+
```

You could directly use the above dataset to create a predictive model that predict the target value at time T in function of the events (A & B) that occurs at the same time T. ...but you won't get any good predictive model because, most of the time, the target value at time T can only be predicted using the events at time T, T-1, T-2, T-3, ...

TIMi creates predictive models that are predicting the Target on a given ROW based on the value contained in the other columns on the SAME ROW, I.e. You need a dataset that contains on each row all the required values to do the prediction for that row.

For example, for time-series prediction, if the target value at time T depends only on events at time T,T-1,T-2 (i.e. the latency of the process to predict is maximum '2'), you should have:

```
+-------------------------------------------------------------------------------------------------------------+
|              TRANSFORMED TABLE  to obtain a suitable learning dataset for time-serie analysis               |
+-------------+---------------+---------------+-------------+---------------+---------------+-------------+
|          ...|            ...|            ...|          ...|            ...|            ...|          ...|
|          ...|            ...|            ...|          ...|            ...|            ...|          ...|
| EventA at T | EventA at T-1 | EventA at T-2 | EventB at T | EventB at T-1 | EventB at T-2 | Target at T |
|          ...|            ...|            ...|          ...|            ...|            ...|          ...|
|          ...|            ...|            ...|          ...|            ...|            ...|          ...|
+-------------+---------------+---------------+-------------+---------------+---------------+-------------+
```

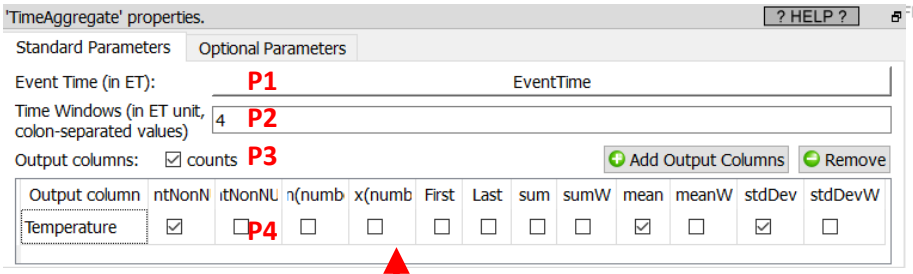
The objective of this operator is to obtain the final "TRANSFORMED TABLE" based on the "ORIGINAL TABLE", so that you can easily use TIMi to perform a time-series predictive analysis.

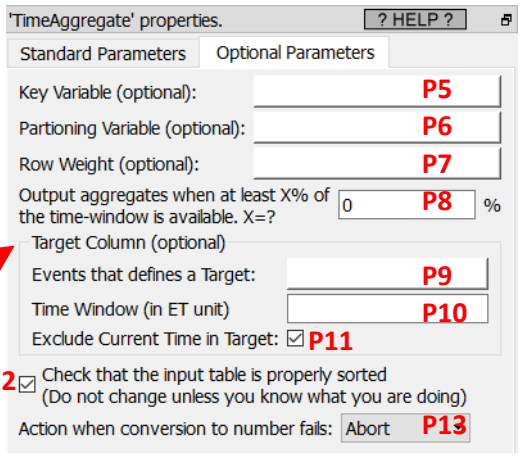
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.17.5. Time Aggregate

Icon:

Property window:



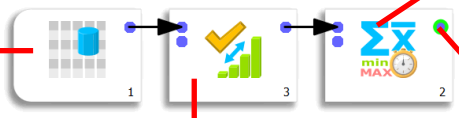


Short description:
Time aggregate

Long Description:
This operator is used to create datasets to do time-series prediction.

Let's assume that you have the following simple graph:

EventTime	Temperature
1	100
2	114
3	122
4	120
5	
6	126
7	108
8	102
9	92
10	96
11	100
12	114
13	122
14	120



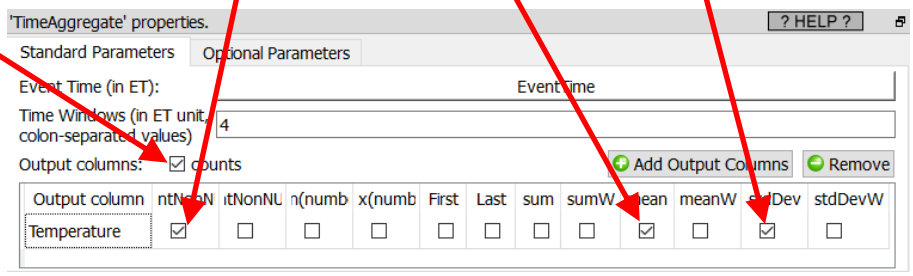
Numeric Sort on "EventTime"

Here is a missing/ unknown value.

Output table:

Temperature	Temperature _countNN	Temperature _mean	Temperature _stdDev	count	EventTime
100	1	100		1	1
114	2	107	9.89949	2	2
122	3	112	11.1355	3	3
120	4	114	9.93310	4	4
	3	118.6666	4.16333	4	5
126	3	122.6666	3.05505	4	6
108	3	118	9.16515	4	7
102	3	112	12.4899	4	8
92	4	107	14.28287	4	9
96	4	99.5	6.99999	4	10
100	4	97.5	4.434711	4	11
114	4	100.5	9.574277	4	12
122	4	108	12.1106	4	13
120	4	114	9.933109	4	14

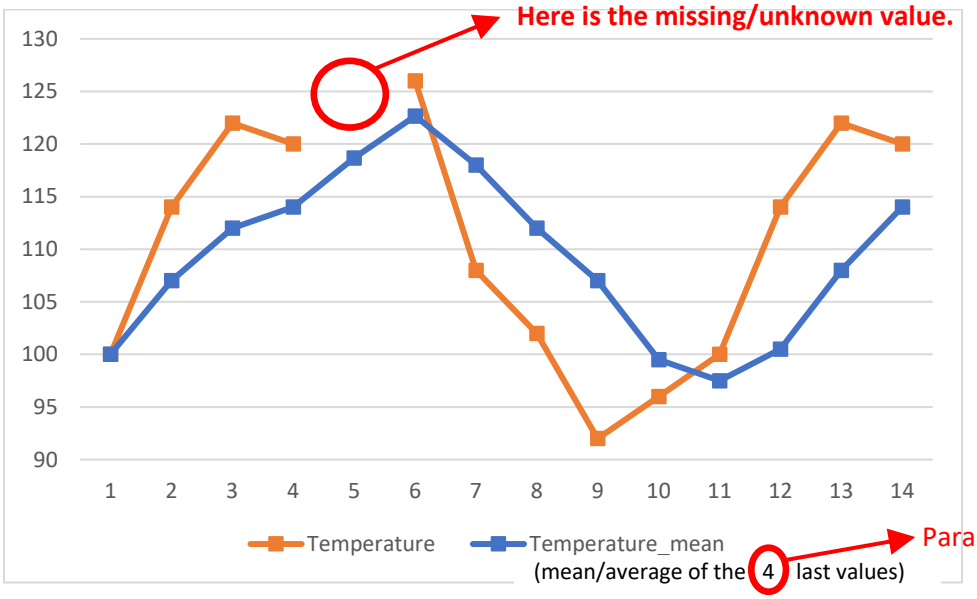
Inside the output table, the columns are:
(1) count, (2)Temperature_countNN, (3)Temperature_mean, (4)Temperature_stdDev:

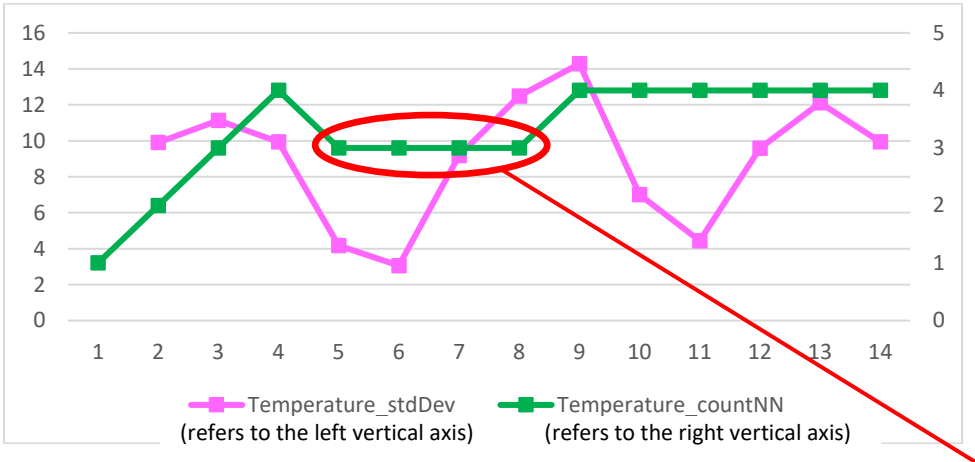


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

"countNN" stands for "*count of non-null values*".

Here is a graphical illustration of the columns:
(1) Temperature, (2)Temperature_mean, (3)Temperature_countNN, (4)Temperature_stdDev:





<u>One little observation here:</u>

Let's compare together these two curves: the curve of the "temperature_mean" and the curve of the "temperature". We notice that the "temperature_mean" curve can be interpreted as a "smoothed out" version of the "temperature" curve that is slightly "lagging" behind. When you compute any kind of aggregate, there is always some kind of "lag" introduced inside the aggregate. This "lag" increases with (is proportional to) the duration of the "time window".

Using the parameter **P4**, you can compute the following aggregates for many different columns and many different time-windows (the different time-windows are defined using parameter **P2**):

- "Count": the number of samples inside the time window
- "CountNonNull": the weigthed number of non-null samples inside the time window. The weight of each row is (optionnaly) defined using the parameter **P7**.
- "Min(number)": the minimum value inside the non-null samples inside the time window.
- "Max(number)": the maximum value inside the non-null samples inside the time window.
- "First": the first value inside the non-null samples inside the time window.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- "Last": the first value inside the non-null samples inside the time window.
- "Sum": the weigthed sum of the values inside the samples inside the time window. The weight of each row is defined using the parameter **P7**.
- "Mean": the weigthed Mean/Average of the values inside the samples inside the time window. The weight of each row is (optionnaly) defined using the parameter **P7**.
- "StdDev": the Standard deviation of the values inside the samples inside the time window. The weight of each row is (optionnaly) defined using the parameter **P7**.

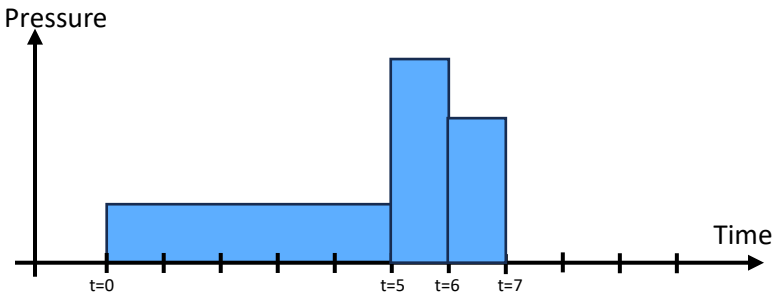
In addition to the above "classical" aggregates, you have 4 more "enhanced" aggregates (all their names end with the letter "W", to separate them from their equivalent "classical" aggregate):

- "CountNonNullW"
- "SumW"
- "MeanW"
- "StdDevW"

5.17.5.1. About the "Enhanced" Aggregates

To explain why these "enhanced" aggregates exists, let's use two simple examples where the time-interval between two samples is not constant:

Example 1

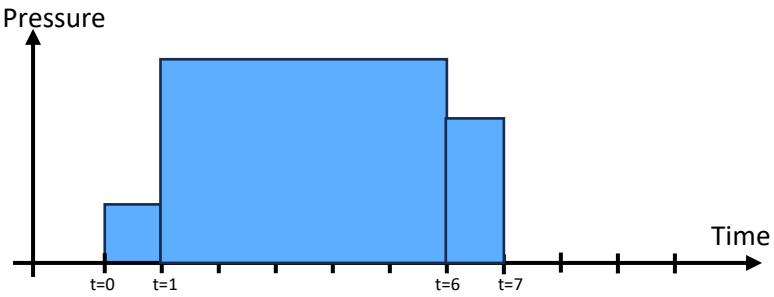


The DataSet for example 1 is:

EventTime	Pressure
0	10
5	40
6	31

In this first example, the pressure is at "10" for 5 seconds (or 5 "unit of time") and then it rises to 40 for 1 second.

Example 2



The DataSet for example 2 is:

EventTime	Pressure
0	10
1	40
6	31

In this second example, the pressure is at "10" for only 1 seconds and then rises to 40 for a longer period of time of 5 seconds.

At the timestamp t=6, the "Pressure_mean" (i.e. the average of the last samples of the "pressure" column) over a large time window (i.e. over the last 7 seconds) is equal for both example 1 and example 2: More precisely, it is:

Pressure_mean = (10+40+31)/3=27.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If you look at the above two charts (related to our two examples), you'll see that this does not seem right: i.e. The "average pressure" cannot be the same for the two charts here above. Indeed, on the first chart, the Pressure is at 10 for 5 seconds (or 5 "unit of time") and then rises to 40 for only 1 second. While on the second chart, the Pressure is at "10" for only 1 seconds and then it rises to 40 for a "much longer" period of time of 5 seconds.

The situation described here above is typically the situation where you should rather use the "enhanced" aggregates rather than the "simple" aggregates. Why? That's because **the "enhanced" aggregates are taking into account the duration of each sample**. These "enhanced" aggregates will thus deliver better results, closer to the common sense. Let's now compute the "enhanced" aggregate that is named "Pressure_meanW" for both examples:

For the example 1, at the time stamp t=6:

$$Pressure_{meanW} = \frac{5 \times 10 + 1 \times 40}{1 + 5} = 15$$

For the example 2, at the time stamp t=6:

$$Pressure_{meanW} = \frac{1 \times 10 + 5 \times 40}{5 + 1} = 35$$

Please note that, inside the above two equations (to compute the "meanW"), we did not use the value "31" (that is the "Pressure" at the time stamp t=6). This means that the "meanW" aggregates has slightly more "lag" compared to the "classical" aggregate "mean".

Here is a summary:

When to use "enhanced" aggregates?

- When you want to analyze signals or samples that are coming from "physical sensors".
  For example: temperature, pressure, flow, current, tension, etc.
- When the samples can each represent a different duration.
- When the aggregates can have slightly more "lag" than the "classical" aggregates (another way to reduce the lag is to reduce the duration of the time window)

When to use "classical" aggregates?

- When you want to analyze discrete signals.
  For example: the number of calls/events since the last timestamp, the number of incidents since the last timestamp, the number of manufactured pieces since the last timestamp, etc.

5.17.5.2. Feature engineering for Predictive Modeling/AI

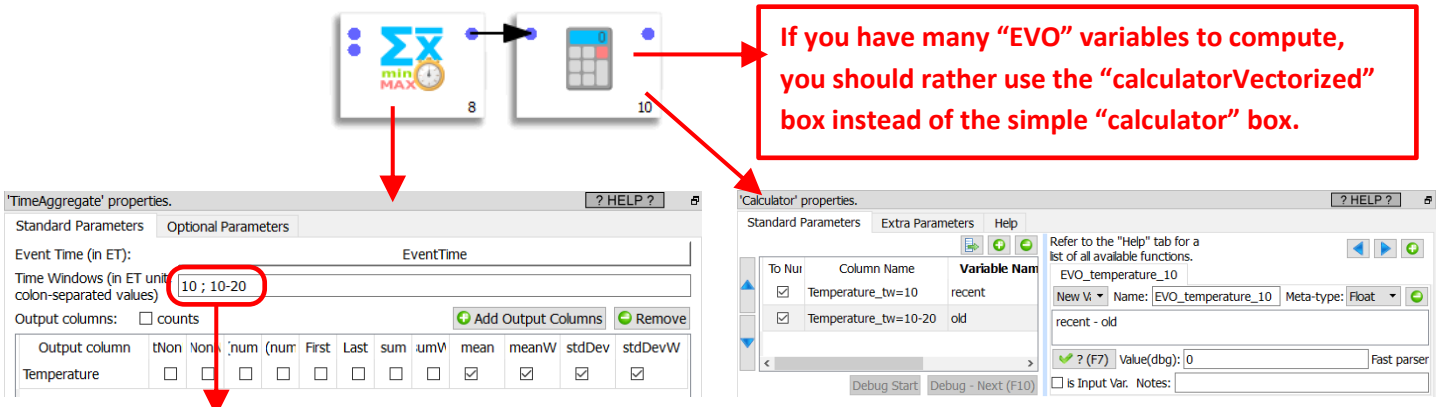
If you are using the timeAggregate box to compute some features to put inside a "Learning/Creation dataset" with the final objective of creating a new predictive model, you should be careful.
Why? Because the "classical" aggregates are strongly correlated to the "enhanced" aggregates (i.e. the way these two aggregates are computed is closely related). This means that, most of the time, it's best to either use the "enhanced" aggregates or either use the "classical" aggregates, **but not both**, inside your "Learning/Creation dataset" (to create your predictive model). Why? Because if you use both aggregates (i.e. the "enhanced" aggregates and the "classical" aggregates), you might end-up with a very bad predictive model because of "multi-colinearity problems".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If you use "TIMi Modeler" to create your predictive models, it's relatively safe to use both "aggregates styles" simultaneously inside your Learning/Creation dataset. Why? Because "TIMi Modeler" is relatively insensitive and quite robust against "multi-colinearity problems".
Furthermore, if your Target size is very small (less than 0.1%), it's always better to use a Learning/Creation dataset with a small quantity of variables (even when using "TIMi Modeler") so that the feature-selection algorithm works better. So, when you have a small Target, I advise you to use one of the two "aggregate styles" (but not both), just to have less variables.

What's really important for feature engineering for predictive modeling is to select the right "time scales" when aggregating your signals. The process of finding the right "time scales" is of "trial and error": i.e. You need to test many different "time scales" before finding the right ones. I would advise you to create a very "wide" "Learning/Creation dataset", with plenty of variables (that are coming from different "time scales") and let "TIMi Modeler" select for you which are the meaningful variables, linked to the meaningful "time scales". You can easily use the timeAggregate box to create these very "wide" datasets for predictive modeling: We designed this box especially for that. You just need to define many different time-windows inside the parameter **P2** and you'll get in output many variables to "inject" into "TIMi Modeler".

When doing any "feature engineering" exercice, it's always interesting to create "evolution" variables. "Evolution" variables are the variables that are encoding a "change of state" (e.g. a deterioration, an improvement). For example, an interesting "evolution" variable would be the difference of temperature between the average temperature computed over the last 10 minutes compared to the same average temperature but computed on the 10 minutes before the last 10 minutes. This variable would help encode a "sudden drop" in temperature (which can sometime be a very good indicator of a forecoming breakdown). To compute this "EVO" variable, do the following:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

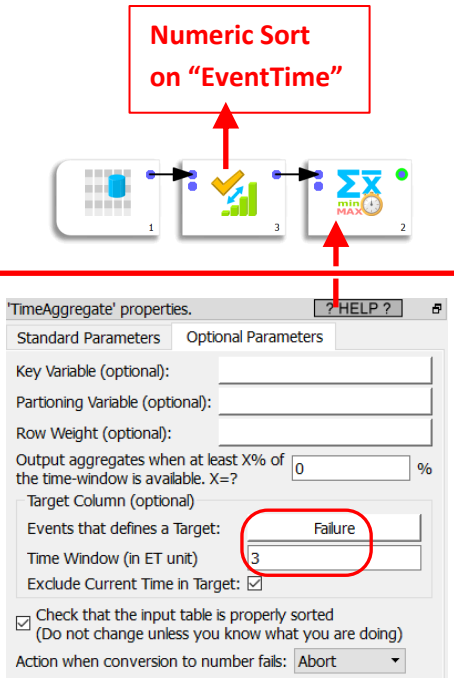
5.17.5.3. Target events

The parameters P9, P10 and P11 allows you to easily create a the "Target" columns to put inside a "Learning/Creation" dataset for predictive modeling. The output "target" columns created by the timeAggregate Action are BINARY targets. To create these output "target" columns, Anatella does the following for each different input target column (as defined using the parameter P9):

- Anatella looks into a time window "TW" that is into the "future" (i.e. that is inside the input rows "down below" compared to the current row). The size of this time-window is defined using the parameter P10.
- If the target column inside the time window "TW" inside the input table is a number (F or K meta-type) that is non-zero (and non-null), then the target is set to one on the current row.

Here is an example:

EventTime	Temperature	Failure
1	100	0
2	114	0
3	122	
4	120	0
5	128	0
6	126	1
7	106	0



Temperature	Temperature _mean	Failure _target	EventTime
100	100	0	1
114	107	0	2
122	112	1	3
120	114	1	4
128	121	1	5
126	124		6
106	120	0	7

Turns to "null" when there is a target in input at this timestamp.

Returns to zero as soon as we find a "full" time window "TW" without any targets. Otherwise, it stays "null".

5.17.5.4. Other

The optional parameter **P5** ("Key variable") is useful if you want to simply forward one column from input to output. This is useful to be able to keep a "key" column to do some join later on.

The optional parameter **P6** ("Partioning") is self-explanatory: All the aggregates "resets" ("as if" the input table was restarted again) when the content of the "partionning" column changes.

5.17.5.5. Dynamic Aggregation

It can happen on some occasion that you don't know "in advance" the precise nature of the aggregation that you want to compute: i.e. you need to run some computation to now exactly which aggregations to compute. In such situation, you should use the second input pin (pin 1) from the timeAggregate Action. More precisely, the table on the second input pin contains the "output column" information. It only has two columns:

- The first column is the name of the variable to process/aggregate.
- The second column contains a string that describes the aggregate to compute.
  The accepted values are (case insensitive):
  - IMin (number minimum)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

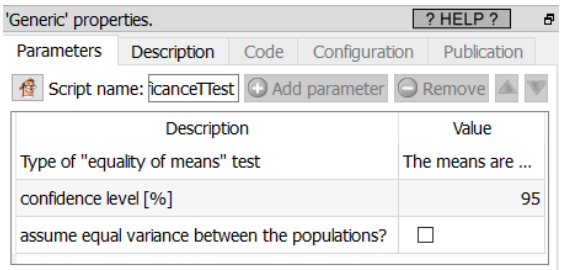
- IMax (number maximum)
- First
- Last
- Sum
- Mean
- StdDev
- CountNonNull
- SumW
- MeanW
- StdDevW
- CountNonNullW

5.18. TA – Stats (TA=Transformations Actions)

5.18.1. Student T-test

Icon:

Property window:



Short description:
Compare if two means are equal.

Long Description:
We are comparing different means computed on the first population (that is described on the first row of the input table) to the means of the other populations (that are described on the subsequent rows of the input table). We thus compute several different comparisons between 2 populations. If you need to compare together **more than 2** populations (to see if **all** their means are equal), you should rather use an ANOVA test.

The test implemented here assume a normal distribution of the variables (it's the classical Student T-test or Welch's T-test).

The output table contains:

- The "P-Value": It's the probability that the 2 means are indeed equal (or not significantly different)
- The "test OutCome": it's equal to one (true) when the test succeed (e.g. when we declare the two means equal, or not significantly different)

What's the "Confidence level"?
The probability of detecting an equality of the means (while the two means are indeed equal).

We can easily declare to have found an equality while there is actually no equality: i.e. We don't want to miss any possible equality.

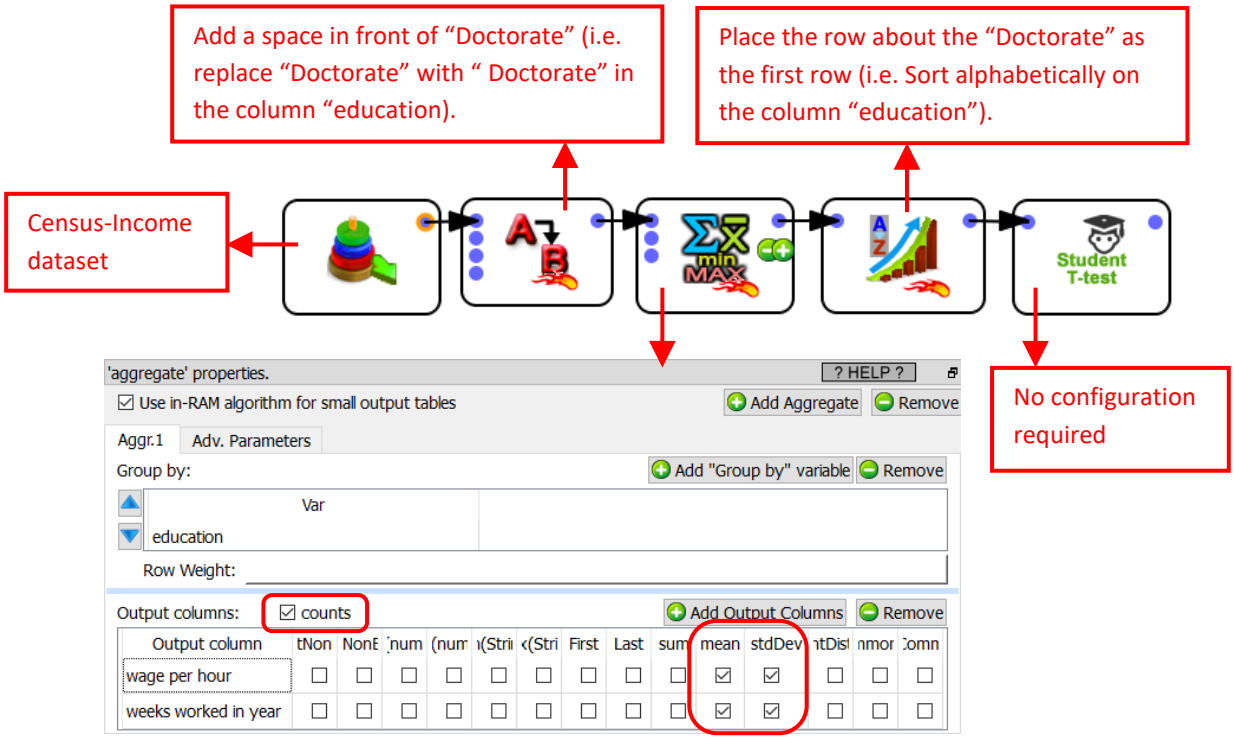
When the "confidence level" increases, we accept more and more greater differences between the means, to avoid missing any positive case.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

A confidence level of 95% indicates a 5% risk of concluding that a difference exists when there is no actual difference.

The most common value for the "confidence level" is 95% (0.95).
We also have this relation: "Confidence level" = 100% - "Significance Level"

Here is an example of usage:



The exact output table of the Student T-test student T-Test action is given on the next page.

What do we see inside this output table? Here are 3 conclusions extracted from the table:

- **Conclusion 1:** Somebody with a doctorate won’t have some wages that are fundamentally different that somebody that only got a “First Grade Degree” (i.e. the average “wage per hour” between these 2 groups of people is not different). So, obtaining a doctorate does not seem to “pay off”?
- **Conclusion 2:** If you have a doctorate, your average “wage per hour” is different than somebody without any degree at all (with the expection of some basic College degree). Since these two education levels seems really different (in terms of “wage per hour”), the next question is: What’s the best degree to have between these two (in terms of “wage per hour”)? The answer might surprise you!
- **Conclusion 3:** Because the last column on the right is filled with plenty of zero’s, it seems the average “number of weeks worked in year” for the people with a doctorate degree is significantly different (i.e. higher) than for most of the other eduction levels. Basically, the people with a “doctorate degree” are working more than the others?

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Compared Populations	wage per hour P-Value	wage per hour TestOutcome	weeks worked in year P-Value	weeks worked in year TestOutcome
" Doctorate degree(PhD EdD) vs "10th grade"	0.287207	1	6.66E-16	0
" Doctorate degree(PhD EdD) vs "11th grade"	0.0329071	0	3.33E-16	0
" Doctorate degree(PhD EdD) vs "12th grade no diploma"	0.202915	1	1.11E-16	0
" Doctorate degree(PhD EdD) vs "1st 2nd 3rd or 4th grade"	0.466529	1	3.33E-16	0
" Doctorate degree(PhD EdD) vs "5th or 6th grade"	0.611911	1	1.11E-16	0
" Doctorate degree(PhD EdD) vs "7th and 8th grade"	0.536412	1	4.44E-16	0
" Doctorate degree(PhD EdD) vs "9th grade"	0.79993	1	0	0
" Doctorate degree(PhD EdD) vs "Associates degree-academic program"	1.32E-09	0	0.00069187	0
" Doctorate degree(PhD EdD) vs "Associates degree-occup /vocational"	8.60E-13	0	6.06E-06	0
" Doctorate degree(PhD EdD) vs "Bachelors degree(BA AB BS)"	0.00040626	0	0.00012546	0
" Doctorate degree(PhD EdD) vs "Children"	0.00448568	0	1.11E-16	0
" Doctorate degree(PhD EdD) vs "High school graduate"	3.98E-07	0	4.44E-16	0
" Doctorate degree(PhD EdD) vs "Less than 1st grade"	0.25901	1	0	0
" Doctorate degree(PhD EdD) vs "Masters degree (MA MS MEng MEd MSW MBA)"	0.0902351	1	0.0596281	1
" Doctorate degree(PhD EdD) vs "Prof school degree (MD DDS DVM LLB JD)"	0.424434	1	0.085153	1
" Doctorate degree(PhD EdD) vs "Some college but no degree"	4.07E-07	0	1.11E-15	0

Conclusion 1

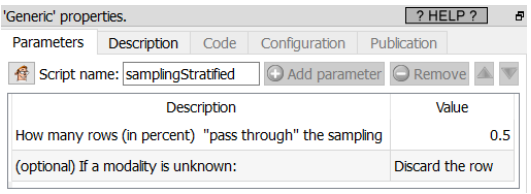
Conclusion 2

Conclusion 3

5.18.2. Stratified Sampling

Icon:

Property window:



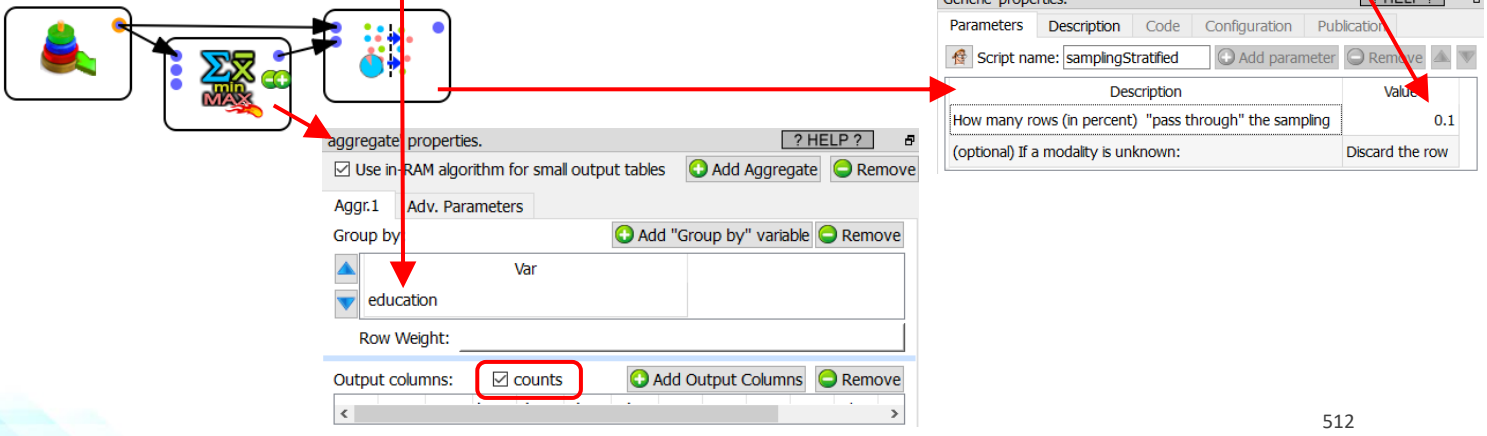
Short description:
Stratified sampling on modal variable(s).

Long Description:
stratified sampling on modal variable(s).

Here is an example: Let's assume that:

- …we want to randomly select 10% of the input rows.
- …we want to "stratify" on the column "education": i.e. we must ensure that the distribution of the variable "education" stays exactly the same inside our sample than inside our original population.

We'll have:

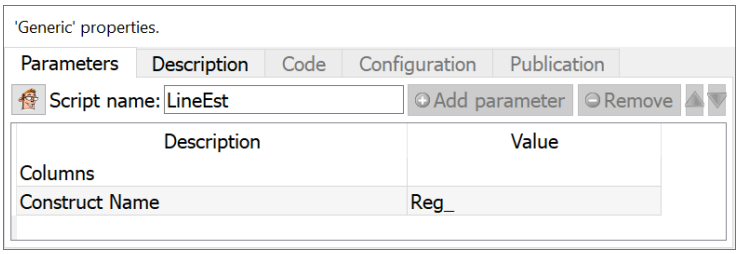


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.18.3. Line Regression

Icon:

Property window:



Short description:
Compute a linear regression, row-by-row.

Long Description:
Compute a linear regression by row, when each column corresponds to an equal interval of time. When working with time series, this is often a very useful transformation as it allows to compute the trend overtime at a row level.

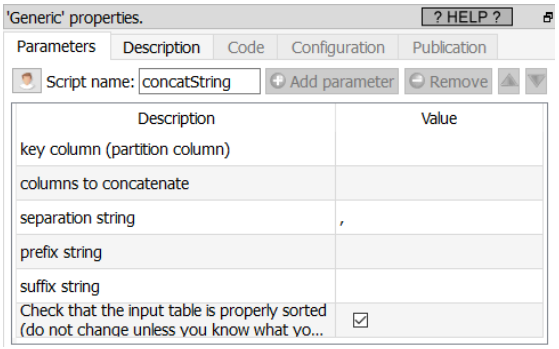
Typically, we compute: trend at 3,6,12 and 24 months. The LineEst note output consists of two variables: Beta 0 and Beta 1. Those variables are also useful inputs of segmentation, much easier to deal with than original purchase history.

5.19. TA – Formatters (TA=Transformations Actions)

5.19.1. Concat

Icon:

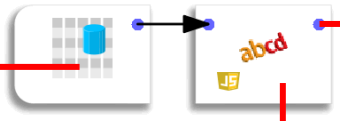
Property window:



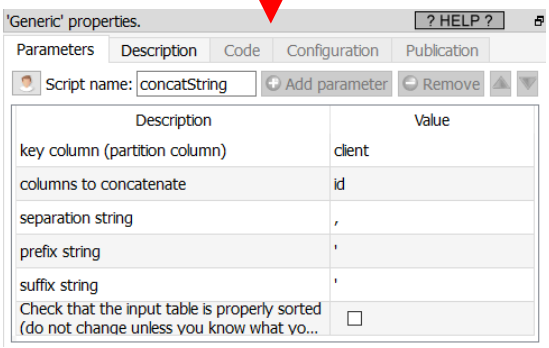
Short description:
Aggregate Strings on different rows using the concatenation operation.

Long Description:
Let's give an example:

client	id
frank	a
frank	b
frank	c
lara	d
lara	e



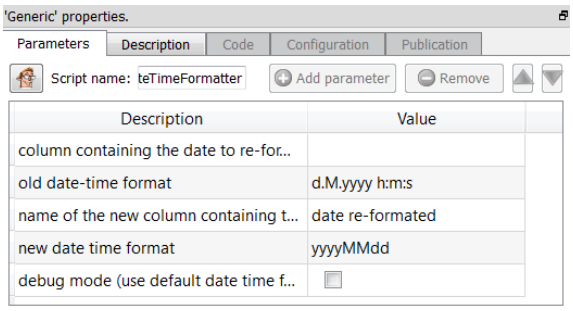
client	id
frank	'a','b','c'
lara	'd','e'



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.19.2. Date/Time Formatter

Icon:

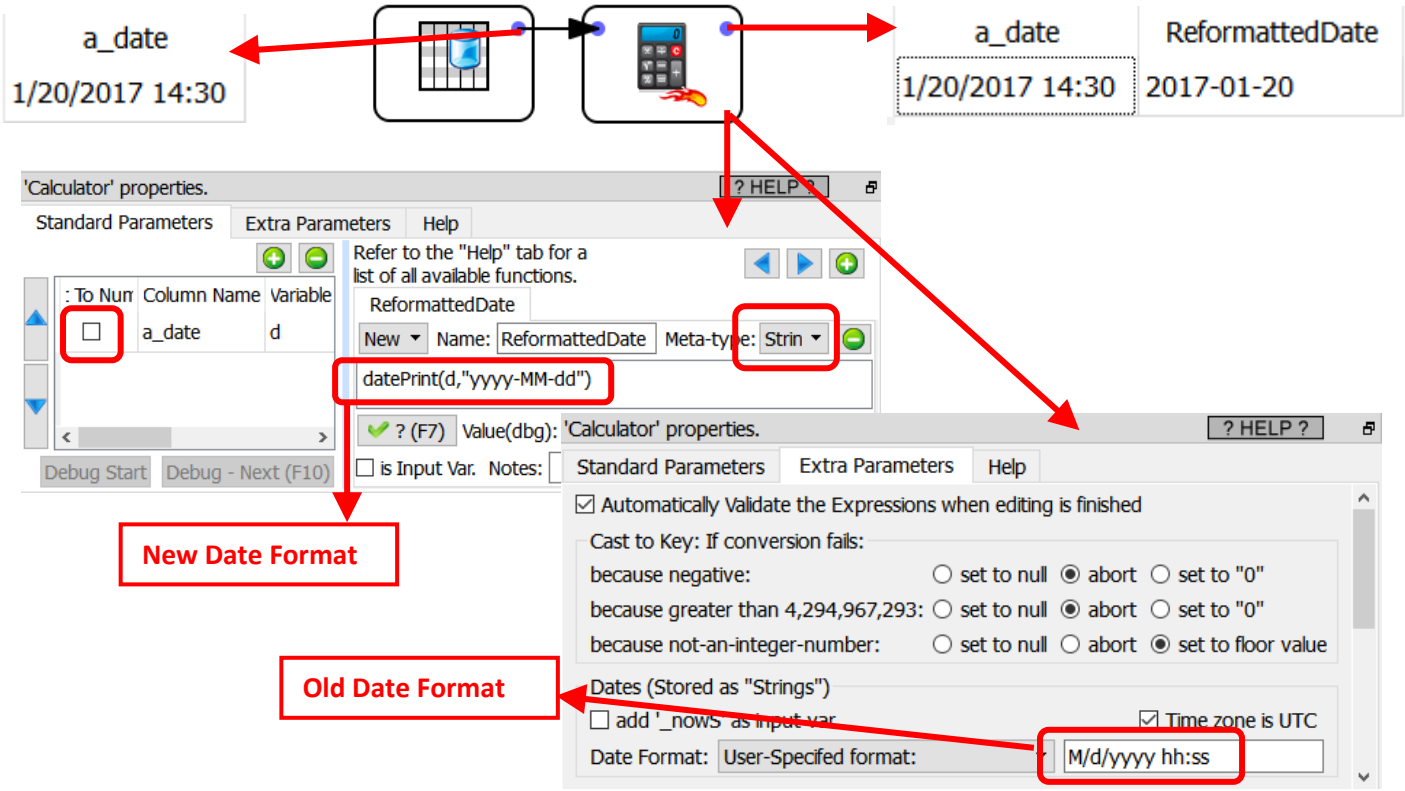


Property window:

Short description:
Reformat dates.

Long Description:

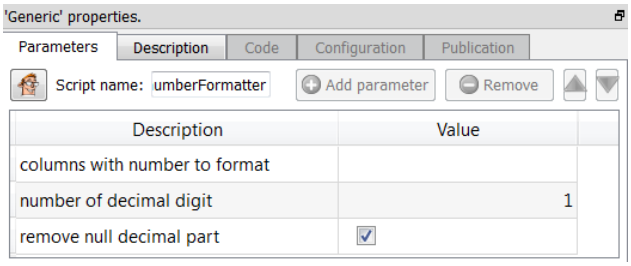
This Action is deprecated: i.e. You should rather use the Calculator action (see section 5.5.6. about the Calculator action) to reformat the dates. More precisely, you'll use, inside the Calculator action, the functions "datePrint()": Here is an example:



This action is Self-Explanatory. Please refer to section 5.1.3 to know how to specify the "*date formats*".

5.19.3. Number Formatter

Icon:



Property window:

Short description:
Format a number.

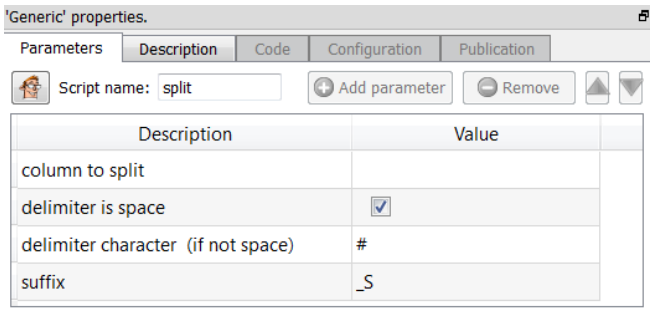
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This converts a number to string using a “standard” notation with the specified number of digits. The Number Formatter Action is a little bit more flexible than the “*conversion to string*” option of the ChangeDataType Action but it’s a lot slower. You’d rather use the ChangeDataType Action.

5.19.4. Split

Icon:

Property window:



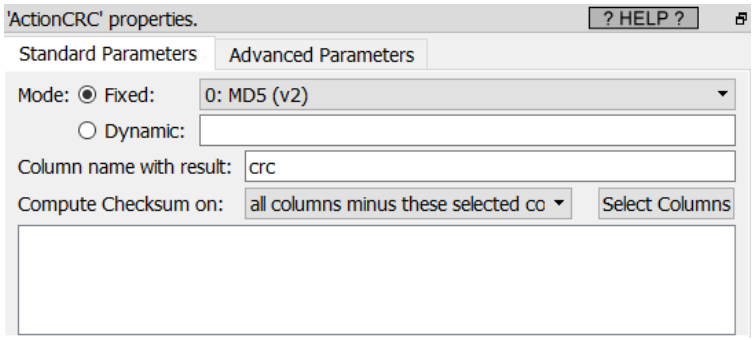
Short description:
Split one column into several columns.

Long Description:
Self-Explanatory.

5.19.5. CRC

Icon:

Property window:



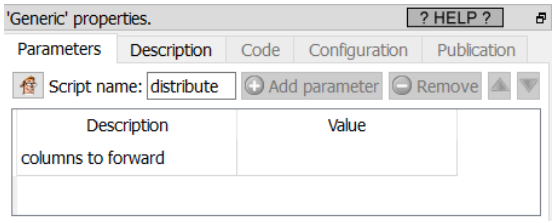
Short description:
Compute the checksum of the content of several columns.

Long Description:
Self-Explanatory.

5.19.6. Distribute

Icon:

Property window:



Short description:
Distribute evenly the rows between all output pins.

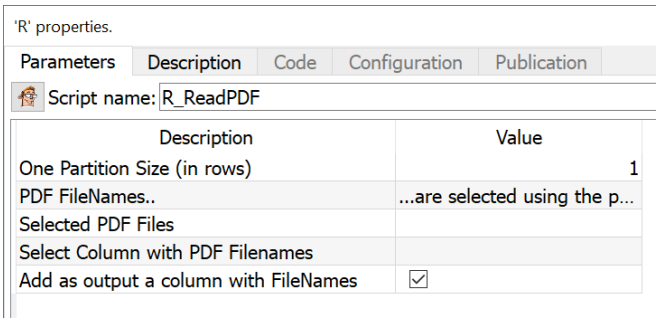
Long Description:
Self-Explanatory.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.19.7. Read PDF

Icon:

Property window:



Short description:
Read a (list of) PDF docunents

Long Description:
The Read pdf document works in two different modes:

- Files are selected manually
- Files comes from the Selected column on the first input pin

To select files manually:

1. Set “Pdf File Names …” to “...are selected using the parameter below”
2. Select the files you wish to open

To select the file from a list

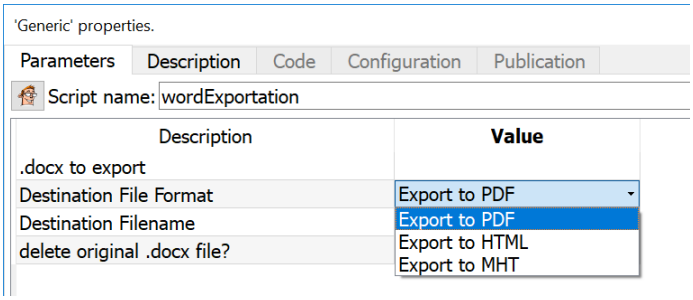
1. Set “Pdf File Names …” to “...comes from the Selected column on the first input pin”
2. Connect as input the result of a (fitlered) ReadDir

In all cases, it’s better that the pdf documents are in text format. Otherwise, this node will perfom an OCR job on image-based PDF documents to extract the text (the used OCR is “tesseract”).

5.19.8. Word Exportation to PDF

Icon:

Property window:



Short description:
Export word document to PDF, HTML, RTF or MHT

Long Description:
Prepare a table that contains:

1. A column with a list of word documents (2003+ format)
2. A column with a destination file name

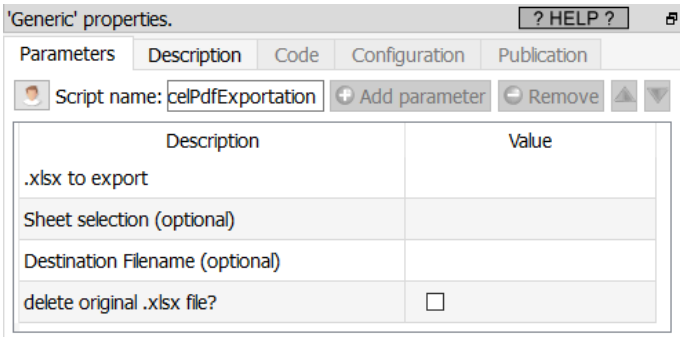
Select the format you wish to use as export type.
Run

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.19.9. Excel Exportation to PDF

Icon:

Property window:



Short description:
Export Excel .XLSX documents to PDF

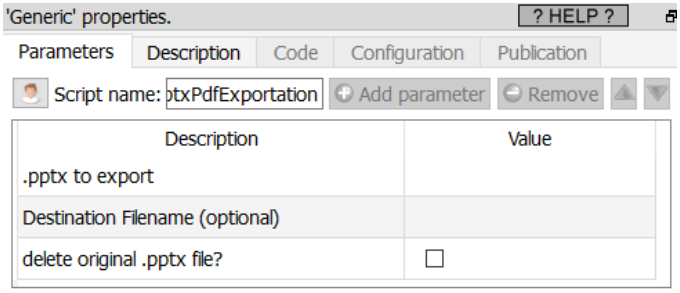
Long Description:
The "sheet selection" parameter is a column that contains, either:

- a number
- a name
- "_TIMiALL_": to export all the sheets inside the .xlsx file
- a comma-separated list of sheets to save.
  for example: "Sheet2,Sheet3" will only save the "Sheet2" and "Sheet3"

5.19.10. Powerpoint Exportation to PDF

Icon:

Property window:



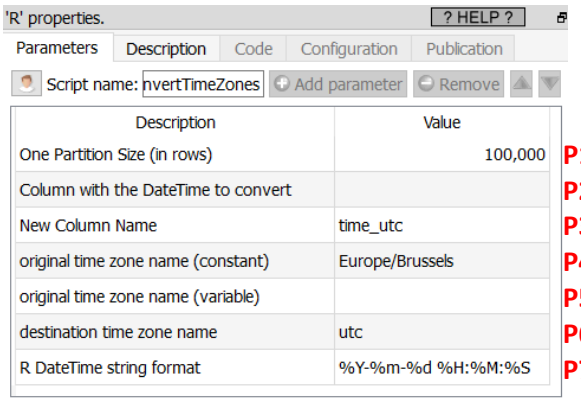
Short description:
Export Powerpoint .pptx document to PDF

Long Description:
Self-Explanatory.

5.19.11. Convert between TimeZones

Icon:

Property window:



P1
P2
P3
P4
P5
P6
P7

Short description:
Convert between TimeZones

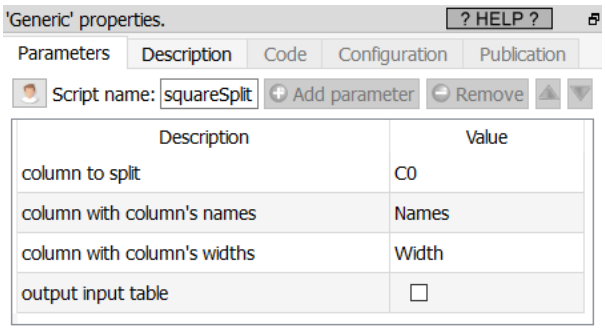
Long Description:
Self-Explanatory.
When the parameter **P5** is given, the parameter **P4** is ignored.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.19.12. Square Split

Icon:

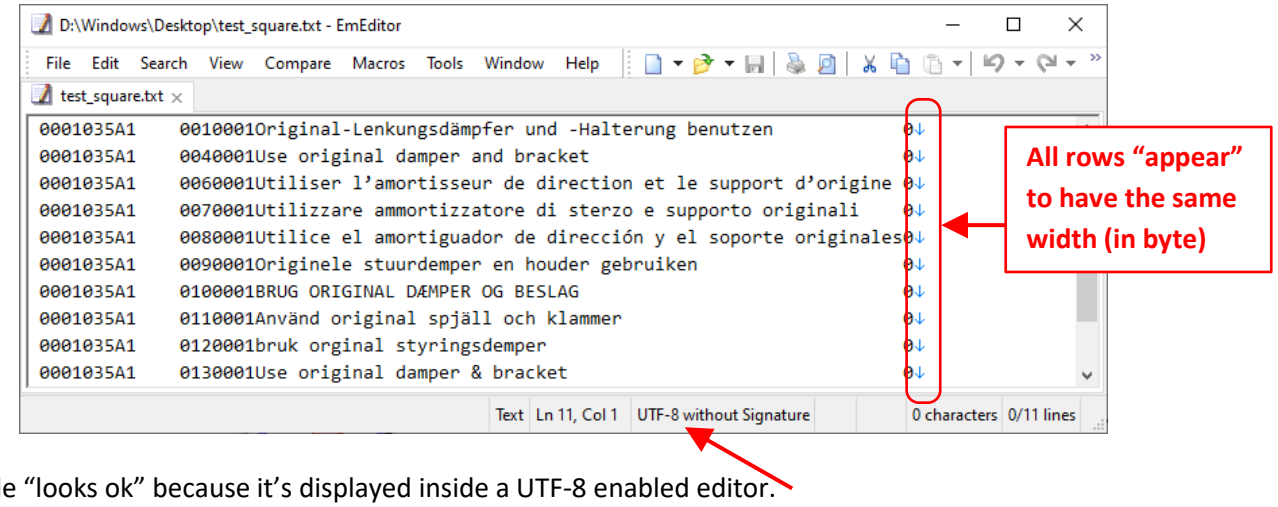
Property window:



Short description:
split a column into many constant-width columns

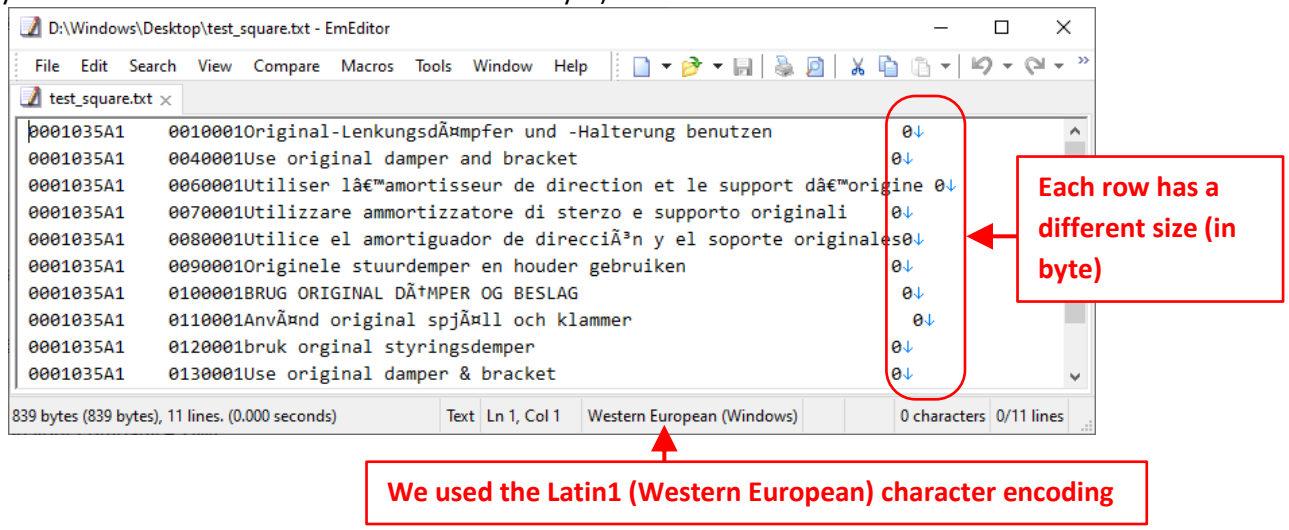
Long Description:
This Action is useful when reading “False/Erroneous” square files.

Square files are characterized by the fact that all rows have exactly the same number of bytes. Some erroneous softwares create square files that have, on each row, the same number of characters (not bytes!). When a square file is using the (now very common) UTF-8 character encoding, this “error” is catastrophous (because one UTF-8 character has a random length between 1 to 4 bytes). For example, this file “looks like” a square file because it looks like all rows have the same width (in byte):



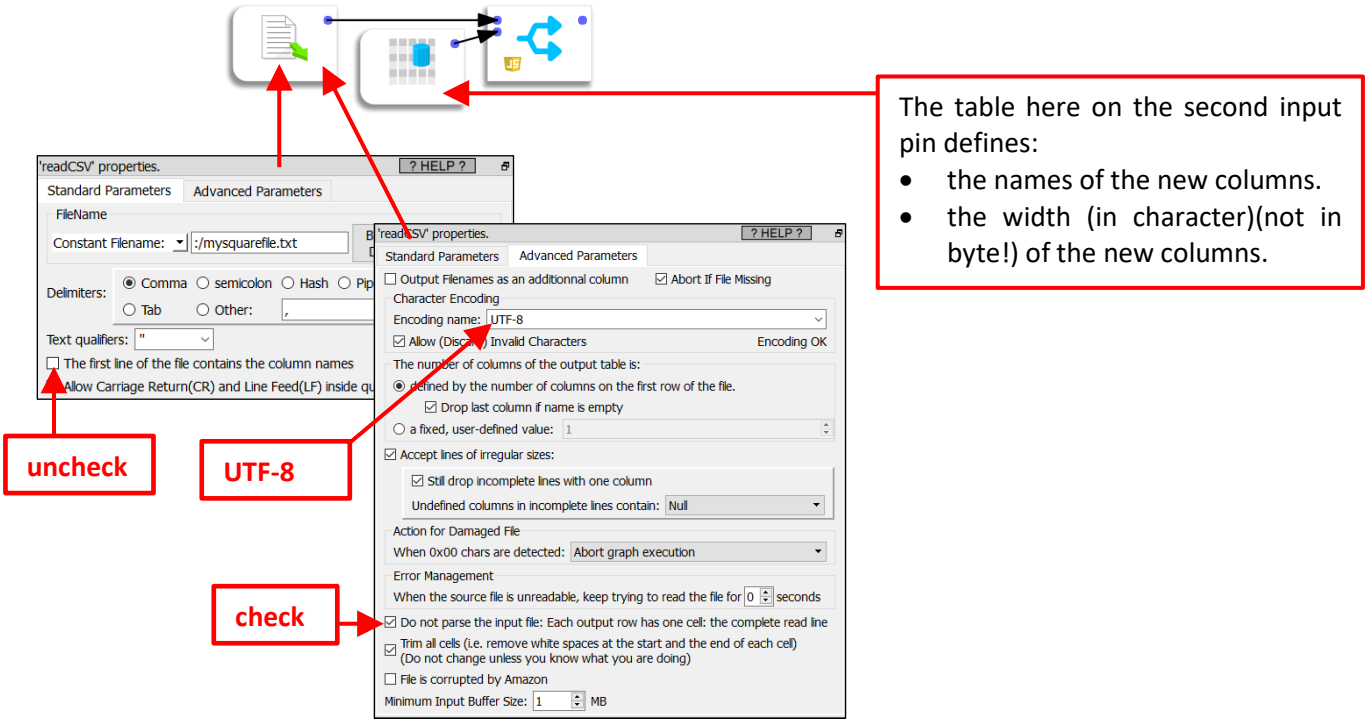
This file “looks ok” because it’s displayed inside a UTF-8 enabled editor.

Let’s open *the same file* using the Latin1 character encoding to be able to see the exact size (in byte) of each row (since, inside the Latin1-character-encoding, each character is exactly one byte, it allows to easily check if all the rows have the same width in byte):



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

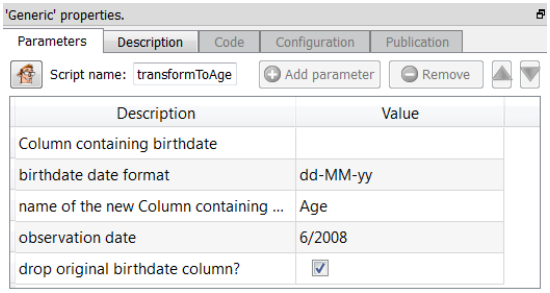
To read such erroneous square files with Anatella, use the 2 following actions:



5.19.13. Transform Birthday to Age

Icon: (this is a birthday cake with some candles)

Property window:



Short description:

This transforms a birthdate to an age in year.

Long Description:

This is a deprecated action: i.e. You should rather use the Calculator action (see section 5.5.6.) to compute the difference (in days, months or years) between two dates. More precisely, you'll use, inside the Calculator action, the following functions:

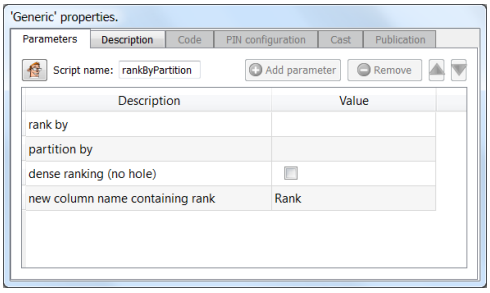
Function Name	Output
etDiffInMonth(ET1,ET2)	number of months between ET1 and ET2
etDiffInYear(ET1,ET2)	number of years between ET1 and ET2
dateDiffInMonth(date1,date2)	number of months between date1 and date2
dateDiffInYear(date1,date2)	number of years between date1 and date2
dateDiffInDay(date1,date2)	number of days between date1 and date2
dateDiffInSecond(date1,date2)	number of seconds between date1 and date2

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This transforms a birthdate to an age in year. Anatella is computing the age "as if" we were at the observation date (and not at the current date). Please refer to section 5.1.3 to know how to specify the "*date formats*". See the section (5.19.2.) to have an example of "date manipulation" with the Calculator action.

5.20. TA – Key Performance Indicators (TA=Transformations Actions)

5.20.1. Rank by Partition



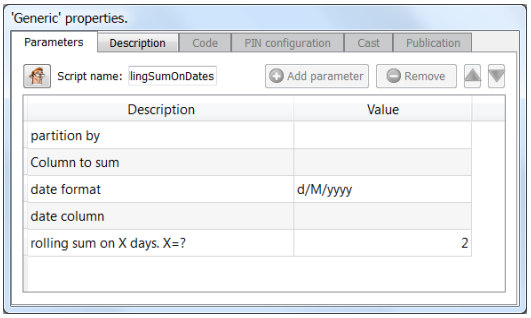
Icon:

Property window:

Short description:
Rank the values per clients to know which day had the highest, second highest, etc.

Long Description:
Rank the values per clients to know which day had the highest, second highest, etc.

5.20.2. Rolling Sum on Dates



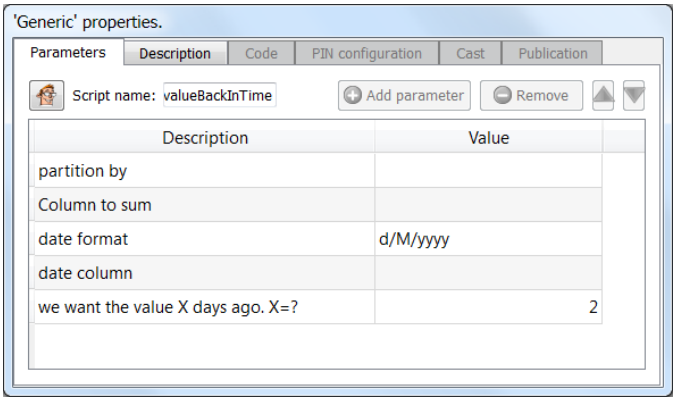
Icon:

Property window:

Short description:
TODO.

Long Description:
TODO.

5.20.3. Value Back in Time



Icon:

Property window:

Short description:
Get the value corresponding to a particular day in the past

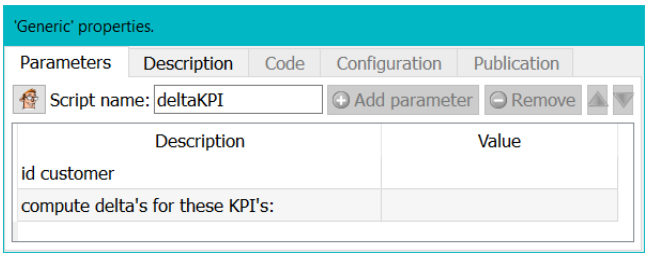
Long Description:
This node gets the value corresponding to N time periods in the past. This is often used in timeseries analysis, to get the sales level 7, 28, or 365 days before, and use this to predict future sales.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.20.4. KPI

Icon:

Property window:



Short description:
Computes delta between rows

Long Description:
This action computes, for each transaction, the difference in sales, or any other metrics you are using. The data must be sorted by ID and by DATE for this functionality to work. The KPI action will simply return the difference, positive or negative between each row. The first row is always NULL, as there is no previous transaction to compute the difference from.

Here is an example:

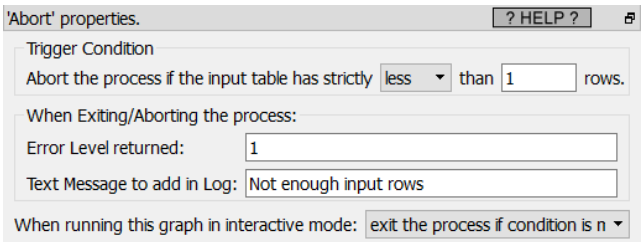
ID	date	Sale	Delta
1	1/26/2017	10	
1	3/4/2017	17	7
1	4/3/2017	11	-6
2	2/27/2017	20	
2	3/20/2017	27	7
2	4/23/2017	15	-12
2	6/8/2017	31	16
3	9/15/2016	25	
3	10/20/2016	30	5
3	11/26/2016	20	-10
3	12/16/2016	22	2

5.21. TA – System Tools (TA=Transformations Actions)

5.21.1. Abort a Graph

Icon:

Property window:

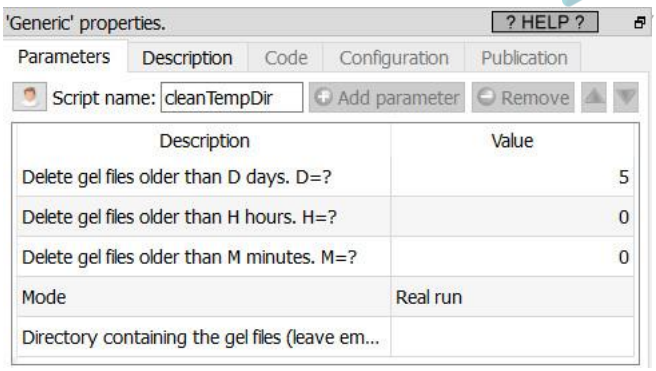


Short description:
Stop the execution of a graph.

Long Description:
Self-explanatory.
For more details on how to use this Action, see the section 5.21.10.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.21.2. Clean HD Cache Temp Dir

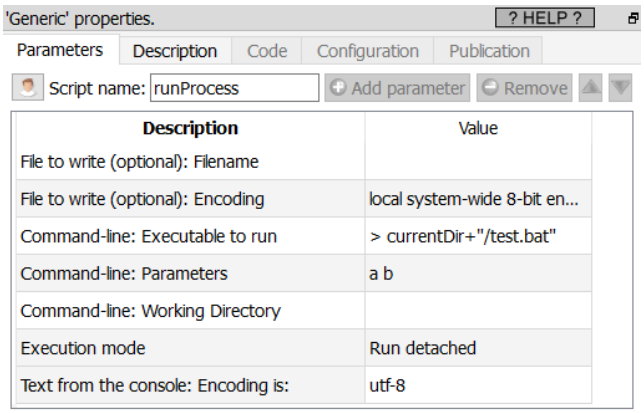


Icon:

Property window:
Short description:
Delete old .gel_anatella files.

Long Description:
When you use Anatella, you typically create many HD Caches. Over a long period of time, these HD Caches accumulate and can consume a large amount of disk space. To prevent to run "out-of-disk-space", you should run at regular interval this Action to clean your temporary directory of all the old .gel_anatella that might still be there. Please refer to section 4.8.4. to learn how to run the cleanTempDir Action at regular intervals (e.g. once every day). See the sections 4.5.1., 4.5.6. and 7.1. for more information about the "*Temporary Directory for HD Cache files*".

5.21.3. Run Processes



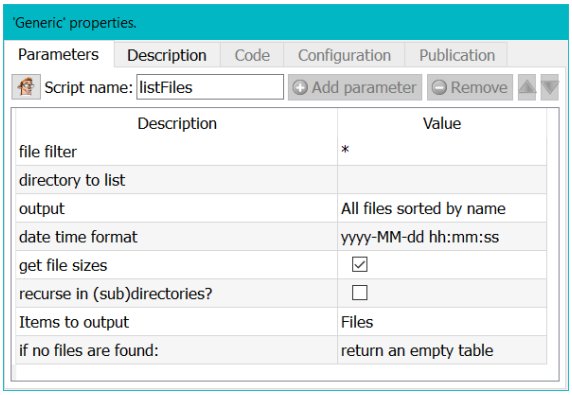
Icon:

Property window:

Short description:
Run external processes.

Long Description:
The RunProcess action runs an external process.

5.21.4. List Files



Icon:

Property window:

Short description:
Returns a list of files from a directory and its subdirectories.

Long Description:
The ListFile action does reads a directory tree and returns file names and locations. This is often used in conjunction with the FileOpen functionalities, as input pin.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

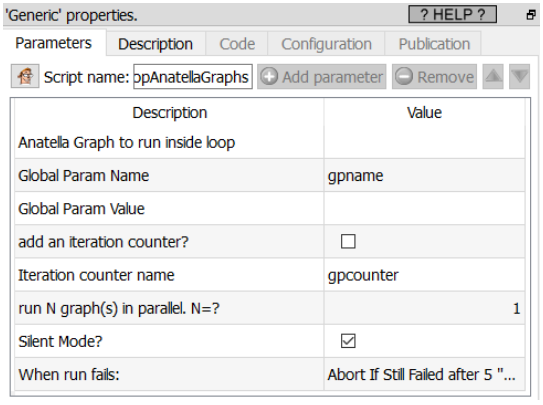
This action outputs a table with the following columns:

Column Name	Content
filePath	Full file path with file name
fileName	file name and extension only
created	creation date
modified	last motification date
lastAccessed	last access date
absolutePath	full path, without the file name
isHidden	hidden attribute, 1/0
isReadable	Read Only Attribute, 0/1
isWritable	Write Protected Attribute, 0/1
isExecutable	Executable attribute. 1/0
Size	Size of the file

The parameters are as follows:

- File filter: set the filter based on the file name or extension. For example, *2017*.* will return all the files containing 2017 in the name. This is an easy way to open files stored with dates in the name.
- Directory to list: set the root directory to analize.
- Output: set the sort criteria to return the file names:
  - All files sorted by name
  - All files sorted by time
  - All files sorted by size
  - First file (by name, by time, or by Size)
  - Last file (by name, by time or by size)
- Date-time format: set the time format to return in the columns: created, Modified, and Last Accessed
- Get File Size: a Boolean parameter to specify whether to return the file size in the table
- Recurse in (sub)directories: Boolean parameter to specify whether to browse sub directories, or only return information from the selected root directory.
- If no files are found:
  - Return an empty table: this is the default option. No error flag is generated
  - Abort anatella process with a warning: an warning is generated, but process can still run.
  - Abort anatella process with an error: the node turns red, and all is aborted.

5.21.5. Loop Anatella Graphs



Icon:

Property window:

Short description:
Loop an anatella graph on a global paramater
Long Description:

The loopAnatellaGraph action executes a given “sub-graph” (i.e. a given .anatella file) several times “in a loop”. At each iteration of the loop, the loopAnatellaGraph action sets a different value for **one** Global Parameter (whose name is, by default, “gpname”) that is used inside the “sub-graph”. The number of iterations is equal to the number of rows in the input table.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If you can execute all the different "sub-graph(s)" in any order (i.e. when there are no depencies towards previous iteration of the loop), then you can execute several "sub-graphs" in parallel (simultaneously): Just set the parameter "*Run N graph(s) in parallel. N=?*" to a higher value. This allows to easily use the several cores available inside your CPU. If you have limited RAM (or if you are on 32-bit PC), be carefull to not use a large value ("1" is our safest option).

Loops are probably the trickiest operations in anatella, but no worries, this is still manageable without any line of code.

One way to introduce "loops" inside Anatella is to "code" them in JavaScript, R, or Python as these languages offer complete freedom to code whatever you need/want. This is not always the best option as those languages will process data much slower than standard anatella components. When speed and scalability matters, it's usually better to avoid Javascript, R or Python and use the loop functionality from Anatella. You'll find more information and a detailed example on how to make loops in Anatella inside the next section 5.21.6 and sections 7.6. and 7.7.

5.21.6. Loop Anatella Graphs Advanced

Icon:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: atellaGraphsAdv | Add parameter | Remove

Description	Value
Anatella Graphs to run inside loop	GraphToRun
Global Loop Params (optional)	myFile, myDate
Test Column (optional)	
add an iteration counter?	☐
Iteration counter name	gpcounter
run N graph(s) in parallel. N=?	3
Silent Mode?	☐
When run fails:	Abort If Still Failed after 5 "...

Property window:

Short description:
Loop different Anatella graphs.

Long Description:
The loopAnatellaGraphAdv action is very similar to the loopAnatellaGraph action described in the previous section (section 5.21.5). The two differences are:

- While the loopAnatellaGraph action always run **the same** "sub-graph" (i.e. the same .anatella file), the loopAnatellaGraphAdv action can run **different** "sub-graphs" at each different iteration of the loop.
- While the loopAnatellaGraph action changes, at each iteration, the value of **one** Global Parameter (whose name is, by default, "gpname"), the loopAnatellaGraphAdv action changes, at each iteration, the value of **several** Global Parameter.

An example
The procedure to create a loop inside Anatella always involves (at least) two graphs (i.e. two .anatella files):

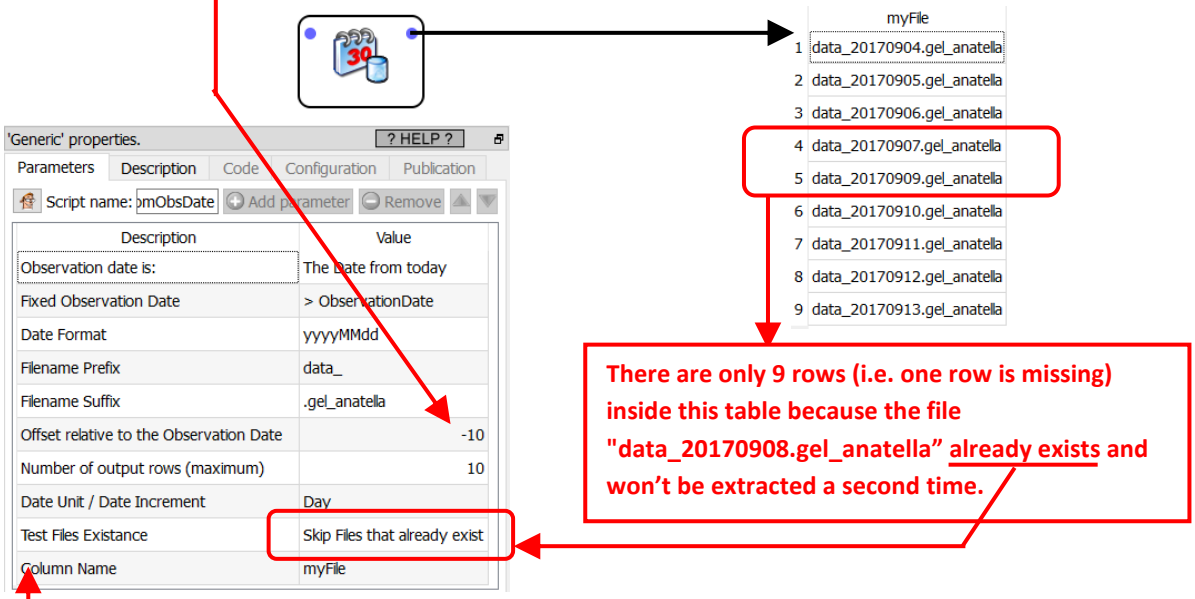
1. The first graph is the "control" graph: it's running the loop and decide how much iterations will be performed.
2. The second graph is the "inner part" of the loop (it's also named the "sub-graph"): it defines which Actions will be executed as part of the loop. This graph will be executed several times. At each execution (i.e. at each iteration of the loop), the "inner" graph is executed with a

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

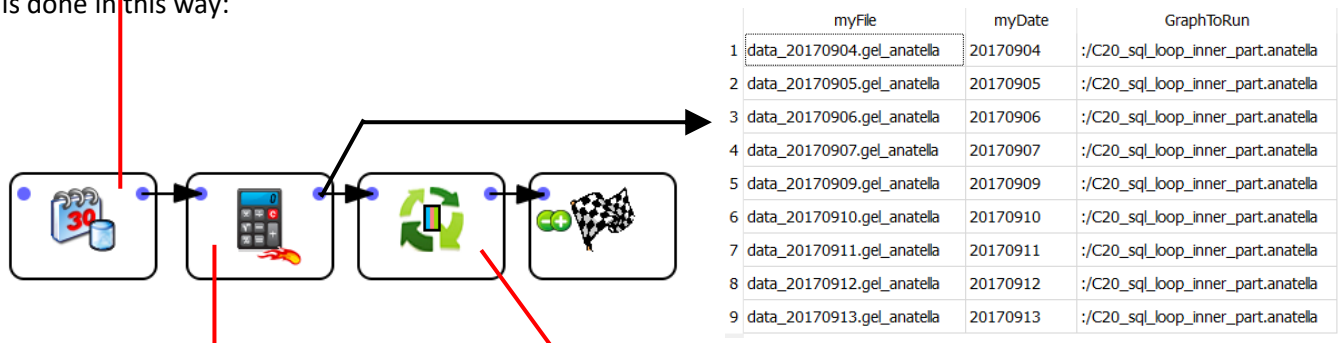
different set of values inside its "graph global parameters": see sections 4.7.1. and 5.1.5 to know more about "graph global parameters".

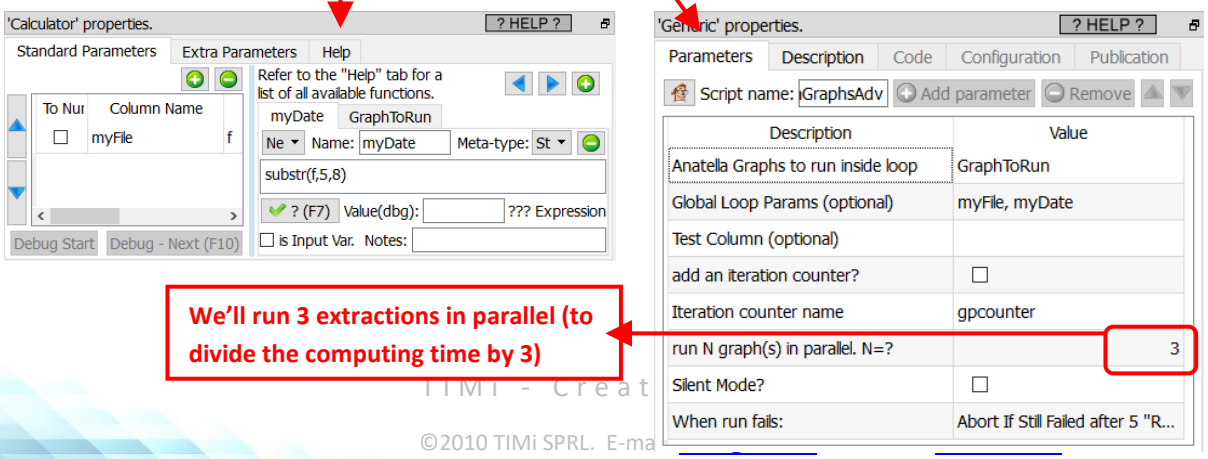
Let's assume that we want to extract, each day, the content of a table stored in a remote database and save this content inside different .gel_anatella files (one file per day). We could simply run one Anatela graph, each day, that performs the extraction. …but what's happening if this graph is not executed at one specific day? The corresponding .gel_anatella files won't be created and some data will be missing. So, a better solution would be to check if all the .gel_anatella files from the last 10 days have been correctly created. …and if some .gel_anatella files are missing, run a graph to create them.

One easy way to check if a file is present/missing is to use the fileListFromObsDate action available in the "Other" category (see section 5.23.5). For example, these parameters will generate a table that contains (for the last 10 days): the name of all the .gel_anatella files that are missings :



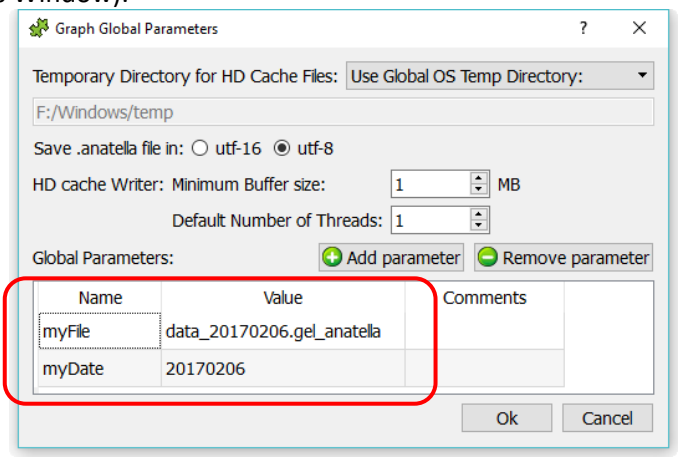
For each missing file (i.e. for each row of the above table), we'll run a sub-graph (i.e. an Anatella graph named "C20_sql_loop_inner_part.anatella") that will do the extraction and create the missing file. This is done in this way:





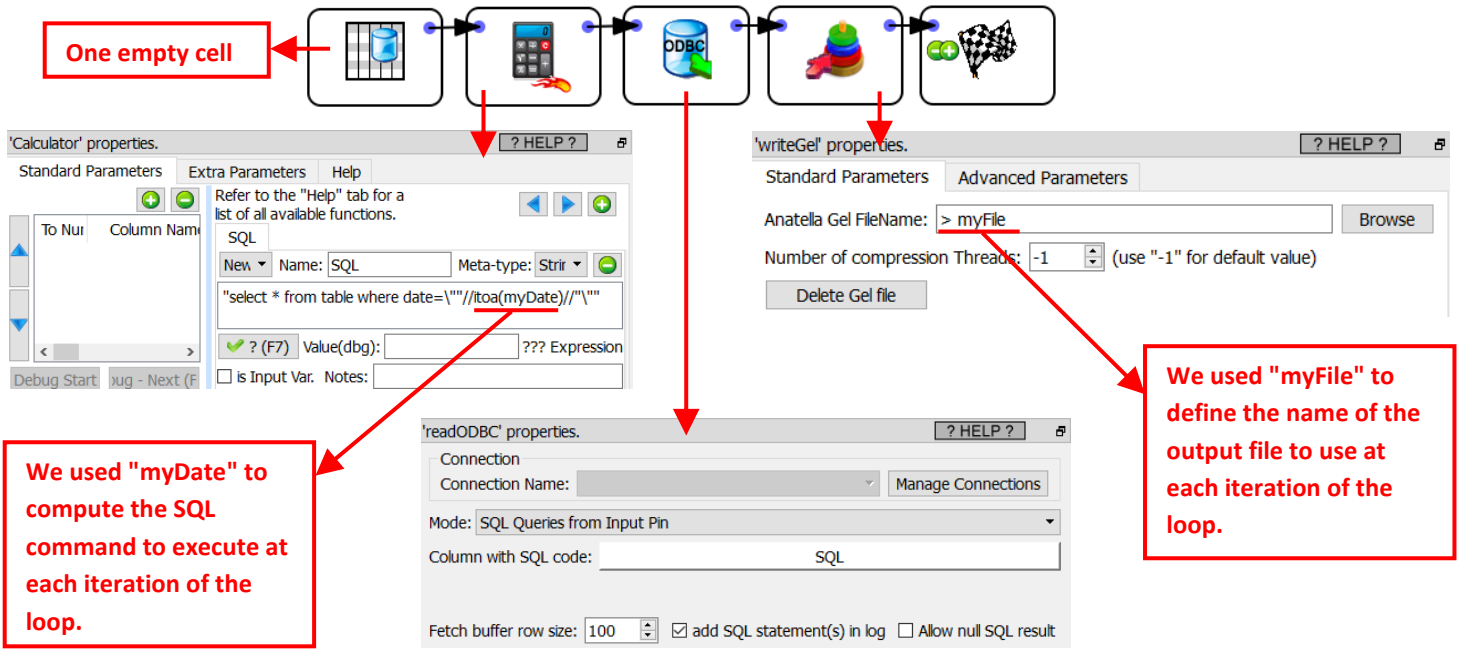
©2010 TIMi SPRL. E-ma

The “C20_sql_loop_inner_part.anatella” file is the sub-graph executed at each iteration of the loop. It contains two graph-global-parameters: “myDate” and “myFile”. You can see these 2 graph-global-parameters here (Select the option “Graph Global Parameters” inside the “Edit” drop-down menu to see this Window):



In the above screenshot, you can see that we entered some specific “default” values for the two graph global parameters (i.e. we used the “default” values “data_20170206.gel_anatella” and “20170206”). These “default” values are useful because they allow you to easily “debug” your sub-graph (when initially creating the subgraph). During the loop execution (i.e. when running the “control” graph), these values will be discarded and replaced by the values computed inside the “control” graph (More precisely: these values are extracted from the input table of the loopAnatellaGraphAdv action).

This is the “C20_sql_loop_inner_part.anatella” sub-graph:

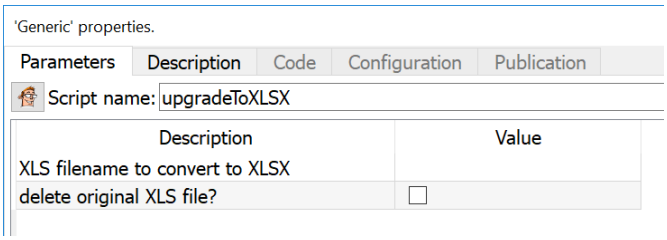


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.21.7. Upgrade to XLSX

Icon:

Property window:



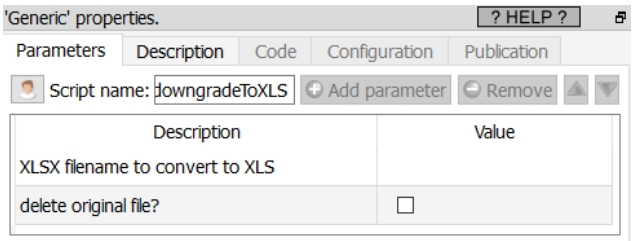
Short description:
Upgrade Excel 97 format (.xls) to Excel 2003+ format (.xlsx)

Long Description:
Prepare a table that contains a column with a list of Excel 97 documents
Connect this table to the input pin
Run

5.21.8. Downgrade to XLS

Icon:

Property window:



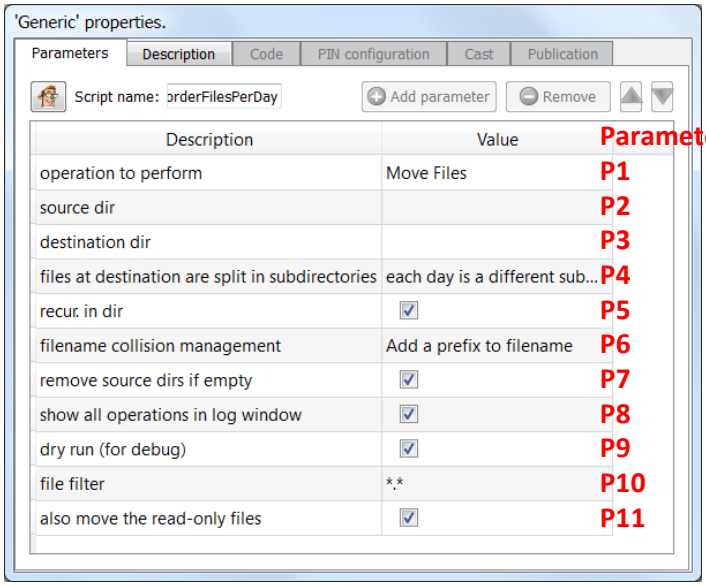
Short description:
Downgrade Excel 2003+ format (.xlsx) to the old Excel 97 format (.xls)

Long Description:
Prepare a table that contains a column with a list of .xlsx documents
Connect this table to the input pin
Run

5.21.9. Reorder Files per Day

Icon:

Property window:



Short description:
Move/Copy files from many directories to many directories.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action is typically used to gather many files located in possibly several directories and dispatch these same files to a set of destination directories. The parameters are:

- Parameter P1: Operation to perform: The 2 options are: Move or Copy Files: Self-Explanatory. Moving a file from one directory to another directory inside the same disk/partition is instantaneous, whatever the size of the file.
- Parameter P2: Source Directory: Self-Explanatory.
- Parameter P3: Destination Directory: Self-Explanatory.
- Parameter P4: Files at destination are split in subdirectories: The different options are:
  - each day is a different subdirectory
  - each month is a different subdirectory
  - each year is a different subdirectory
  - no subdirectory

  This is interesting when you received a large amount of files all located inside the same directory and you want to dispatch&re-order them into different directories based on their last modification date.
- Parameter P5: Recurse in (source) directory: When this parameter is checked Anatella also explore all the subdirectories of the "Source Directory", searching for files to move or copy.
- Parameter P6: Filename collision management: When the parameter 5 is checked, it can happen that two different files with the same name but originally located inside different directories are copied/moved inside the same destination directory. When this happens, it's named a "Filename Collision". The different options are:
  - Add a prefix to filename (This prefix is based on the name of the directory where the original "source" file was located).
  - No prefix. If collision: Abort
  - No prefix. If collision: overwrite file & continue
  - No prefix. If collision: Simply continue
- Parameter P7: Remove source directories if empty: Self-Explanatory.
- Parameter P8: Show all operations in log window: Self-Explanatory.
- Parameter P9: Dry run (for debug): When this parameter is checked Anatella does NOT move or copy any files. It just displays all the move&copy opearations, it's about to perform. This allow you to check if Anatella will correctly performs the required operations.
- Parameter P10: File Filter: Self-Explanatory: All the files that match the filter will be moved/copied.
- Parameter P11: Also move the read-only files: Self-Explanatory.

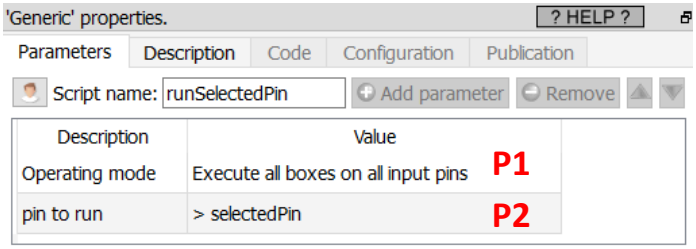
Typical usages of this Action are:

- Move the thousands of CDR files scattered randomly inside many directories to one organized directory structure where the destination directory names are based on the day the CDR file has been produced.
- Gather all the many *.ModelXML files that are composing a Multi-Class predictive model to one unique directory, to be able to use them inside one TIMiModelMerger Action automatically & easily. This functionality is used inside the "Model Factory".

5.21.10. Run Selected Pin

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Short description:
Do a "If…then…else …" inside a graph (branching instruction).

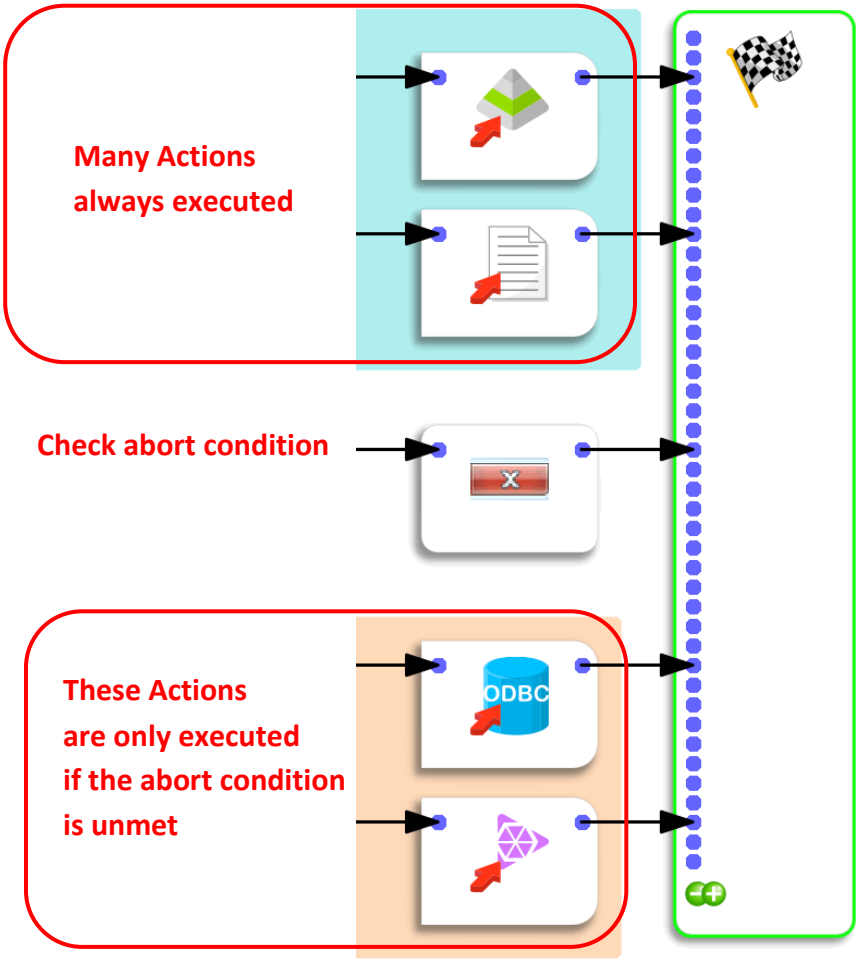
Long Description:
The parameter **P1** can have two different values :

- Execute all boxes on all input pins
- Execute the boxes on the input pin number given here below

There are, basically, three different techniques to do a "If…then…else …" inside Anatella.

"If…then…else …" technique (1): based on the "Abort" action

This first technique is very easy to use:

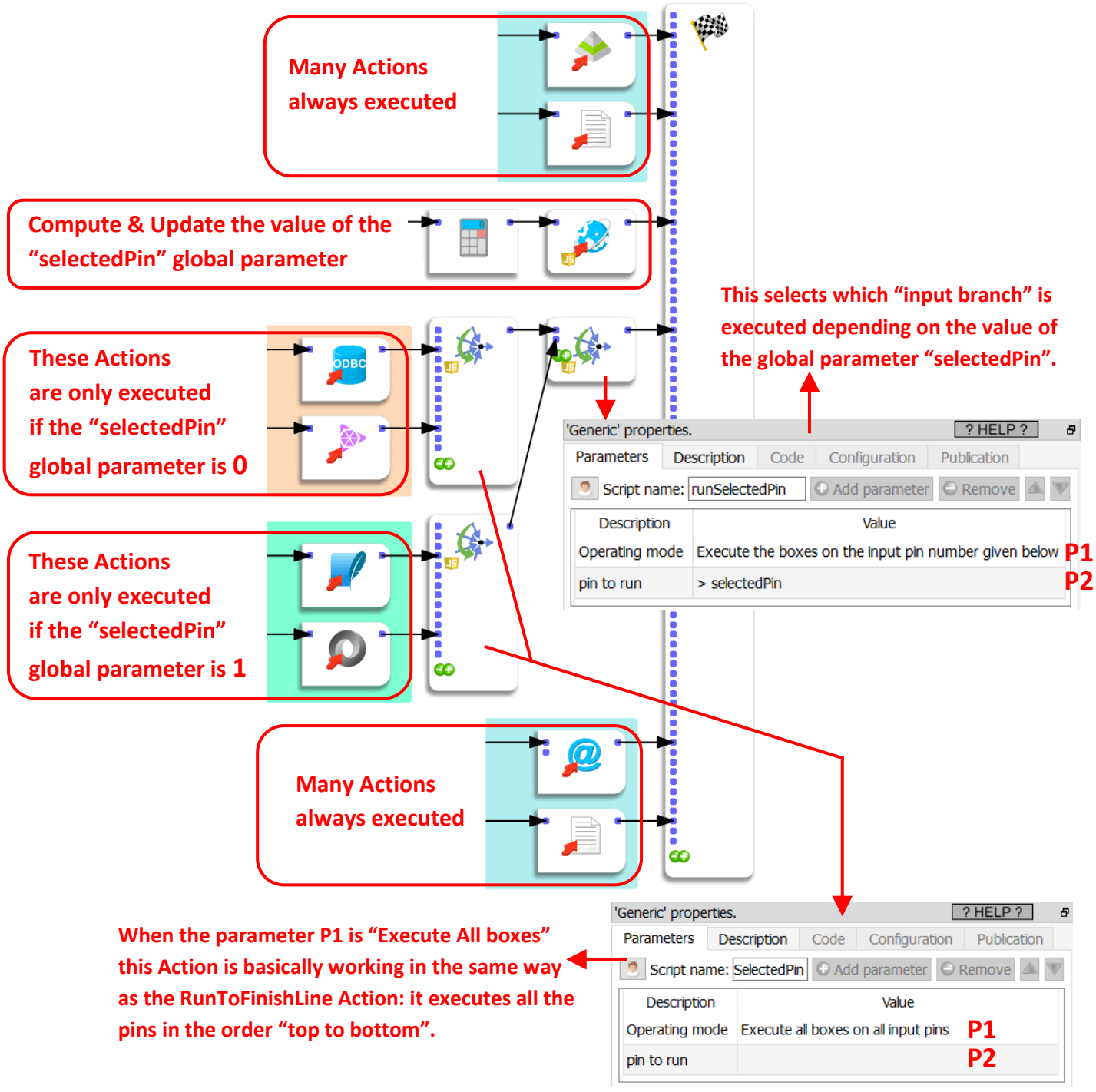


The draw back of this technique is: This technique is not very flexible: it only allows to "skip" the execution of some Actions at the end of an Anatella graph.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

"If…then…else …" technique (2): based on the "runSelectedPin" Action

Here is an example:



This technique is very flexible and very useful.

In the example above, the global parameter "selectedPin" can takes two values (0 or 1) and thus the first or the second input-branch of the runSelectedPin Action is executed. Now, nothing is stopping you from using a global parameter "selectedPin" that is in the range from 0 to 10: you'll just have 10 input-branches to the runSelectedPin action (and Anatella will execute only one of

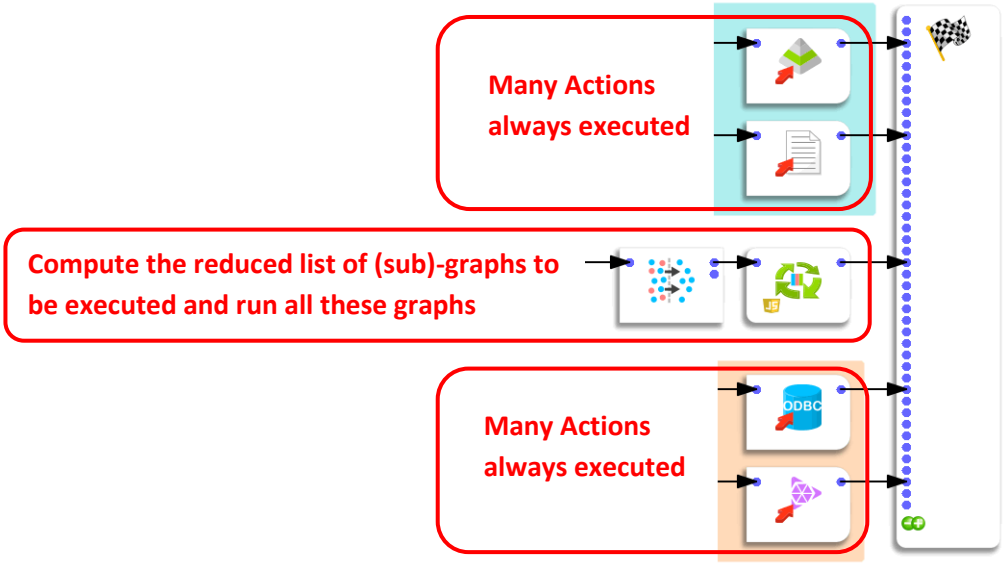
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

these branches). In this sense, the runSelectedPin action is more like a "CASE…WHEN" branching instruction (where you can have many different outomes).

The only (little) drawback of the runSelectedPin action is that you need to be familiar with the way how "global parameters" works inside Anatella. For more information on "global parameters", see the section 5.1.5.

"If…then…else …" technique (3): based on the "LoopAdv" Action

Here is an example:

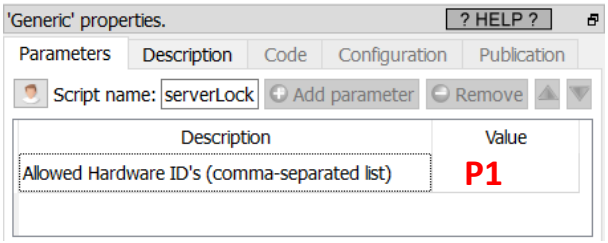


The draw back of this technique is that you need to create a new .anatella file (i.e. a new sub-graph) for each different "branch" of your "if…then…else…" instruction. This means that you might end-up with a cluterred working directory containing many different .anatella files (one file for each of your branch).

5.21.11. Lock Graph Execution to some given Servers

Icon:

Property window:



Short description:

Prevent the execution of a graph on any random server.

Long Description:

The parameter **P1** contains the list (comma-separated) of all the HardwareID where this graph is allowed to run. To get the HardwareID from the current machine, run the Action with an empty parameter **P1** and look in the log window.

To really be useful, this action should be used inside an encrypted graph: See the section 4.6.3. to kown how to encrypt an Anatella graph.

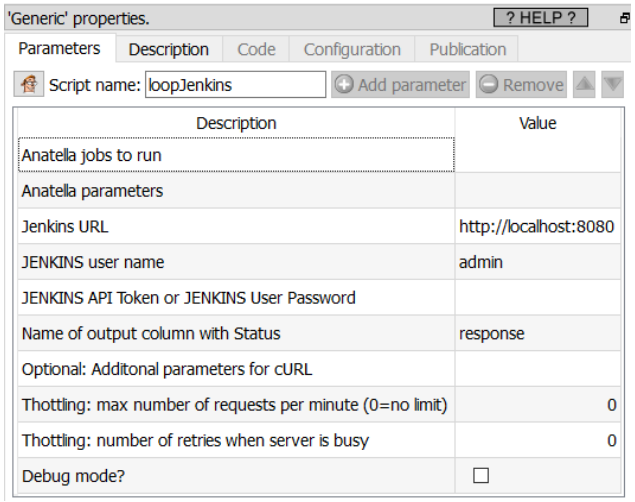
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.22. TA – Distributed Computations (TA=Transformations Actions)

5.22.1. Loop with Jenkins

Icon:

Property window:



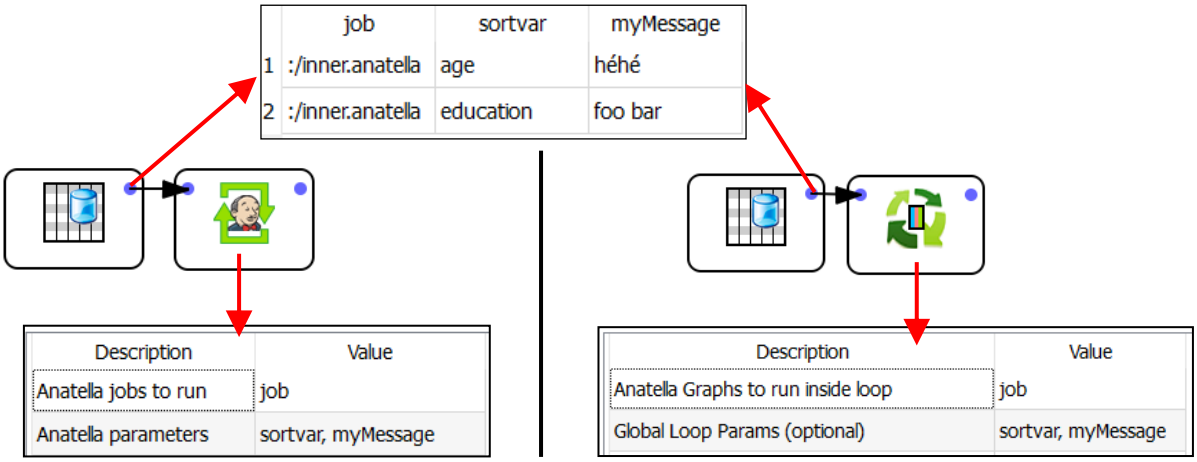
Short description:
Distributed Run of several Anatella graphs

Long Description:

The "Loop Jenkins" action runs several Anatella Graphs using Jenkins. The principle is very similar to the loopAnatellaGraphAdv action (see the section 5.21.6. to know more about this action): i.e. We also have the same "2-graphs-structure":

1. The first graph is the "control" graph: This graph is supervising the "loop" and decide how much iterations will be performed. This is inside the "control" graph that you'll typically find the "Loop Jenkins" action.
2. The second graph is the "inner part" of the loop. At each iteration of the loop, the "inner" graph is executed with a different set of values inside its "graph global parameters": See the sections 4.7.1. and 5.1.5 to know more about "graph global parameters".

Let's start with a small example: These 2 "control" graphs are performing, basically, the same thing:



More precisely: These 2 "control" graphs are running two times (i.e. during the 2 iterations of the "loop") the Anatella graph named ":/inner.anatella".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For the first run, the 2 global parameters named “sortvar” and “myMessage” are initialized to, respectively, “age” and “héhé”.
For the second run, the 2 global parameters named “sortvar” and “myMessage” are initialized to, respectively, “education” and “foo bar”.

The main difference between the above two “control” graphs is the Action that controls each iteration of the loop:

- When using the loopAnatellaGraphAdv action, the process that controls each iteration of the loop is directly the current Anatella process.
- When using the “Loop Jenkins” action, the process that controls each iteration of the loop is an exterior, third party process: it’s the Jenkins Service. This has one big advantage: When Jenkins needs to run an Anatella graph, it can freely decide to run the Anatella graph on the local PC or on any other server where Jenkins is installed&running (this is named “distributed” computations).

In technical terms, a server where Jenkins is installed&running (and that is thus able to execute an Anatella graph) is called a “node”.

This means that, if there are 10 nodes available, the “Loop Jenkins” action will run all the different Anatella graphs (that are given on the input pin) using all the 10 nodes (by default), dividing roughly the computing time by a factor of 10. That’s great! 😊

In other words, the “Loop Jenkins” action allows you to run your Anatella graphs using what is called “*distributed computations on several nodes*”.

Here are some interesting facts about the “Loop Jenkins” action:

- Before using the “Loop Jenkins” action, you must perform a very special configuration of Jenkins: See the next section (section 5.22.1.1) to know more about this subject.
- This action does not, by default, transfer files. This means that, typically, you should save your files (i.e. your .anatella files and your .gel_anatella files) inside a “network drive” (a windows shared drive and/or a HDFS drive) that is accessible from all the nodes.
- This action “adds” the Anatella graphs to execute (obtained from the input pin) to an “Execution Queue” inside Jenkins. Thereafter, Jenkins will immediately start running all the graphs inside the “Execution Queue” using all the computing power available inside all the available nodes.
- This Action **does not wait** for the completion of the execution of the graphs “added” inside Jenkins. This is a completely different behavior than the loopAnatellaGraphAdv action (see section 5.21.6.), the loopAnatellaGraph (see section 5.21.5) action or the ParallelRun Action (see section 5.3.3.) that are waiting for the executed (sub-) graphs to terminate before proceeding any further. You can use the waitJenkins action (see the section 5.22.2. for more details about this action) to wait for the completion of the graphs “added” inside Jenkins.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- You can query at any time the status of the graphs inside the "Execution Queue" inside Jenkins using the queryJenkins action (see section 5.22.3. for more details about this Action).

5.22.1.1. First Time Jenkins Configuration required before using the "Loop Jenkins" action.

This is a "one time" configuration process. Before following the instructions given in this section, you must first install Jenkins: See the section 4.8.2. to have more information on how to install Jenkins on your system.

If you already setup&configured the Anatella Jenkins-Wizard (see the steps 18 and 19 from section 4.8.2.), you don't need to follow the instructions given inside this section (i.e. the Jenkins-Wizard already made everything automatically for you).

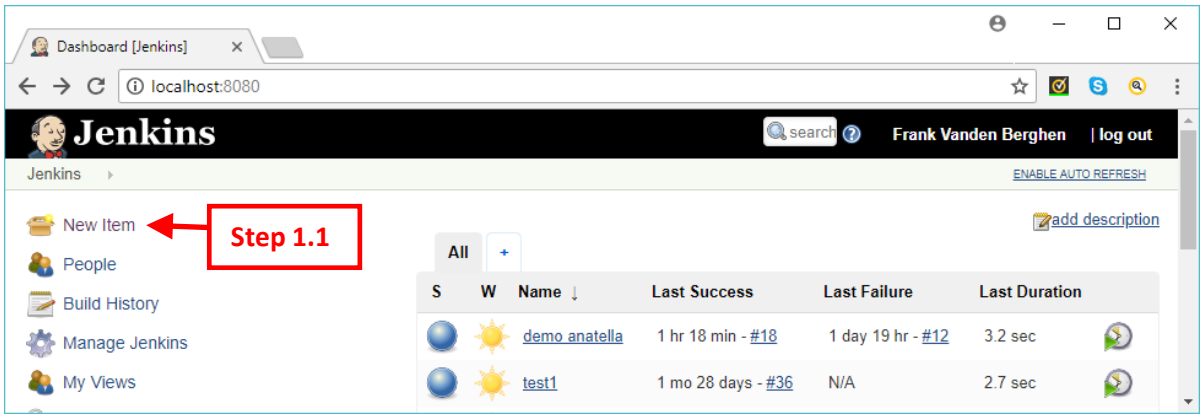
All instructions are illustrated using screenshots.

This is a 3 steps process:

Step 1: Add (and configure) inside Jenkins the Project named "AnatellaRunner".

Since Anatella v2.52, you don't need to manually perform the "step 1": This is done automatically for you when you configure the integratin of Jenkins inside Anatella: You can thus safely skip this first step and go directly to "Step2".

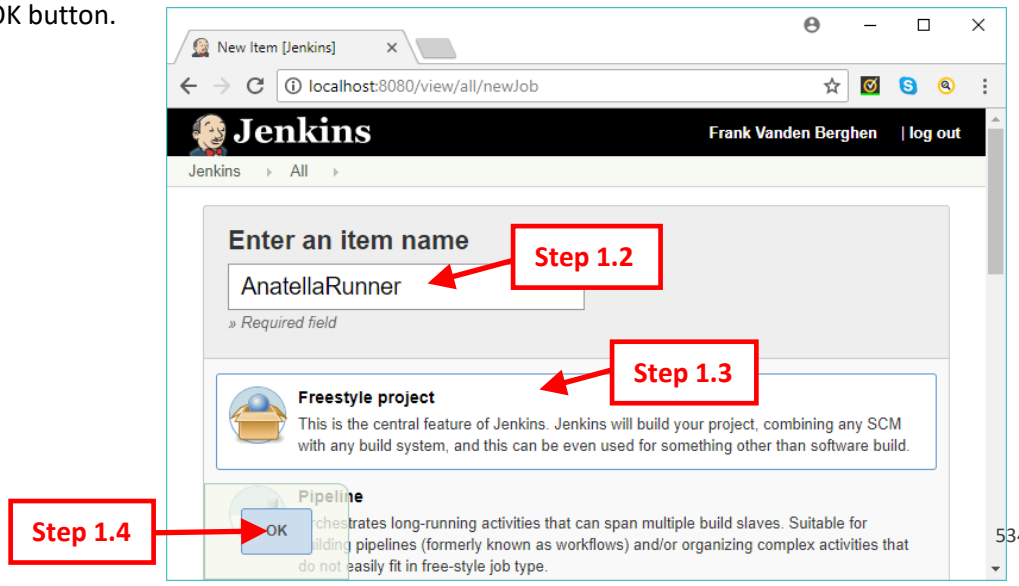
1.1. Open a browser and click on "New Item" in the top left corner of the "Jenkins" home page:



1.2. Give the name "AnatellaRunner" to your "project".

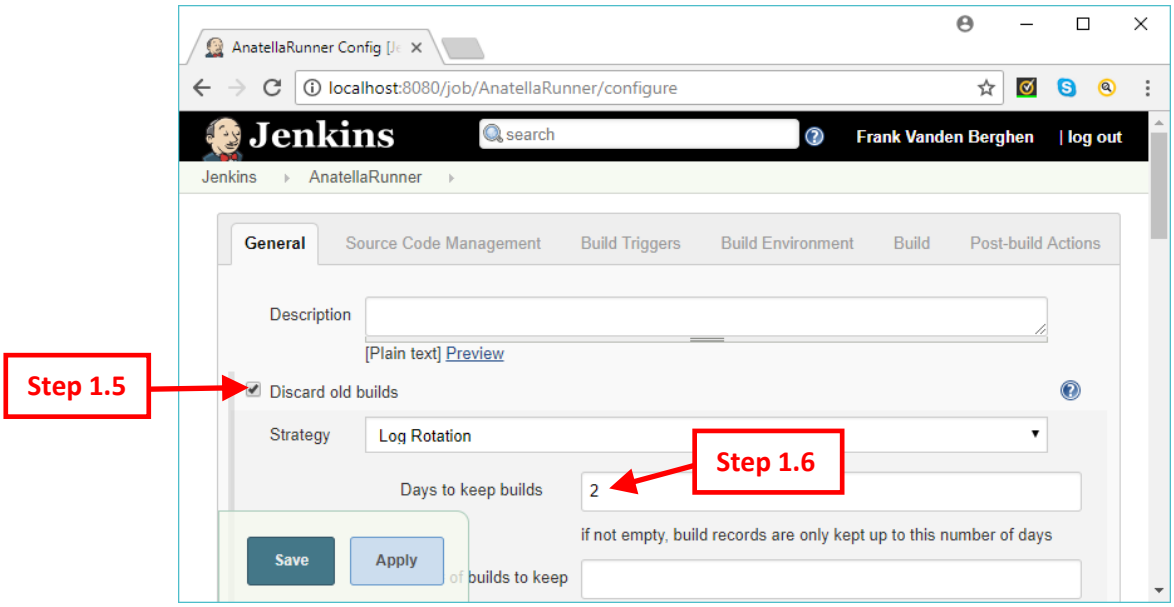
1.3. Select the option "*Freestyle Project*".

1.4. Click the OK button.



1.5. Click (Enable) the "*Discard old builds*" option.

1.6. Enter "2" as the parameter "*Days to keep builds*".



Selecting "2" as the parameter "*Days to keep builds*" means that Jenkins will keep for 2 days the informations about the success or failure of your Anatella jobs.

The queryJenkins action (see section 5.22.3.) and the waitJenkins action (see section 5.22.2.) both need this information in order to work properly.

This means that, if you use the queryJenkin action to get the status (success or failure) of a "old" job that was launched more than 2 days ago, the information required to answer your question won't be available anymore inside Jenkins (since the parameter "*Days to keep builds*" is set to the value "2" by default). Thus, the queryJenkin action will be unable to answer your request (a similar failure will happen for the waitJenkin action).

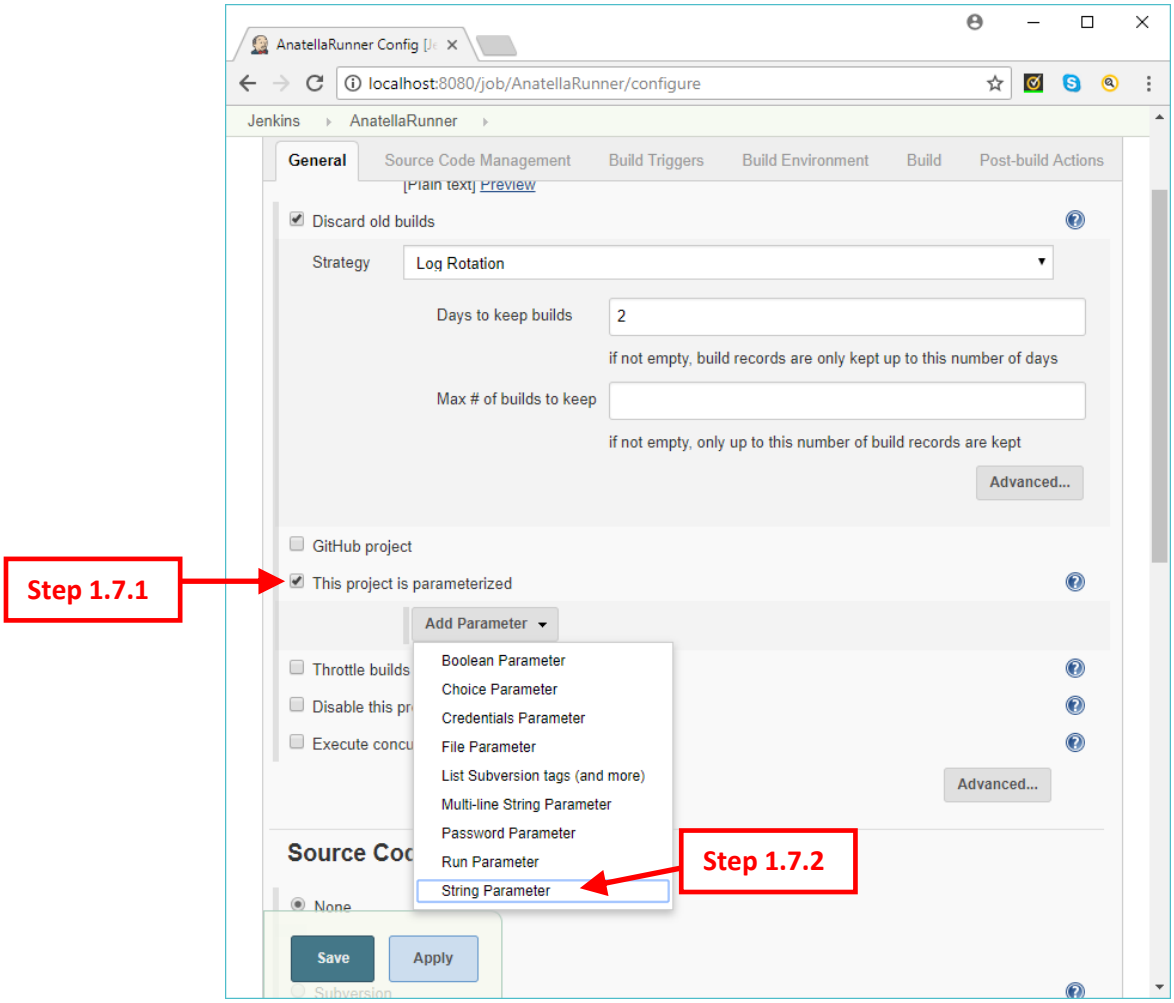
Thus, if you have very long running jobs (e.g. several days), I suggest that you select a larger value for the parameter "*Days to keep builds*".

1.7. Add the three "String" parameters named "AnID", "AnFile" and "AnParameters":

1.7.1. Click (Enable) the "*This project is parametrized*" option.

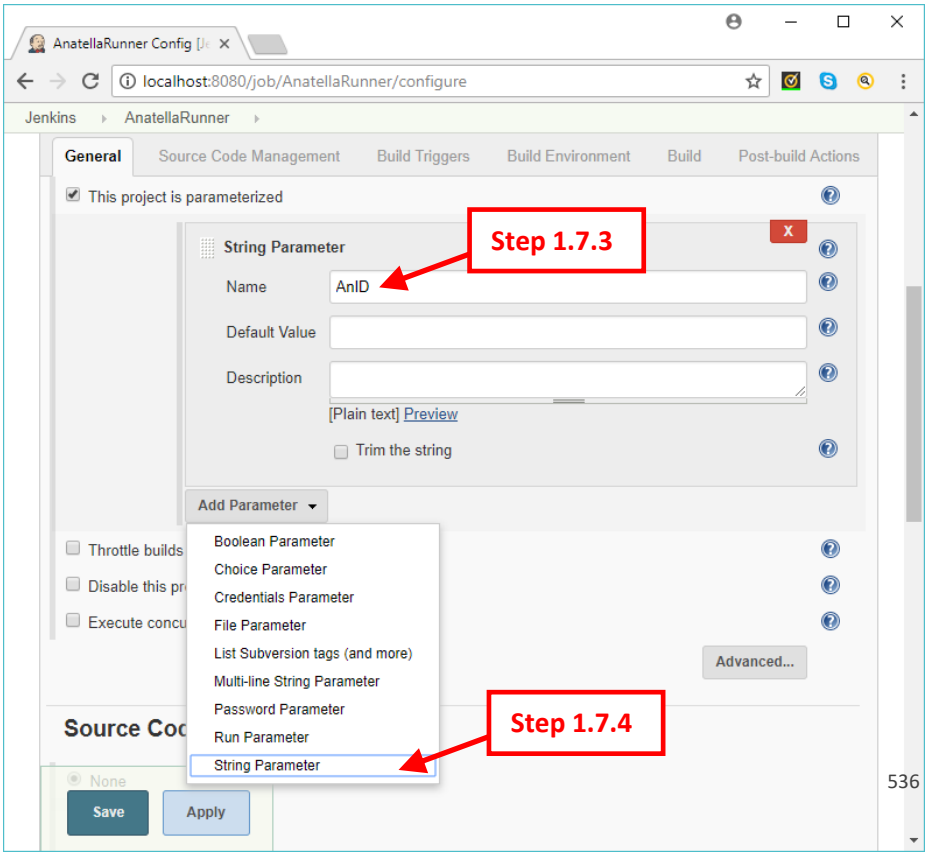
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1.7.2. Click the ComboBox named "*Add Parameter*" and select "*String Parameter*"



1.7.3. Enter "AnID" as the name of our first "String" parameter.

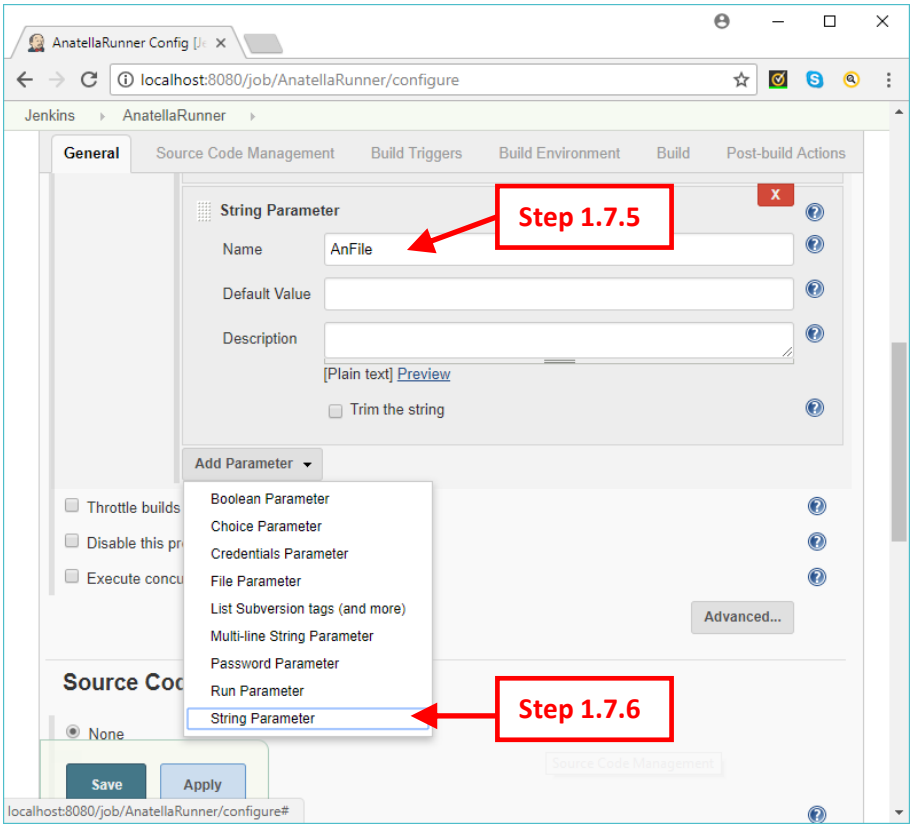
1.7.4. Click again the ComboBox named "*Add Parameter*" and select the option "*String Parameter*"



The "AnatellaRunner" project must have the 3 parameters named "AnID", "AnFile" and "AnParameters". It's important that **the "AnID" parameter is the first parameter** in the list.

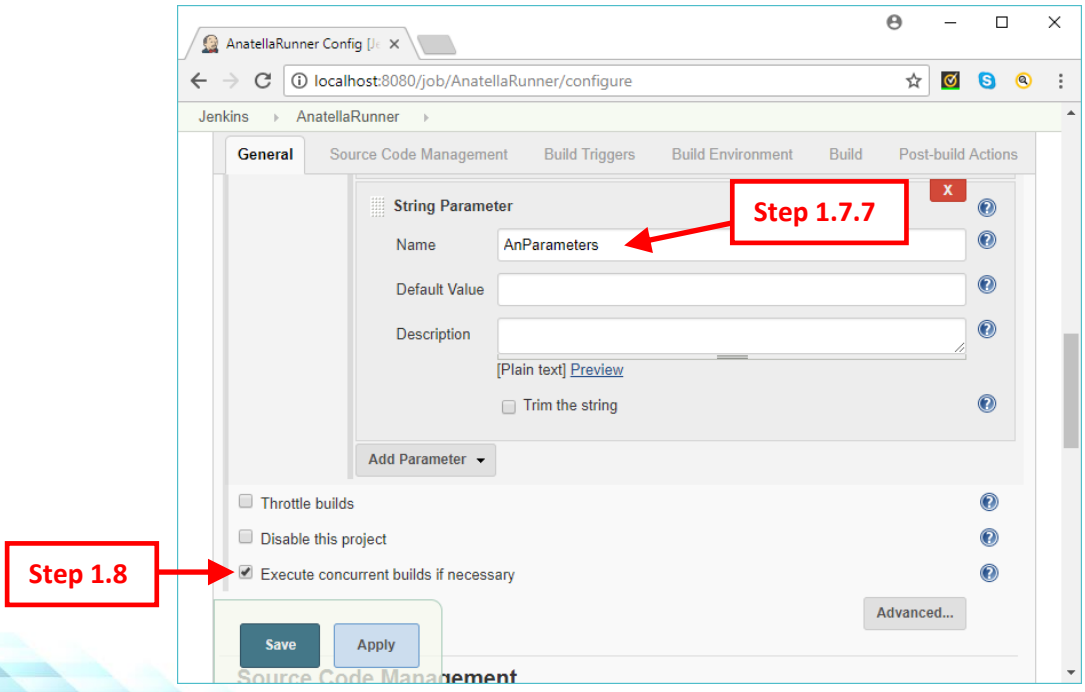
1.7.5. Enter "AnFile" as the name of our second "String" parameter.

1.7.6. Click again the ComboBox named "*Add Parameter*" and select the option "*String Parameter*"

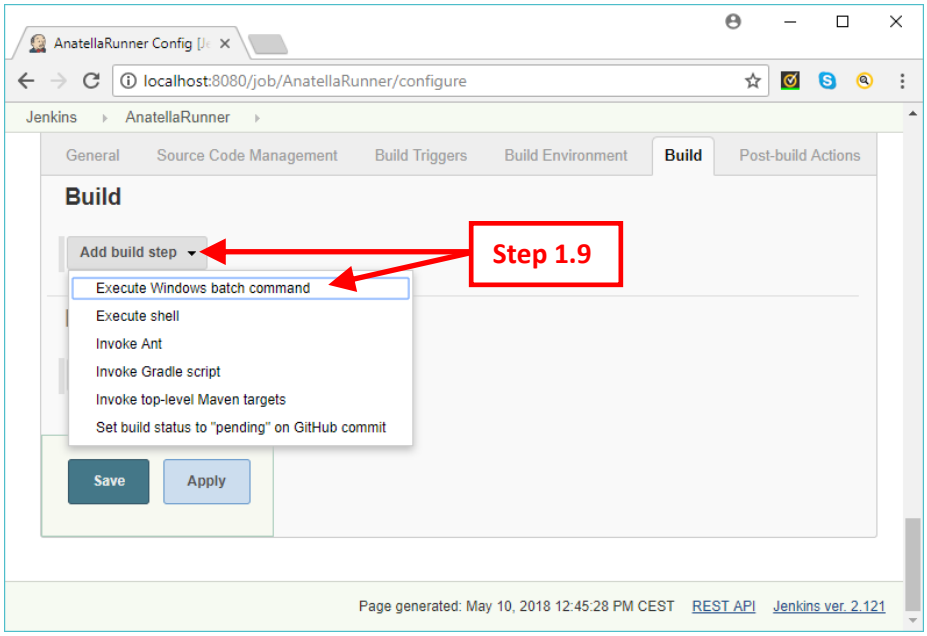


1.7.7. Enter "AnParameters" as the name of our third "String" parameter.

1.8. Click (Enable) the "*Execute concurrent builds if necessary*" option.



1.9. Scroll to the bottom of the webpage and, inside the section "*Build*", click on the "*Add build step*" ComboBox and select the "*Execute Windows batch command*" option:

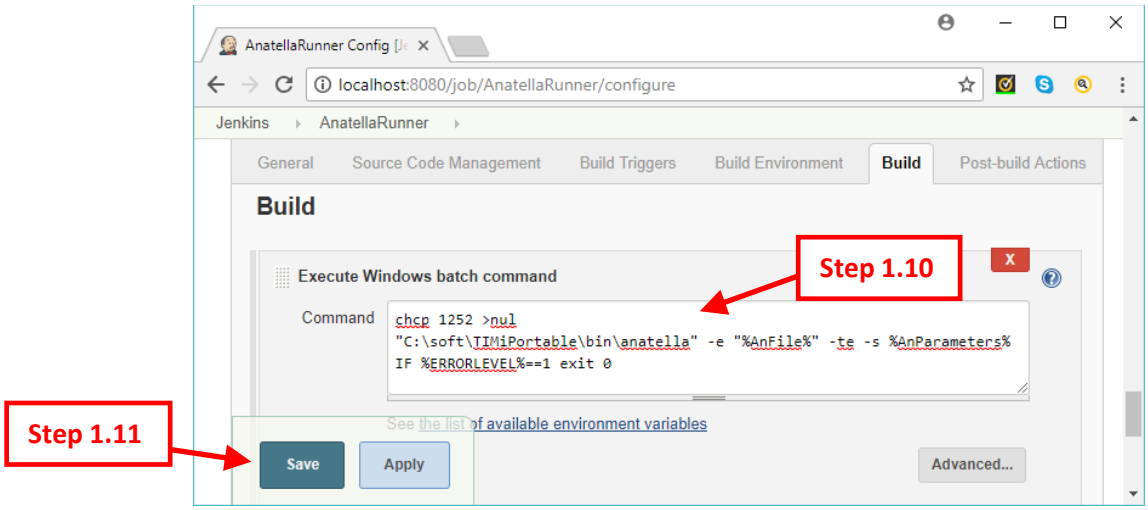


1.10. Inside the "Command" text field, enter the following:

```
chcp 1252 >nul
"C:\soft\TIMi\bin\anatellaConsole" "%AnFile%" %AnParameters%
IF %ERRORLEVEL%==1 exit 0
```

If the "AnatellaConsole.exe" executable is inside another directory that the directory "`C:\soft\TIMi\bin`", please adjust the "Command" text field accordingly.

1.11. Click on the "Save" button.

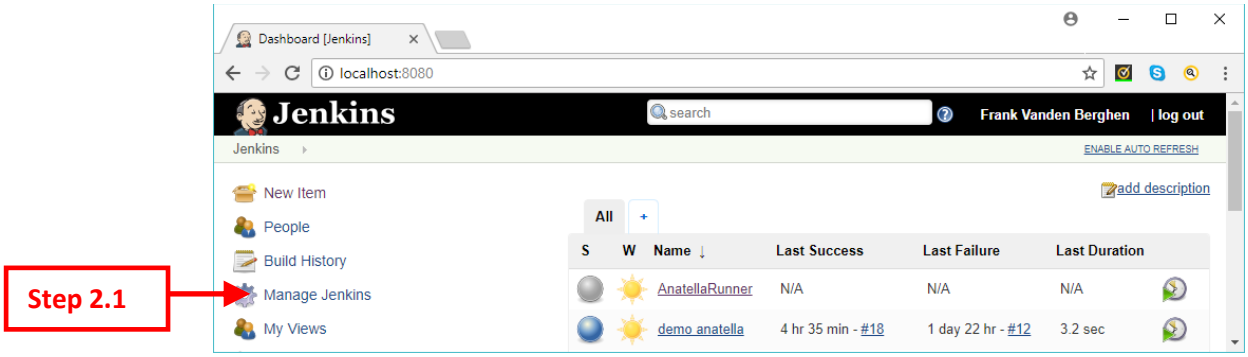


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

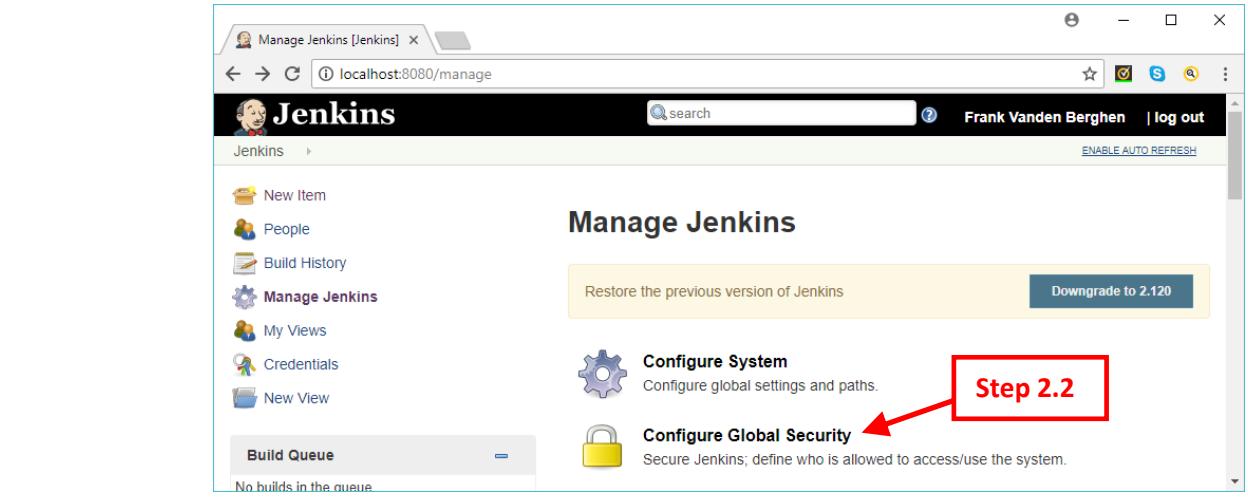
Step 2: Make sure that the CSRF Protection is active

This is a usefull security measure to prevent unauthrorized execution of Jenkins jobs (through the Cross Site Request Forgery -CSRF- exploit). By default, the "CSRF Protection" is already active, but it's still better to check, just to be sure.

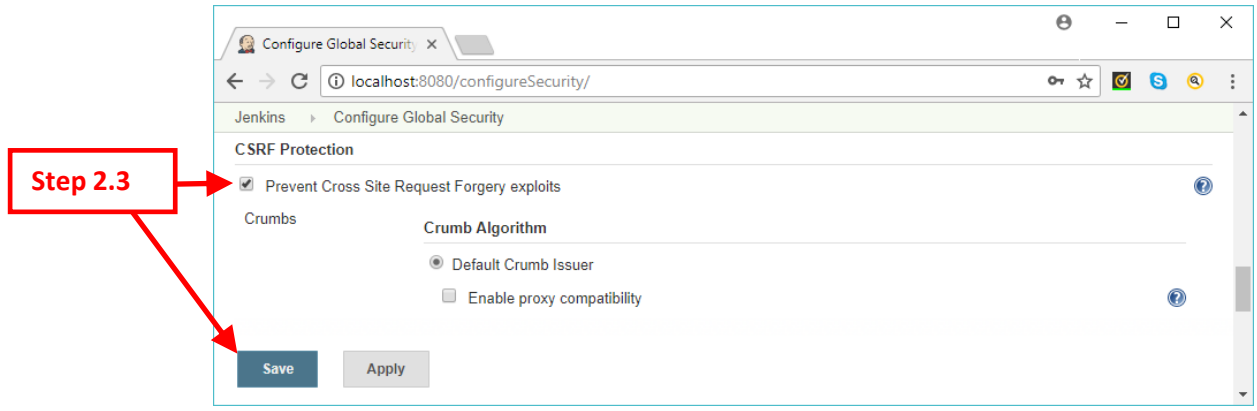
2.1. Inside the Jenkins Home page, click on "*Manage Jenkins*":



2.2. Click on the "*Configure Global Security*" button:



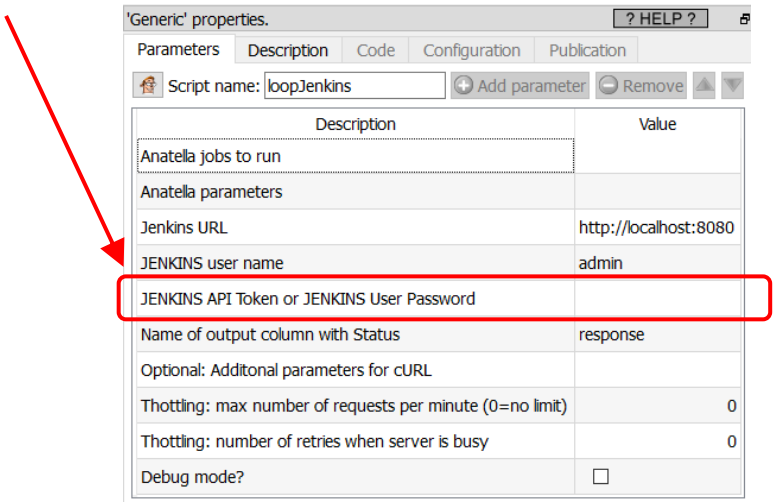
2.3. Make sure that the "*Prevent Coss Site Request Forgery exploits*" option is checked (i.e. enabled). If it's not, enable the option and then click on the "*Save*" button.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

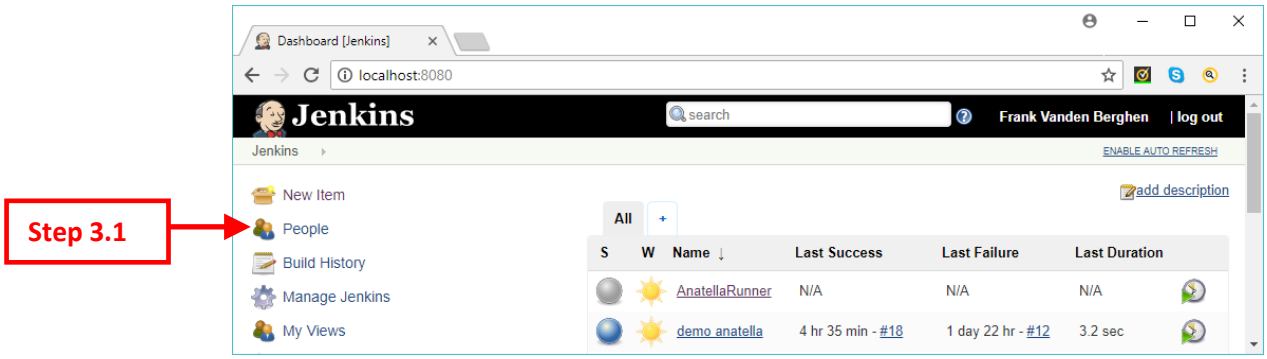
Step 3: Optional Step: Aquire an "API Token".

To use the "Loop Jenkins" action (see section 5.22.1.), the waitJenkins action (see the section 5.22.2.) or the queryJenkins action (see the section 5.22.2.), you must enter your Jenkins password. For example:

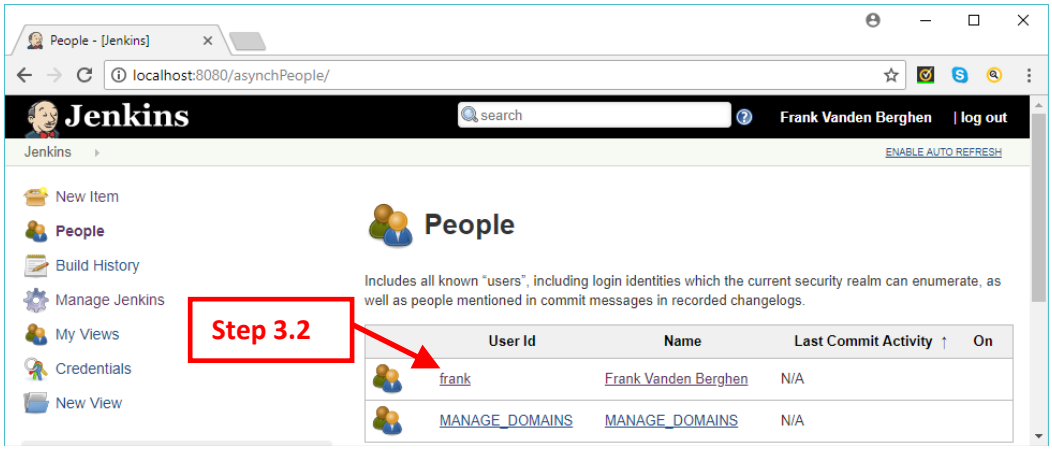


Alternatively, instead of entering you user password, you can enter an "API Token". The procedure to obtain your "API Token" is:

3.1. Inside the Jenkins Home page, click on "*People*":

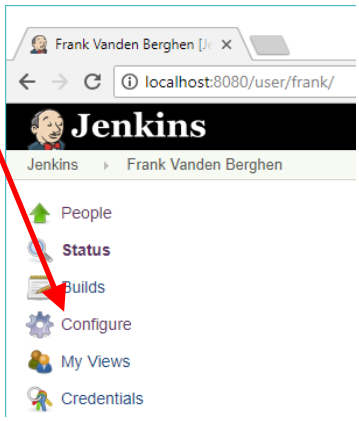


3.2. Click on your "*User ID*" (or on the "*User ID*" of the user that will be using Jenkins from Anatella):

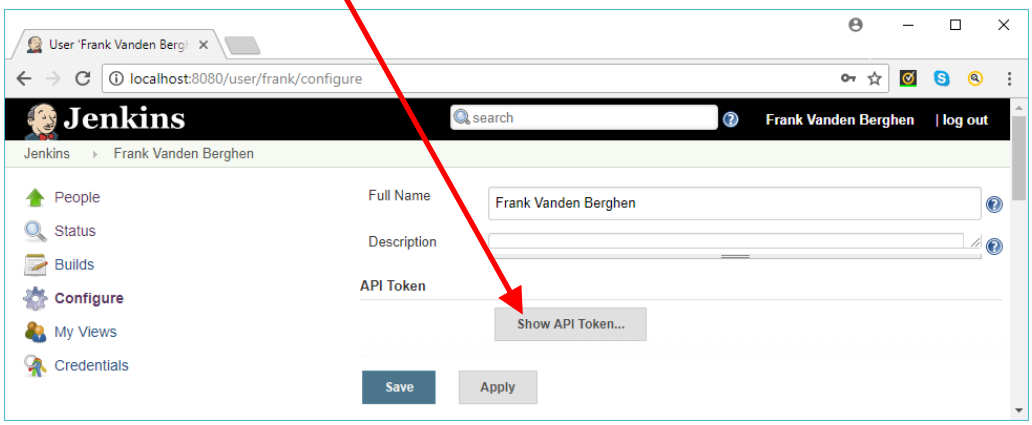


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

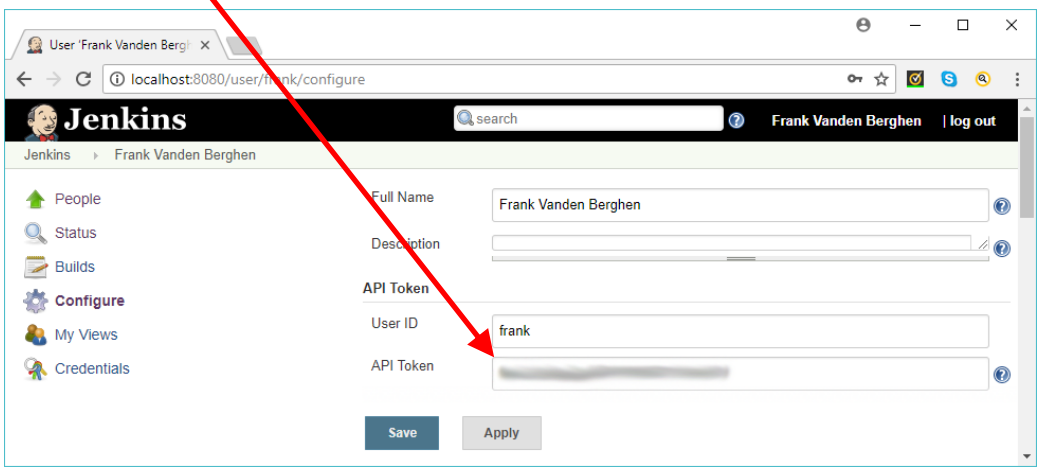
3.3. Click on "*Configure*":



3.4. Click on the "*Show API Token*" button:



3.5. Your "*API Token*" is here:



5.22.1.2. (optional) Configure the Jenkins execution Queue

The "Loop Jenkins" action adds the Anatella graphs to execute (obtained from the input pin) to an "Execution Queue" inside Jenkins. Thereafter, Jenkins will immediately start running all the graphs inside the "Execution Queue" using all the computing power available inside all the available nodes. To configure the jenkins "Execution Queue", see the section 4.8.9.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.22.2. Wait for the termination of Jenkins Jobs

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: ompletionJenkins | Add parameter | Remove

Description	Value	
Anatella JobID's to query inside Jenkins	_Anatella_Job_ID	**Parameter P1**
Poll Jenkins Server every X seconds. X=?	0.5	**Parameter P2**
Display Statistics inside the log window eve...	0	**Parameter P3**
Jenkins URL	http://localhost:8080	
JENKINS user name	admin	
JENKINS API Token or JENKINS User Passw...		
Optional: Additonal parameters for cURL		
Thottling: max number of requests per mi...	0	
Thottling: number of retries when server is...	50	
Debug mode?	☐	

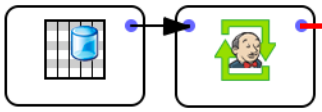
Short description:
Wait for the completion of Anatella Jobs running in Jenkins

Long Description:

The "Loop Jenkins" action (see section 5.22.1) **does not wait** for the completion of the execution of the Anatella graphs running inside Jenkins. You can use the waitJenkins action to wait for the completion of the graphs running inside Jenkins.

The combination of the loopJenkins action and the waitJenkins action allows to obtain a behavior similar to the loopAnatellaGraphAdv action (see section 5.21.6.), the loopAnatellaGraph (see section 5.21.5) action or the ParallelRun Action (see section 5.3.3.) that are waiting for the executed (sub-) graphs to terminate before proceeding any further.

All the Anatella jobs that were added inside Jenkins using the loopJenkins action are uniquely identified through an "_Anatella_Job_ID". This "_Anatella_Job_ID" is given as output of the loopJenkins action: For example:



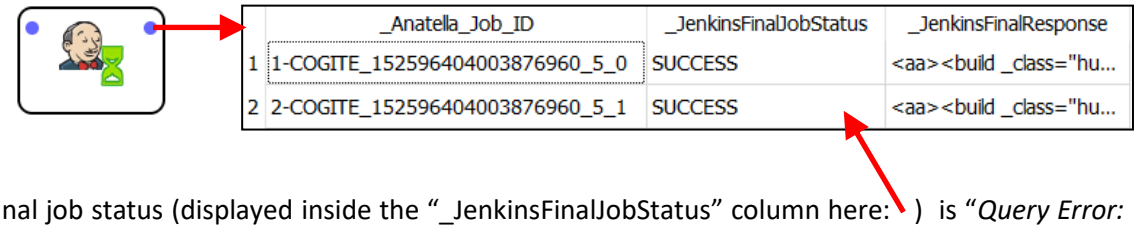
	job	sortvar	myMessage	_Jenkins_Response	_Anatella_Job_ID
1	:/inner.anatella	age	héhé	HTTP/1.1 201 Crea...	86-COGITE_152589657217808640_5_0
2	:/inner.anatella	education	foo bar	HTTP/1.1 201 Crea...	87-COGITE_152589657217808640_5_1

The waitJenkins action is waiting for the completion of all the jobs whose "_Anatella_Job_ID" is given on the input pin (specified using the **Parameter P1**). If Anatella sees an un-finished job X, it re-contacts Jenkins at regular intervals (this procedure is named "polling"), to know if the job X is finally finished. The delay between 2 "contacts" (i.e. the "polling interval") is the **Parameter P2** (in seconds). After Y "contacts" (i.e. after Y "polling" - Y is **Parameter P3**), Anatella will display inside the low window some statistics: For example (when **Parameter P3**>0), you'll see inside the Log Window:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
Running... (10/5/18 16:54:04)
#Total=2 ; #Pending=2 (#Running=1 ; #InQueue=1) ; #Complete=0 (#Success=0 ; #QueryFailure=0 ; #Fail=0)
#Total=2 ; #Pending=2 (#Running=1 ; #InQueue=1) ; #Complete=0 (#Success=0 ; #QueryFailure=0 ; #Fail=0)
#Total=2 ; #Pending=1 (#Running=1 ; #InQueue=0) ; #Complete=1 (#Success=1 ; #QueryFailure=0 ; #Fail=0)
#Total=2 ; #Pending=1 (#Running=1 ; #InQueue=0) ; #Complete=1 (#Success=1 ; #QueryFailure=0 ; #Fail=0)
#Total=2 ; #Pending=0 (#Running=0 ; #InQueue=0) ; #Complete=2 (#Success=2 ; #QueryFailure=0 ; #Fail=0)

Interactive Run: Success!(finished at 10/5/18 16:54:36 after 31.68 seconds - Peak Memory Consumption: 144 MB)
```

As ouput of the waitJenkins action, you get a table that gives you the final status of the monitored jobs. For example:



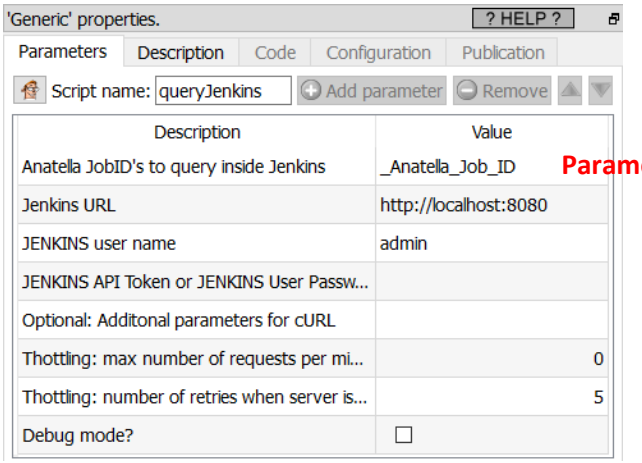
	_Anatella_Job_ID	_JenkinsFinalJobStatus	_JenkinsFinalResponse
1	1-COGITE_152596404003876960_5_0	SUCCESS	<aa><build _class="hu...
2	2-COGITE_152596404003876960_5_1	SUCCESS	<aa><build _class="hu...

If the final job status (displayed inside the "_JenkinsFinalJobStatus" column here:) is "*Query Error: Build Information is too old and has been discarded by Jenkins*", it means that you have to increase the value of the Jenkins configuration parameter named "*Days to keep builds*": See the section 5.22.1.1 (and more precisely the step 1.6) to kown how to do so.

5.22.3. Query Jenkins

Icon:

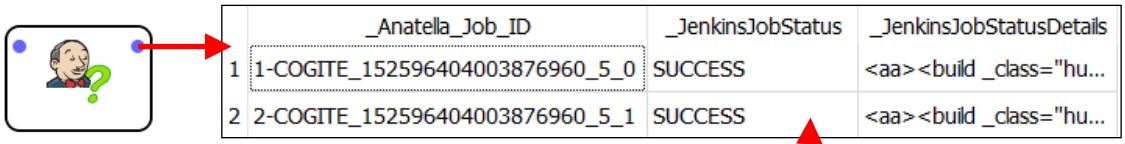
Property window:



Short description:
Query the Status of Anatella Jobs running in Jenkins

Long Description:

The queryJenkins action returns the status (BUILDING, IN QUEUE, SUCCESS, …) of all the jobs whose "_Anatella_Job_ID" is given on the input pin (specified using the **Parameter P1**).



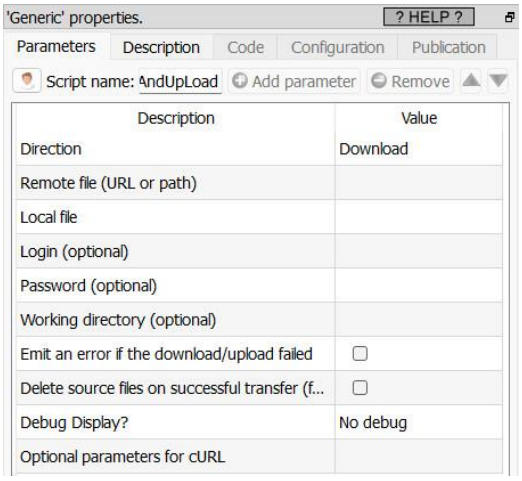
	_Anatella_Job_ID	_JenkinsJobStatus	_JenkinsJobStatusDetails
1	1-COGITE_152596404003876960_5_0	SUCCESS	<aa><build _class="hu...
2	2-COGITE_152596404003876960_5_1	SUCCESS	<aa><build _class="hu...

If the job status (displayed inside the "_JenkinsJobStatus" column here:) is "*Query Error: Build Information is too old and has been discarded by Jenkins*", it means that you have to increase the value of the Jenkins configuration parameter named "*Days to keep builds*": See the section 5.22.1.1 (and more precisely the step 1.6) to kown how to do so.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23. TA – Cloud services (TA=Transformations Actions)

5.23.1. Down and Uploads



Icon:

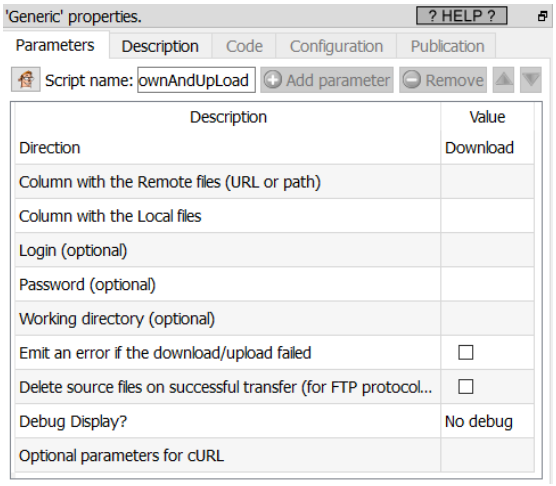
Property window:

Short description:
Download and upload files

Long Description:

The Down and Up load action uses a cURL command to download/upload one file from/to remote servers, via HTTP, FTP, HTTPS, SSH, SCP, and SFTP protocols (and more). You'll find more information in the next section (5.23.2).

5.23.2. Multiple Down and Uploads



Icon:

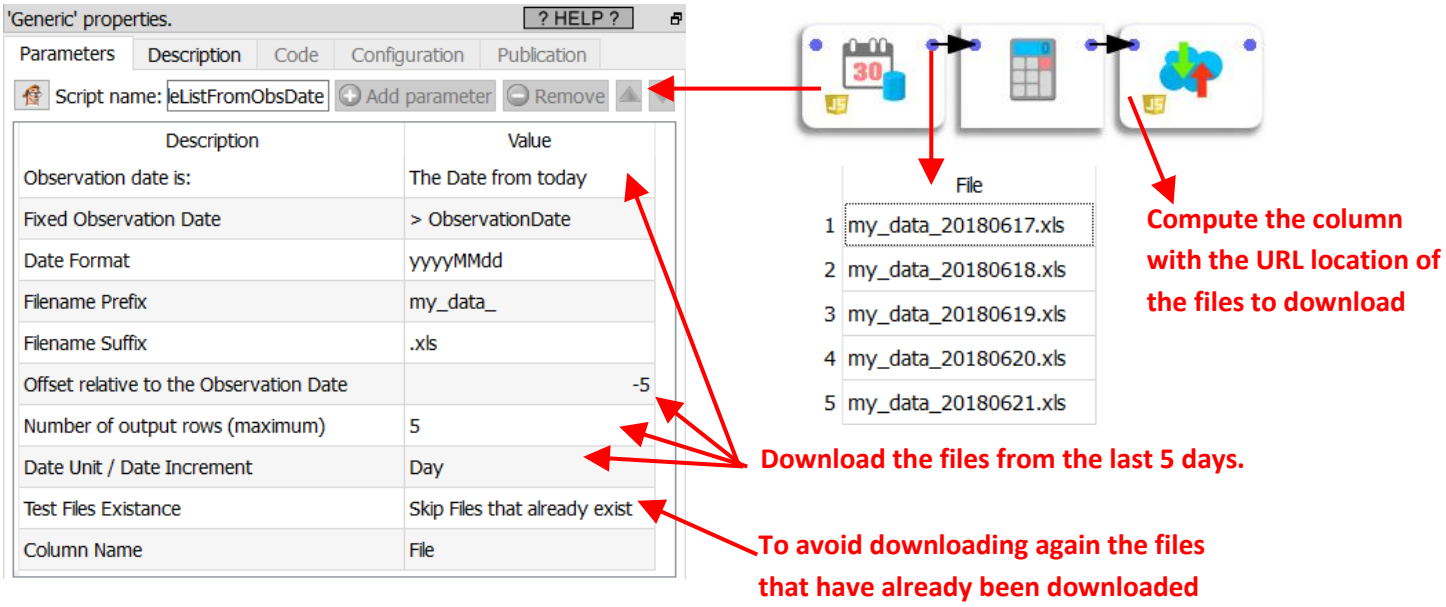
Property window:

Short description:
Upload or download a series of files

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

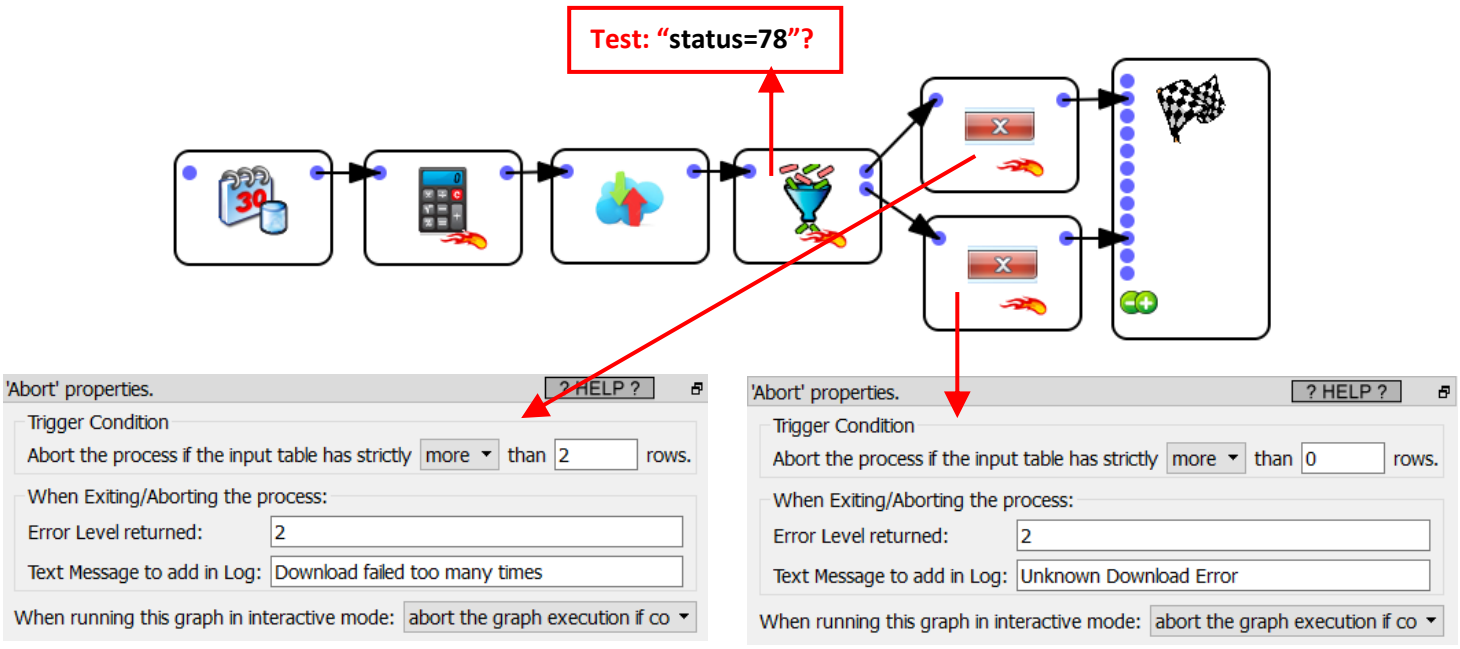
This Action is often used with the fileListFromObsDate Action (see section 5.23.5 for more this Action). For example, if we want to download the file names "my_data_<the date>.xls" for the last five days (skipping the files that we already downloaded), we'll have something like this:



As output, for each downloaded/uploaded file, you'll get an "Error Code" that give some explanation about the success/failure of the download/upload. The error code "zero" means: No error. The meaning of the other Error Codes (other than "zero") is given in the section 5.1.9.1.

One very common "error code" (or "Status") is the number "78": REMOTE FILE NOT FOUND. For example: The Anatella graph below stops when:

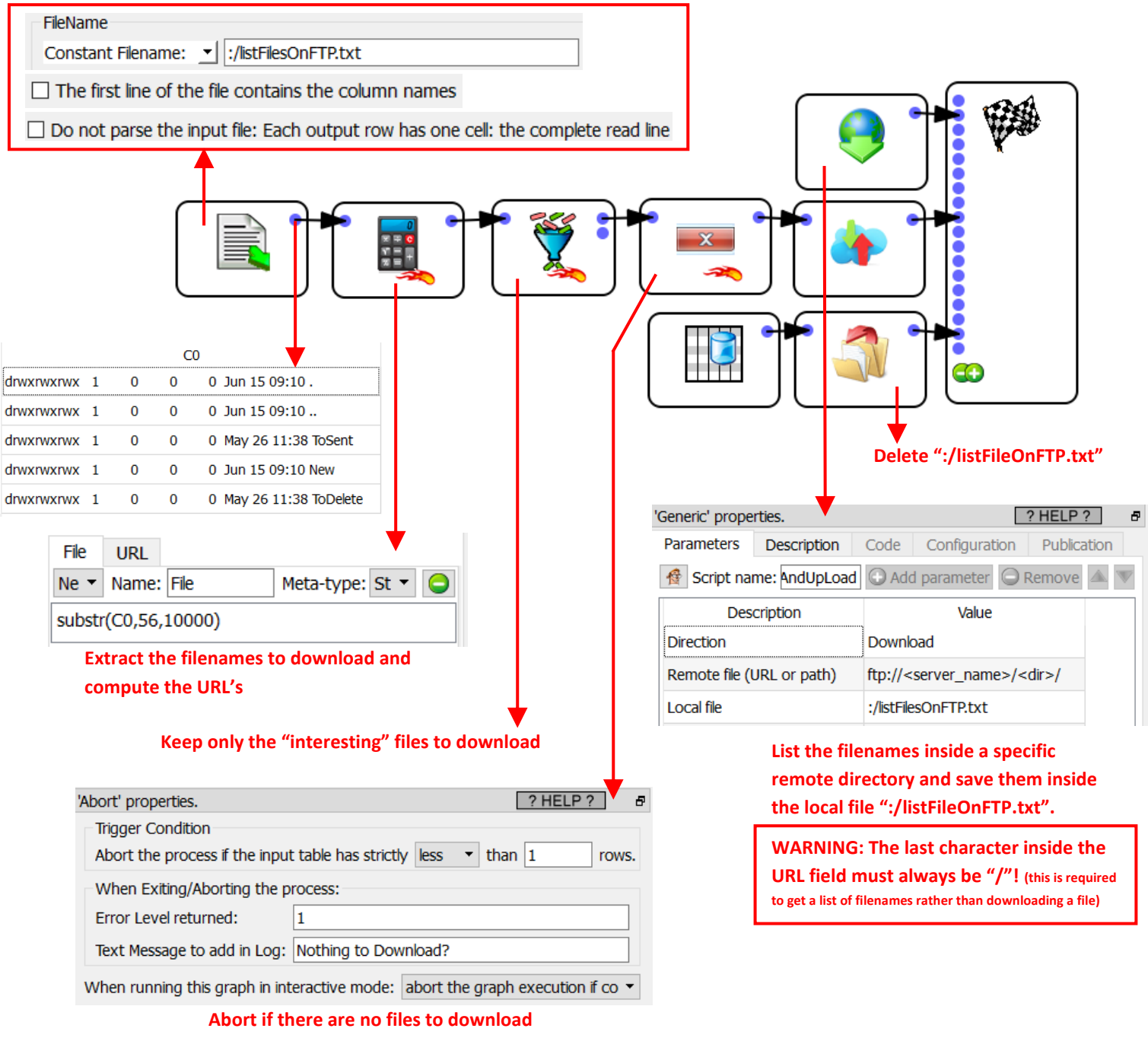
- …the number failure (to download some file) is strictly above 2.
- …directly in case of any other type of error.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.2.1. Downloading files with unknown filenames

When the names of the files to download are totally unknown, you can use the downAndUpLoad Action to, first, list the files in a remote directory and thereafter use the multipleDownAndUpLoad Action to thereafter download the required files. Here is an example:

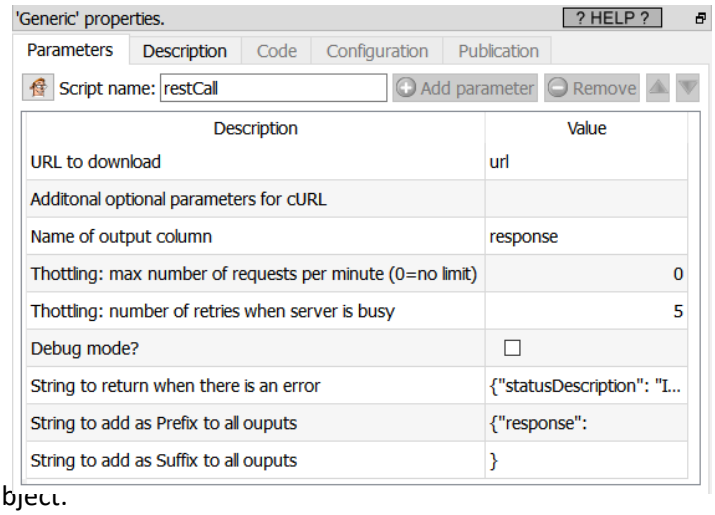


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.3. Calling a generic REST api

Icon:

Property window:



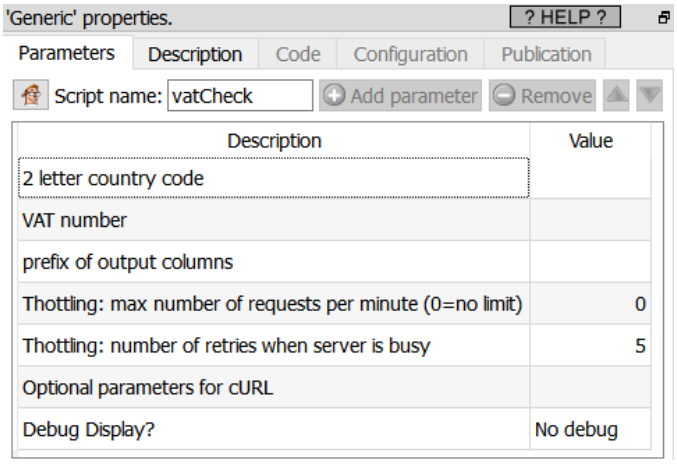
Short description:
Calls a REST API with the URL's given in input

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

5.23.4. Check VAT

Icon:

Property window:



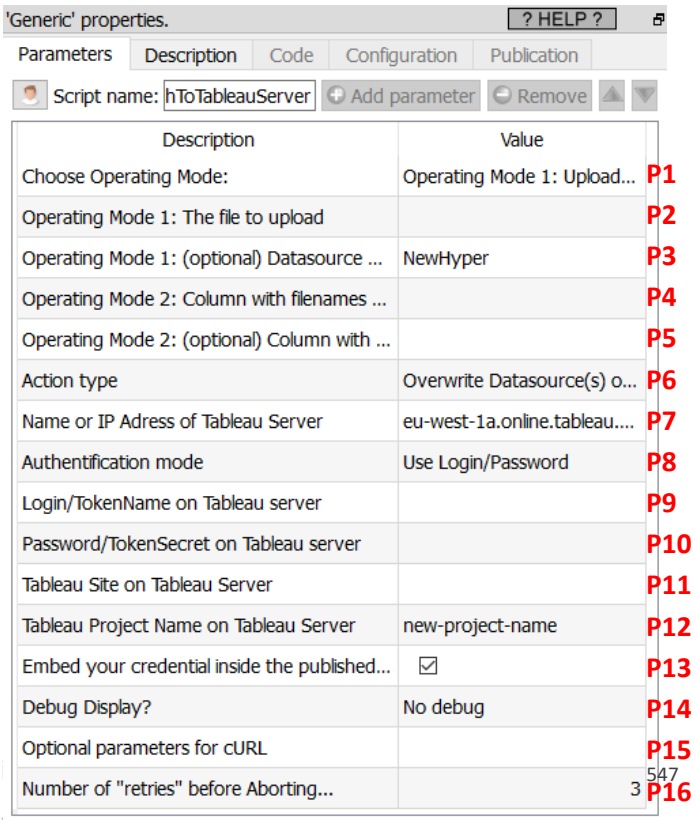
Short description:
Check the VAT numbers given in input

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

5.23.5. Publish to Tableau Server

Icon:

Property window:



Short description:
Upload file(s) to a remote Tableau Server

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when the tableau server is only accessible through an Internet proxy: Please consult the section 5.1.9.2. for more details on this subject.

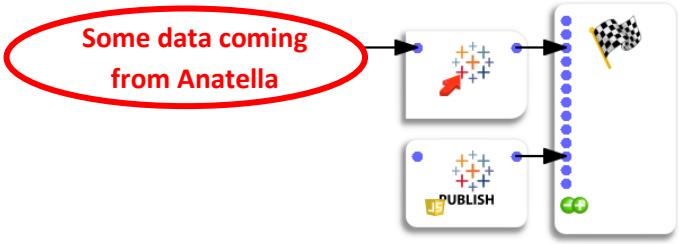
Upload some hyper, .tds, .tdsx, .tde, .twb, .twbx file(s) to a remote Tableau Server.

In a typical business intelligence project, you'll follow the following workflow:

- Using the Anatella writeHyper Action (see section 5.27.16.), you create at high-speed a .hyper file (or a .tde file) that is stored on your local hard drive.
- You copy your .hyper file (or .ttde file) from your local hard drive to a remote Tableau server so that everybody can enjoy your dashboards on "fresh & updated" data: This is done with the current Action: the

PublishToTableauServer Action.

So that, you typically get something like this:



To authenticate yourself on the Tableau server (using the Anatella parameter **P8**), you can select one of these two options:

- Use your Standard Login/password
- Use a Personal Access Token (PAT): **This is the only working option when you are using SSO** (Single-Sign-On to sign into your tableau Server) or when using multi-factor authentication (MFA).

The decision to "switch" your Tableau Server authentification scheme to SSO or MFA has profound implications on the way you'll work with Anatella (to refresh your data sources) because you'll then be forced to use Personal Access Token (PAT) instead of your Standard Login/password to log-in into the Tableau server. Indeed, these Personal Access Token (PAT) have very strong limitations that are given inside the sections 5.23.5.2. and 5.23.5.3. below.

If you get a sign-in error such as this one:

```
An error occurred while running initializion of the script
'publishToTableauServer'
Tableau Error Code : 401001
Tableau Error Summary : Signin Error
Tableau Error Detail : Error signing in to Tableau Server
```

..then you should check your Login/Password (or, if you are using PAT, your TokenName/TokenSecret) **and also your "Tableau Site"** (Parameter **P11**). Indeed, the "Tableau Site" (Parameter **P11**) is an essential part of the sign-in process: If it's wrong, you'll get a sign-in error.

Also: See hereafter for some important information about Parameter **P11**.

How to find the correct value for the parameter **P11** (i.e. the "Tableau site" parameter)? The parameter **P11** is the subpath of a site's full URL. In a server environment, it is referred to as the "Site ID". When you sign in to "Tableau Server" or to "Tableau Cloud" manually (inside your browser), the parameter

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

P11 is the the value that appears after /site/ in the Browser address bar. For example, in the following URLs, the parameter **P11** is *MarketingTeam*:

- **Tableau Server** - http://MyServer/#/site/*MarketingTeam*/projects

  To specify the default site, make the parameter **P11** an empty string (set parameter **P11**="")
  Warning: To specify the default site, do not set the parameter **P11** to "default", but instead to "".

- **Tableau Cloud** - https://10ay.online.tableau.com/#/site/*MarketingTeam*/workbooks

When the parameter **P8** is set to "Use PAT", you cannot set the parameter **P13** to "checked". This means that you cannot update the *.tds *.tdsx *.twb *.twbx files to your Tableau Server (i.e. you can only update .hyper or .tde files when **P8** is set to "Use PAT").

5.23.5.1. How to get a Personal Access Token (PAT) ?

To use a Personal Access Token (PAT) inside Anatella, you need to set parameter **P8** to "Use Personal Access Token (PAT)".

Follow this procedure to get your PAT:

1. Go to your Account Settings page inside your Tableau Server. At the top of a page, click your profile image or initials, and then select My Account Settings:

2. Under Personal Access Tokens, enter a descriptive name for your token in the "Token Name" field (this is the Anatella Parameter **P9**), and then click "Create" to create the new token.

3. In the resulting window you get your "Token Secret" (this is the Anatella Parameter **P10**), click to copy it to the clipboard and then close the window.

5.23.5.2. Limitations linked to Personal Access Token (PAT)

No concurrent usage.
One personal access token is required *for each different concurrent request*. Signing-in again with the same access token terminates the previous session and result in an authentication error!

Thus, a good practice is to create a different PAT for each different data source that you need to update with Anatella. In this way, you'll be able to run many different updates simultaneously without having any authentication error.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Limited number of Personal Access Token (PAT) -1- User Level.

By default, each user can create a maximum of 24 PAT. Thus, this means that each user can have a maximum of 24 data sources that can be updated with Anatella (for more details: see the previous point about "No Concurrent usage").

You can change the maximum number of Personal Access Tokens (PAT) that a user can create using the "refresh_token.max_count_per_user" option with the "`tsm configuration set`" command (set to "-1" to remove any limitations).

Limited number of Personal Access Token (PAT) -2- Device/Site Level.

By default, each device/site can use a maximum of 10 PAT. Thus, this means that each device/site can have a maximum of 10 data sources that can be updated with Anatella.

You can change the maximum number of Personal Access Tokens (PAT) that a device/site can use with the "refresh_token.max_count_per_device" option with the "`tsm configuration set`" command.

Fast expirations.

By default, Personal Access Tokens (PAT) expire after 15 consecutive days (if they are not used) or after 1 year (if they are used continuously). After a year, you must create a new token. Expired personal access tokens will not be displayed on the My Account Settings page anymore.

You can change refresh token expiry time span using the "*refresh_token.absolute_expiry_in_seconds*" option with the "`tsm configuration set`" command (set to "-1" to remove completely the expiration).

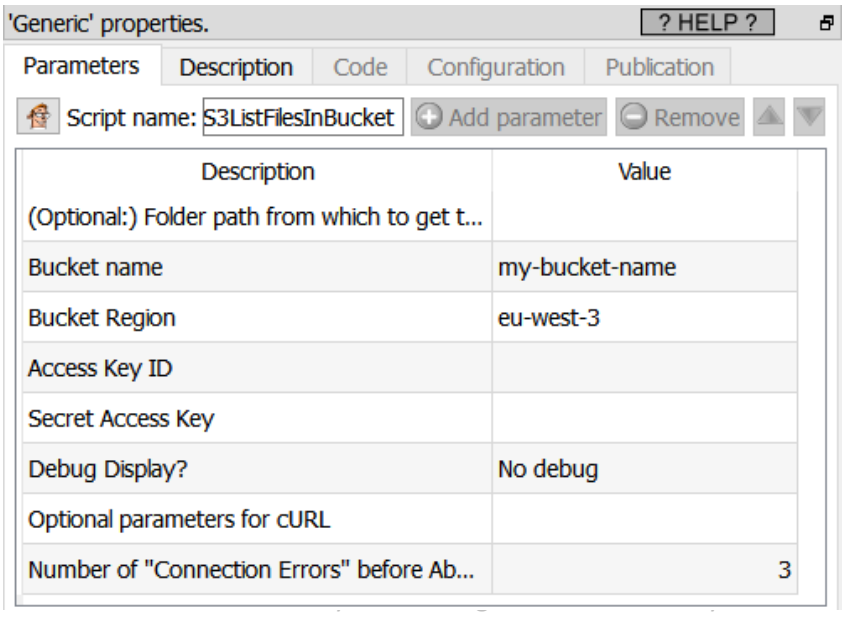
5.23.5.3. Benefits linked to the use of Personal Access Token (PAT)

This is the only working option when you are using SSO (Single-Sign-On to sign into your tableau Server) or when using multi-factor authentication (MFA). When using tokens, then password resets on user accounts will not disrupt automation as it would when credentials are hard-coded into the scripts.

5.23.6. List the Files inside a S3 bucket

Icon:

Property window:



Short description:
List all files inside a S3 bucket

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

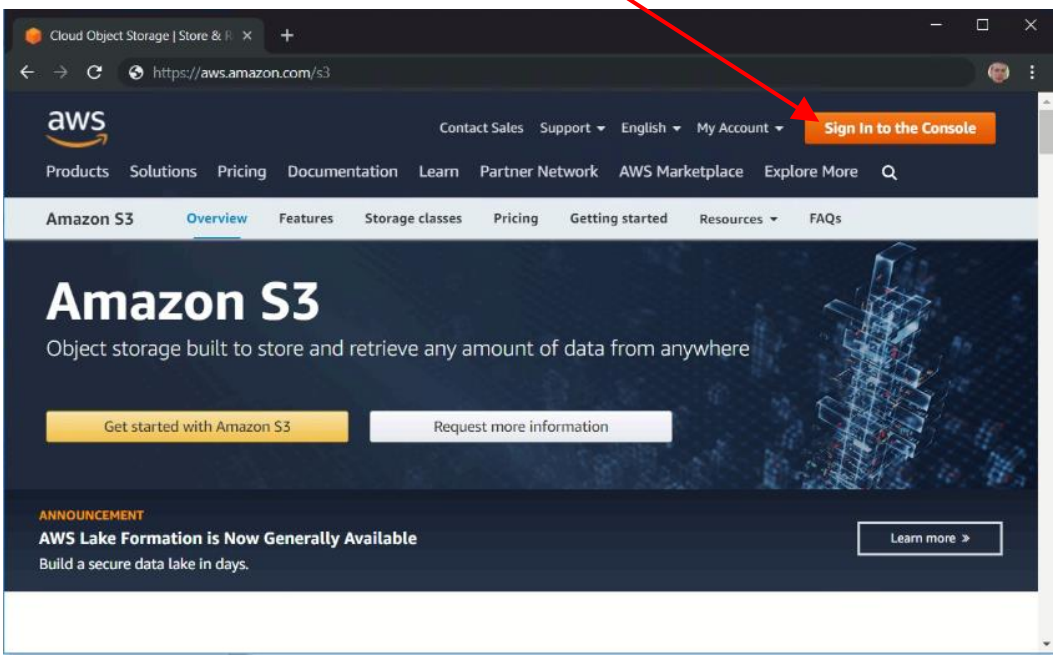
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to connect to your S3 bucket storage, you need to get from Amazon these 4 parameters:

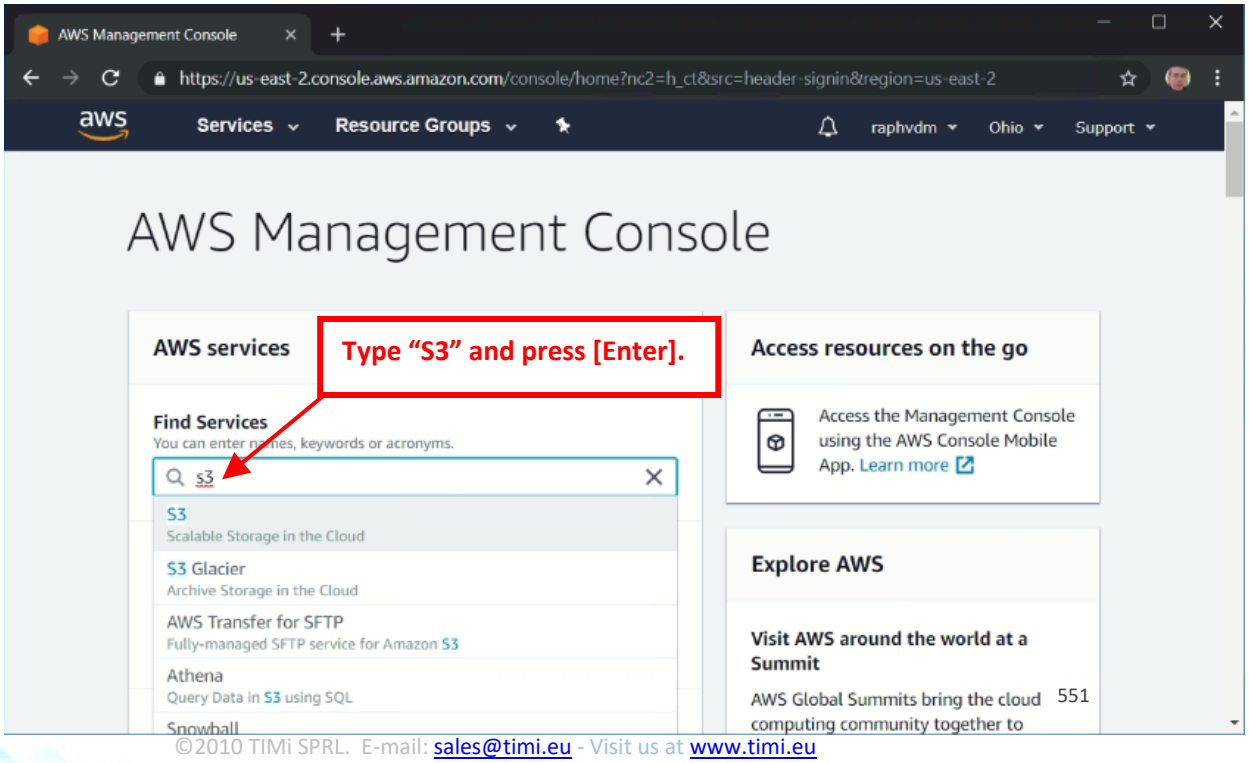
- your “Bucket Name”
- your “Region”
- your “Access Key ID”
- your “Secret Access Key”

The procedure to get these 4 parameters is:

1. Open the URL address http://aws.amazon.com/s3 in a browser and click on the “Sign In to the Console” button on the top-right corner of the window:

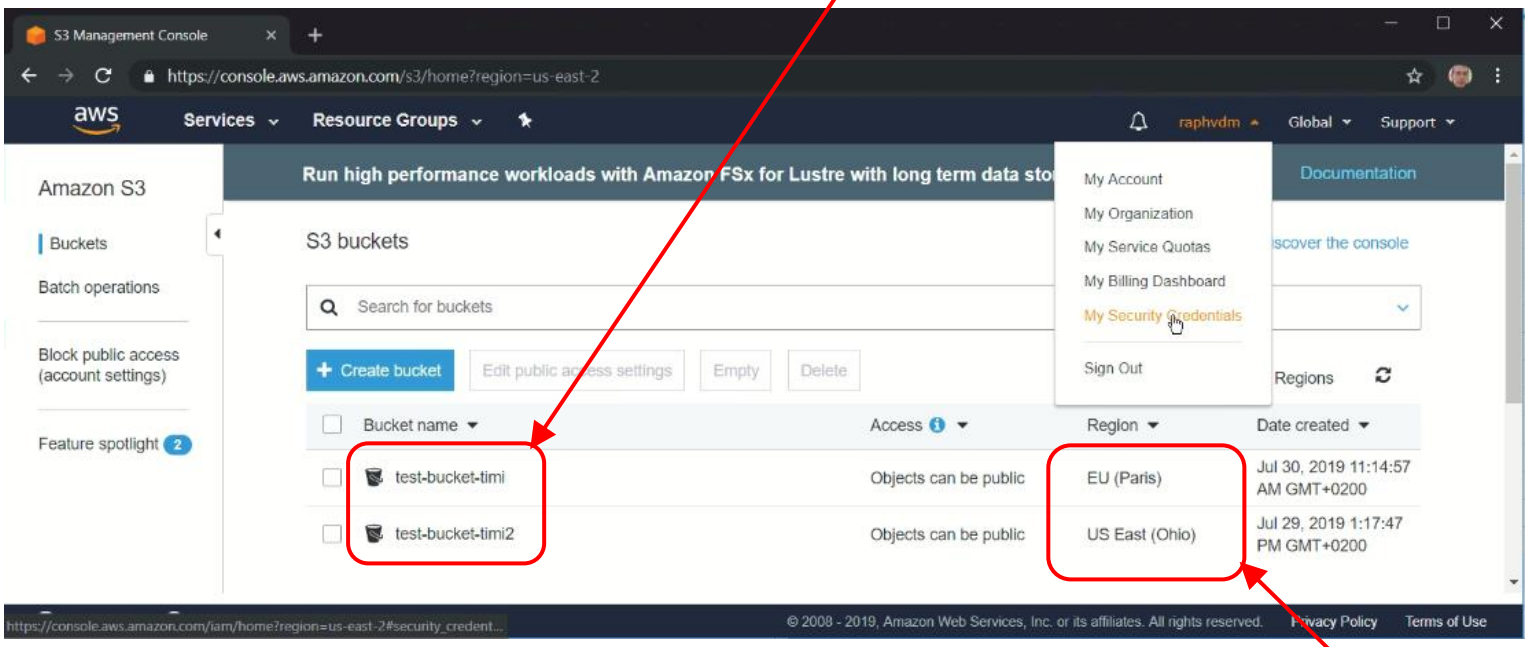


2. Inside the search field, type “s3” to locate the “S3 service” and press Enter to access the “S3 management console”:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Inside the "S3 management console", you get the first 2 required parameters for Anatella: the "Bucket Name" and the "Region". In the screenshot below, there are two S3 buckets named "test-bucket-timi" and "test-bucket-timi2":



The "Region Name" of these two S3 buckets is, respectively, "EU (Paris)" and "US East (Ohio)": Inside Anatella, you don't need the "Region **Name**": i.e. you need to have the "Region" parameter. To get the "Region" parameter, from the "Region Name", you can use the lookup table located at this URL:

https://docs.aws.amazon.com/general/latest/gr/rande.html

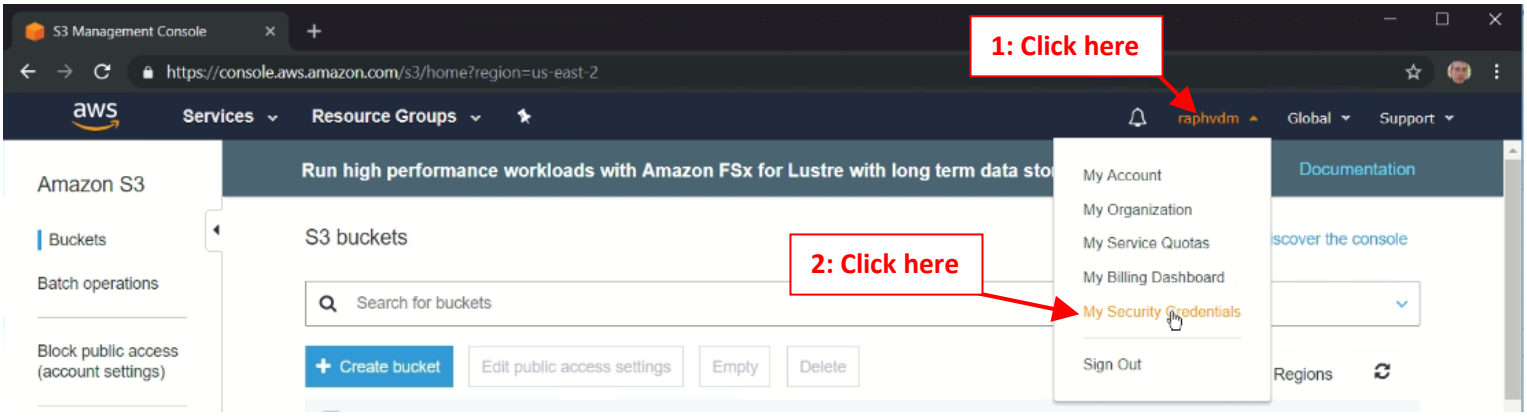
For your convenience, here is a local copy of this lookup table:

Region Name	Region
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
EU (Frankfurt)	eu-central-1
EU (Ireland)	eu-west-1
EU (London)	eu-west-2
EU (Paris)	eu-west-3

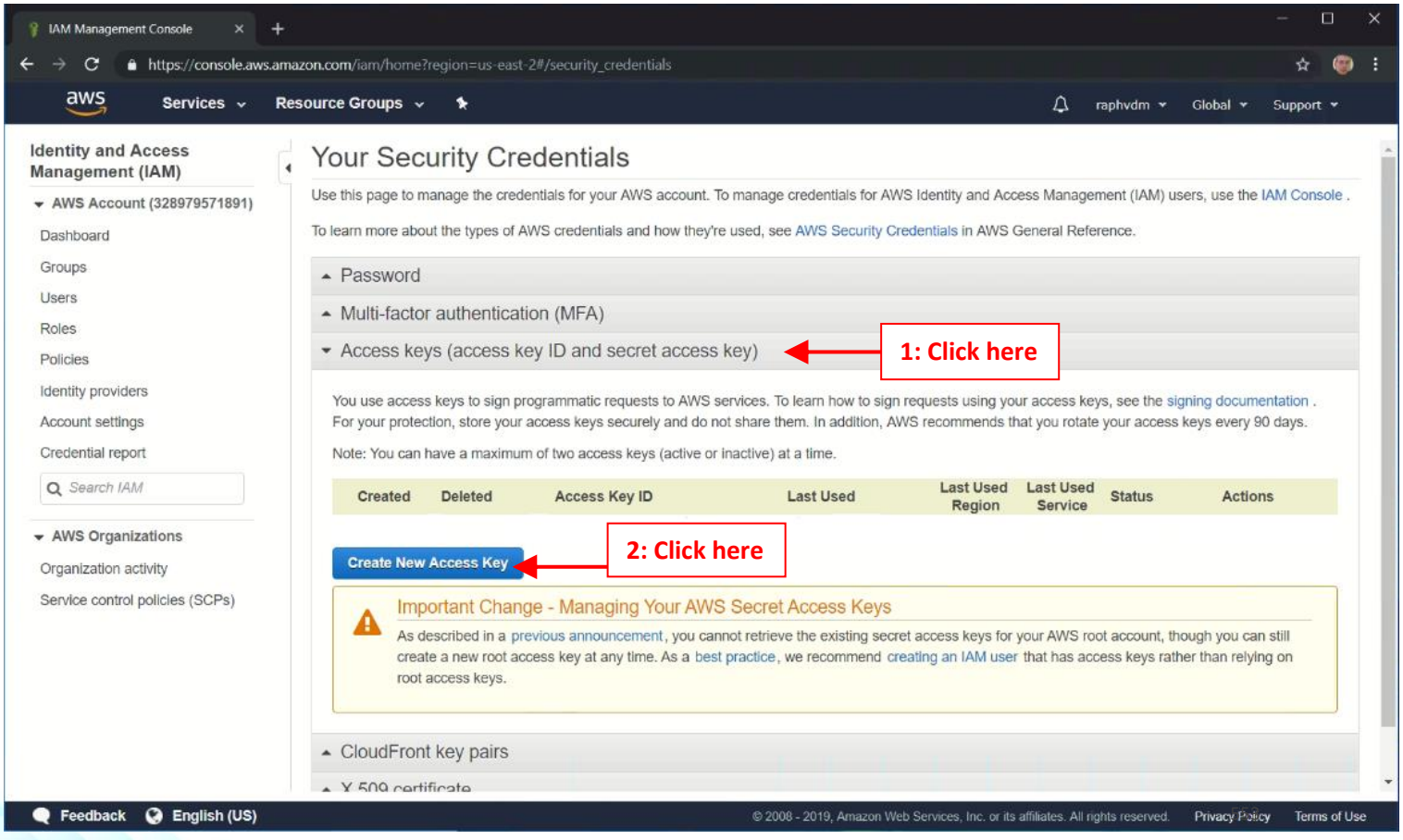
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

EU (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (Sao Paulo)	sa-east-1
AWS GovCloud (US-East)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1
Region Name	Region

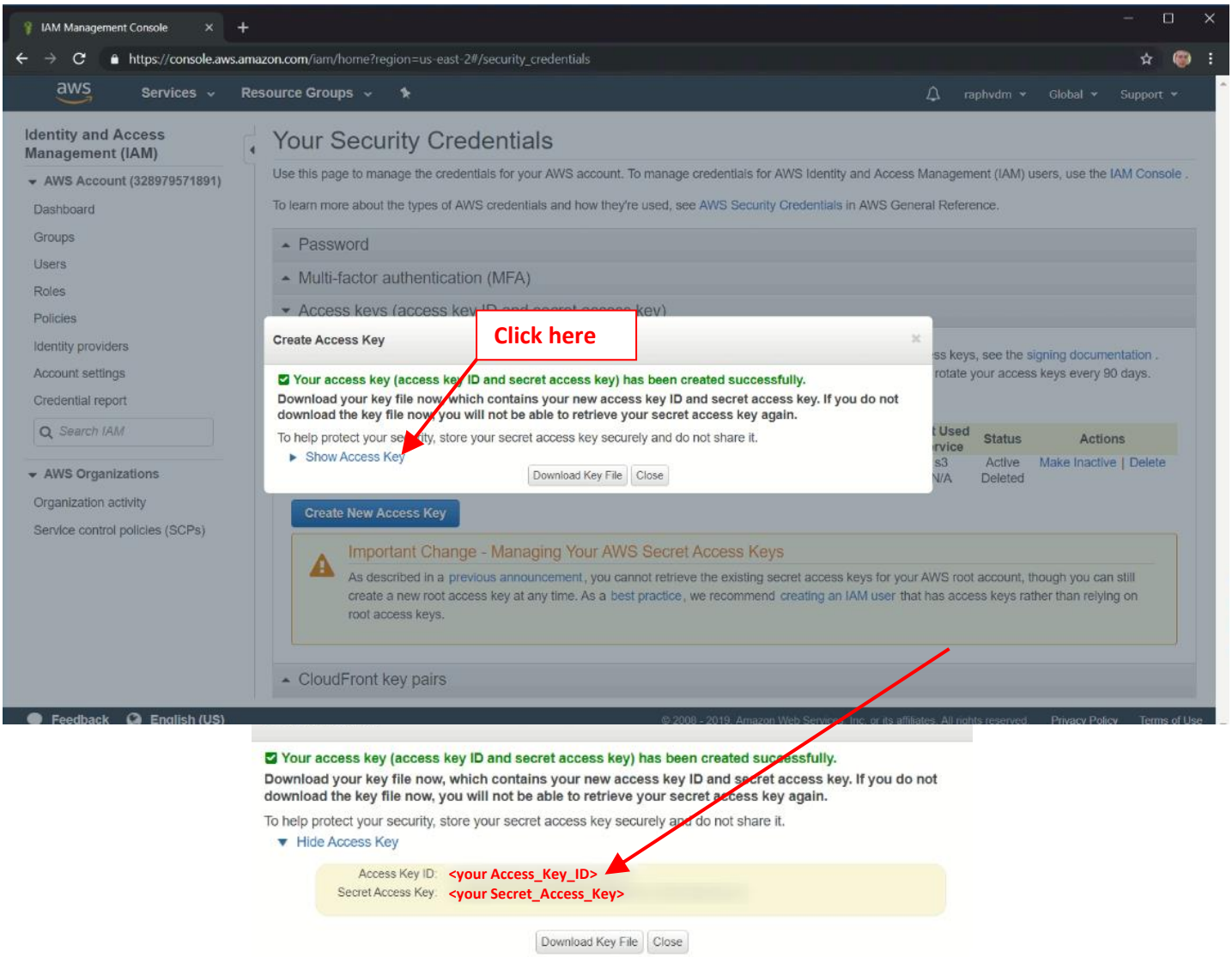
To get the last 2 parameters (i.e. your "Access Key ID" and your "Secret Access Key"), click on the drop-down menu in the top-right corner to see your account options ans select the "My Security Credentials" option:



4. Inside the "Identity and Access Management (IAM) console", click on the button named "*Access keys (access key ID and secret access key)*" and then click on the blue button named "*Create New access Key*":



5. Inside the new pop-up window that just opened in the browser, click on the "*Show Access Key*" link:



Just copy&paste your "Access Key ID" and your "Secret Access Key" inside the corresponding Anatella parameters box, and you are good to go!

5.23.7. List S3 buckets

Icon:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: S3ListBucket | Add parameter | Remove

Description	Value
Access Key ID	
Secret Access Key	
Debug Display?	No debug
Optional parameters for cURL	

Property window:

Short description:
List all available S3 buckets

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To be able to use this Action, you need to get from Amazon these 2 parameters:

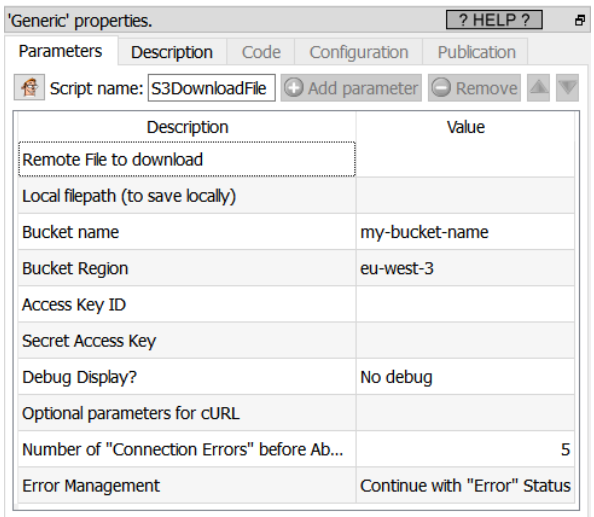
- your “Access Key ID”
- your “Secret Access Key”

Please refer to the previous section (Section 5.23.6.) for the exact procedure on how to get these 2 parameters.

5.23.8. Download Files from a S3 bucket

Icon:

Property window:



Short description:
Download files from a S3 bucket

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to connect to your S3 bucket storage, you need to get from Amazon these 4 parameters:

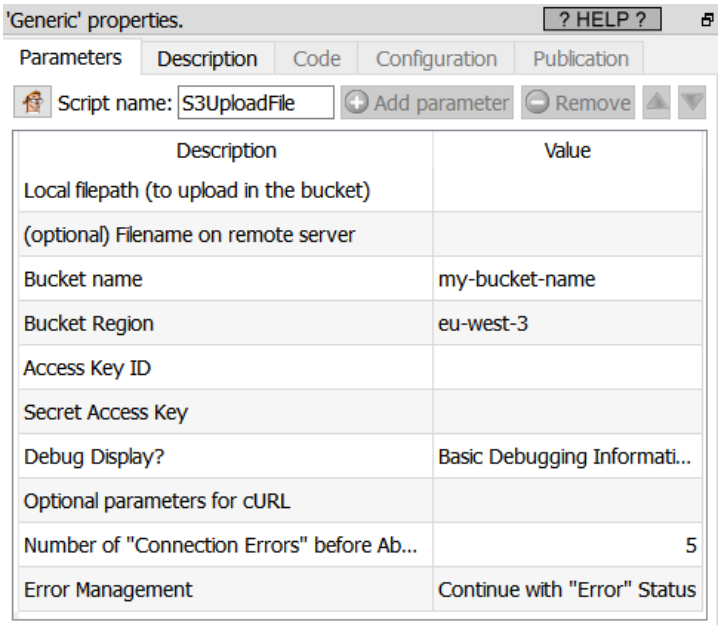
- your “Bucket Name”
- your “Region”
- your “Access Key ID”
- your “Secret Access Key”

Please refer to the Section 5.23.6. for the exact procedure on how to get these 4 parameters.

5.23.9. Upload Files to a S3 bucket

Icon:

Property window:



Short description:
Upload files to a S3 bucket

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to connect to your S3 bucket storage, you need to get from Amazon these 4 parameters:

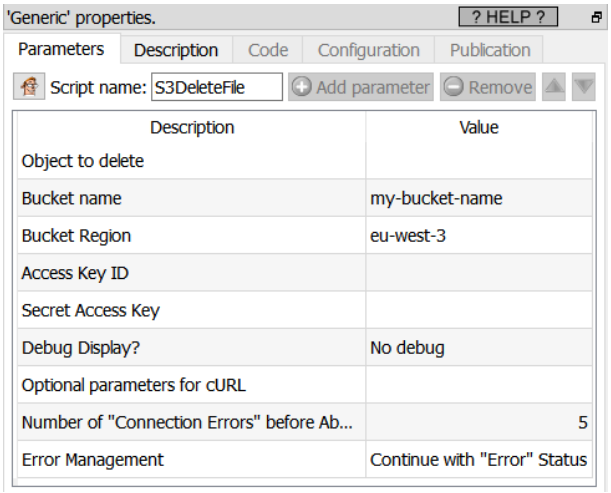
- your “Bucket Name”
- your “Region”
- your “Access Key ID”
- your “Secret Access Key”

Please refer to the Section 5.23.6. for the exact procedure on how to get these 4 parameters.

5.23.10. Delete Files stored in a S3 bucket

Icon:

Property window:



Short description:
Delete files on a S3 bucket
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to connect to your S3 bucket storage, you need to get from Amazon these 4 parameters:

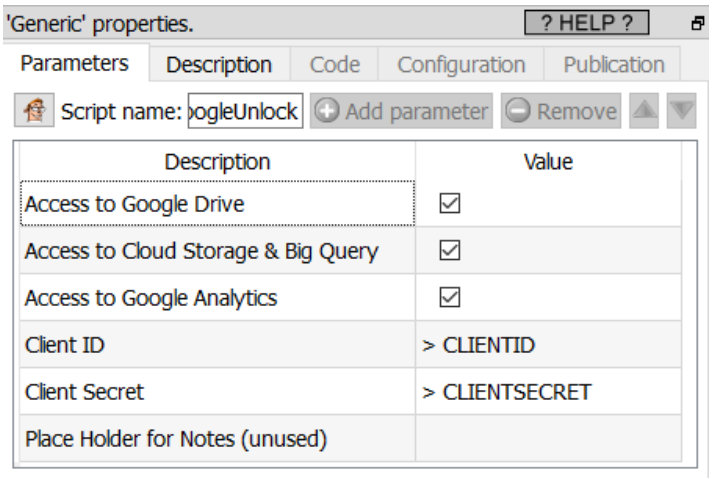
- your “Bucket Name”
- your “Region”
- your “Access Key ID”
- your “Secret Access Key”

Please refer to the Section 5.23.6. for the exact procedure on how to get these 4 parameters.

5.23.11. Unlock Google Services

Icon:

Property window:



Short description:
Unlock access to Google services

Long Description:
You need to get some Google credentials (i.e. you need to get from Google these 3 parameters: your “Client ID”, your “Client Secret” and your “Refresh Token”) before using any of the following Actions:

- GDrive:
  - List the files on a GDrive (section 5.23.12)

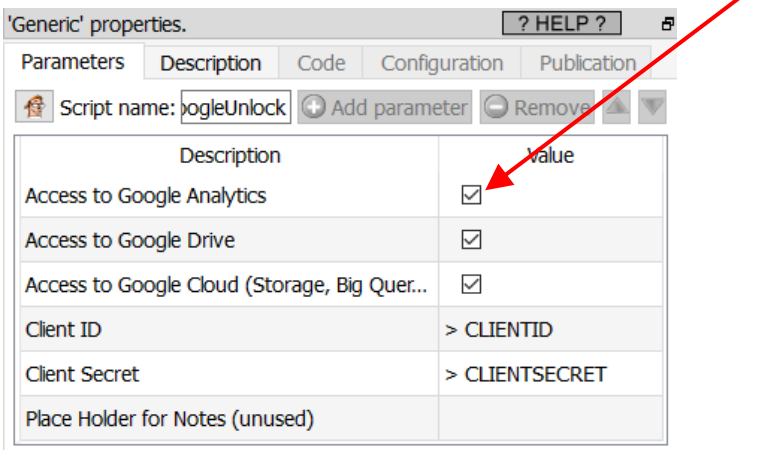
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

  - Download files from a GDrive (section 5.23.13)
  - Upload files to a GDrive (section 5.23.14)
  - Delete files stored in a GDrive (section 5.23.15)
- Cloud Storage
  - List the files on a Google Cloud Storage (section 5.23.16)
  - Download files from a Google Cloud Storage (section 5.23.17)
  - Upload files to a Google Cloud Storage (section 5.23.18)
  - List all your Google Cloud Storage Buckets (section 5.23.19)
  - Delete files from a Google Cloud Storage (section 5.23.20)
- Big Query
  - Execute a SQL query on Big Query (section 5.23.21)
  - Upload CSV files to Big Query (section 5.23.22)
- Google Analytics (section 5.23.23)
- Google Speech-to-Text (section 5.23.48)

To get your Google credentials you don't need to pay anything to Google. You can get all your credentials for free: i.e. You get your "Client ID" (obtained at step 9.7), your "Client Secret" (obtained at step 9.7) and your "Refresh Token" (obtained at step 9.11) for free.

The procedure described in this section allows you to get your Google credentials (i.e. to get a "Client ID", a "Client Secret" and a "Refresh Token") that will unlock ***ALL*** Google services: GDrive, Google Cloud Storage, Google Big Query and Google Speech-to-Text. For security reasons, you might want to get some credentials that are more limited (e.g. that only give you access to "GDrive", and not "Cloud Storage"). If you want some credentials that are:

- …limited to "Google Analytics" only:
  - Skip the point 4, 5 and 6 in the procedure below.
  - On point 8.4, select only the following scope:
    - `../auth/analytics.readonly`
  - On point 9.7, enable (i.e. check) the option "*Access to Google Analytics*":

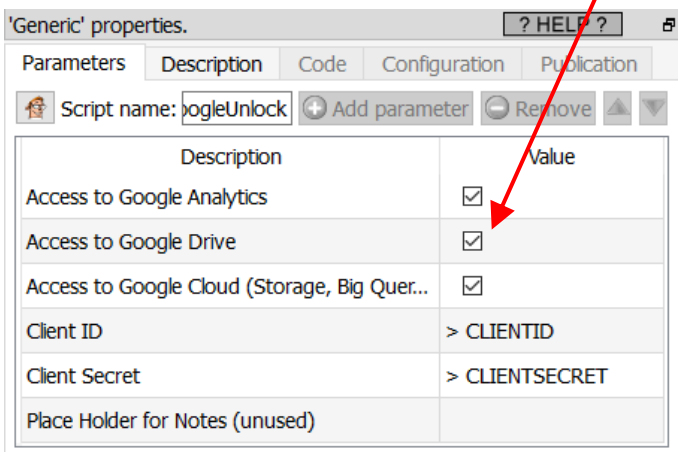


- …limited to "GDrive" only:
  - Skip the points 5,6 and 7 in the procedure below.
  - On point 8.4, select only the following scopes:
    - `../auth/drive.metadata`
      (change file-related metadata in your Google Drive)
    - `../auth/drive`

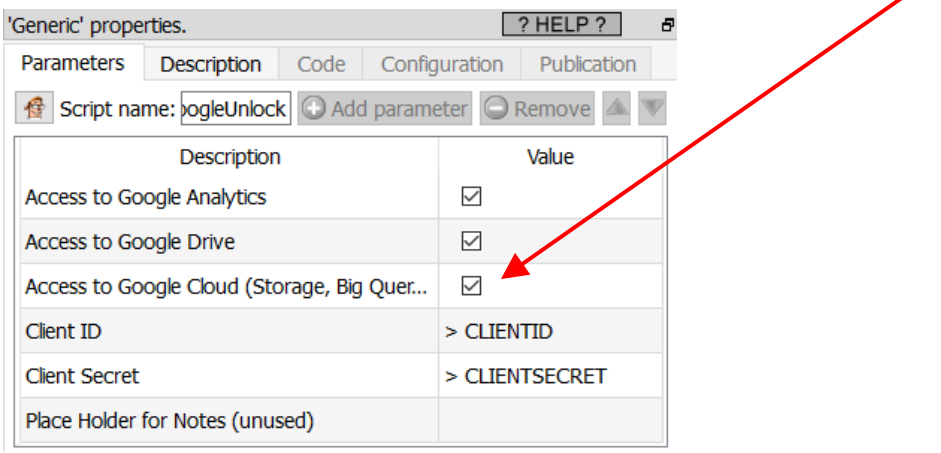
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

(See, Download and Upload files from/to your Google Drive)

- On point 9.7, enable (i.e. check) the option "*Access to GoogleDrive*":



- <u>…limited to "Cloud Storage, Big Query and Speech-to-Text" only:</u>
  - Skip the point 4 and 7 in the procedure below.
  - On point 8.4, select only the following scope:
    - `../auth/cloud-platform`
  - On point 9.7, enable (i.e. check) the option "*Access to Cloud Storage and Big Query*":



The procedure to get you Google Credentials (i.e. to get a "Client ID", a "Client Secret" and a "Refresh Token") is the following:

1. First, a word of caution:

   You will need your "Client ID", "Client Secret" and "Refresh Token" to use (nearly) all the Google Services actions inside Anatella.

   There are no ways to copy/paste back these 3 informations from one Anatella action to another (this is, of course, "by design").

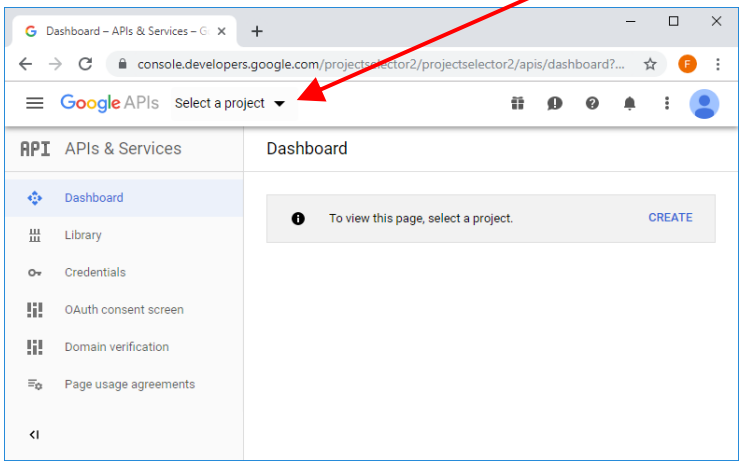
   So, you should keep yourself these 3 informations in a secure place, if you intend to re-use them later.

2. Open in your web browser the "Google Developper Console": Go to the URL:
   https://console.developers.google.com
   ..and sign-in into your Google account.

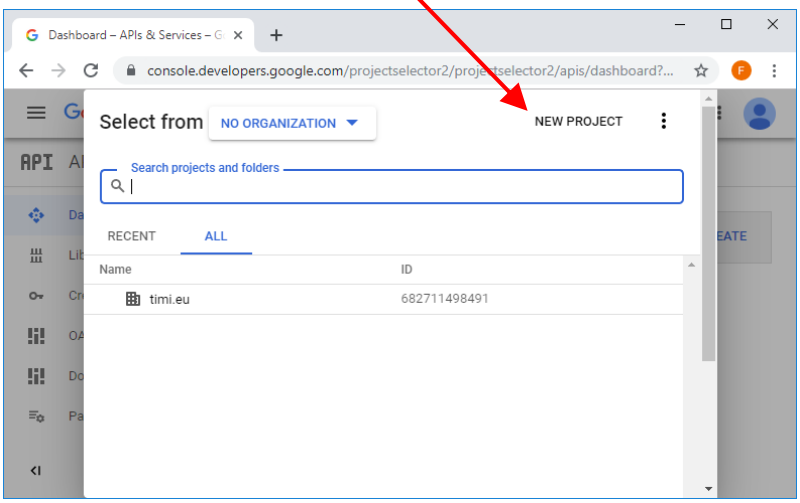
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Create a new Project.

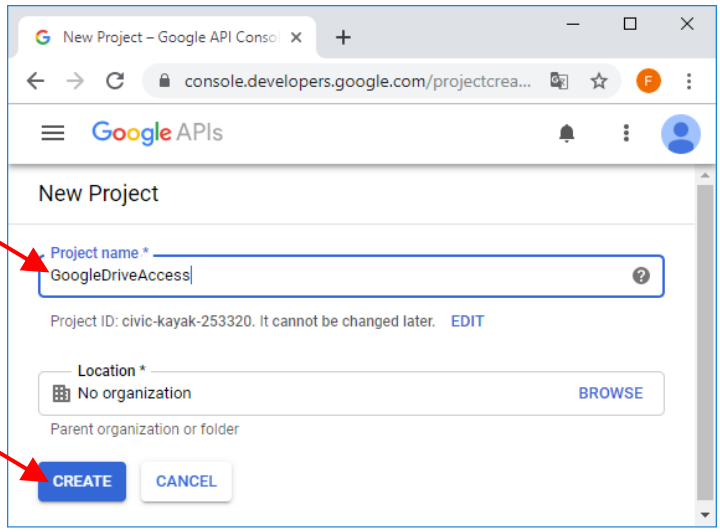
3.1. Click on the “Select a Project” drop-down menu:



3.2. Click on the “New Project” button:



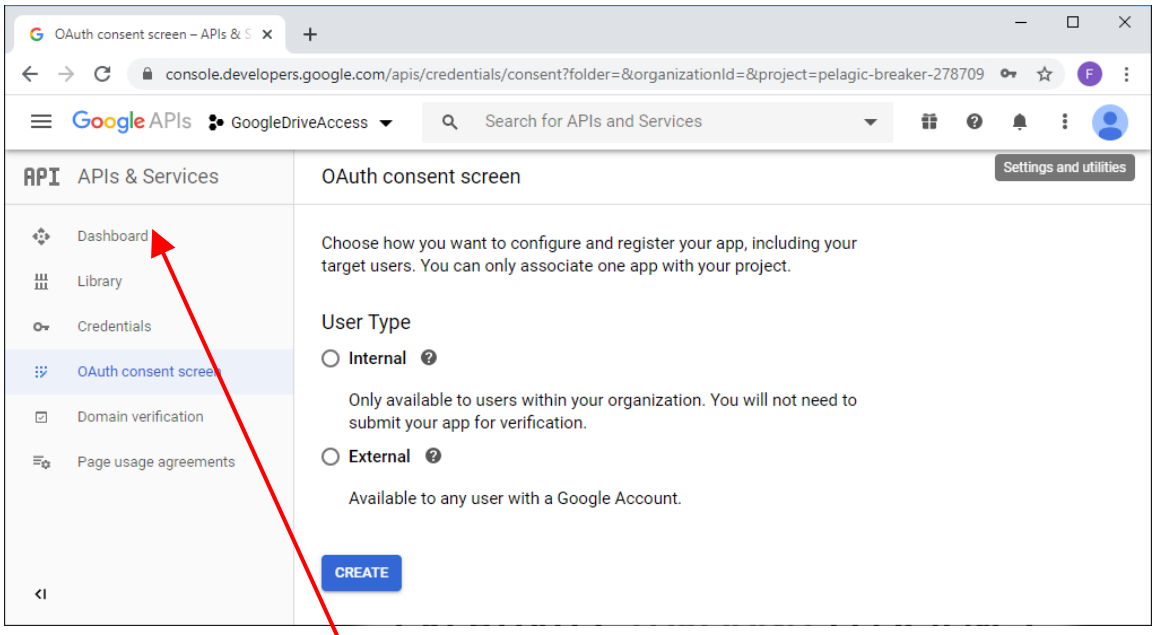
3.3. Write a project name (you can use whatever name you want) and click on the “Create” button:



Use any name here

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3.4. You should now see this page inviting you to configure your OAuth consent screen:



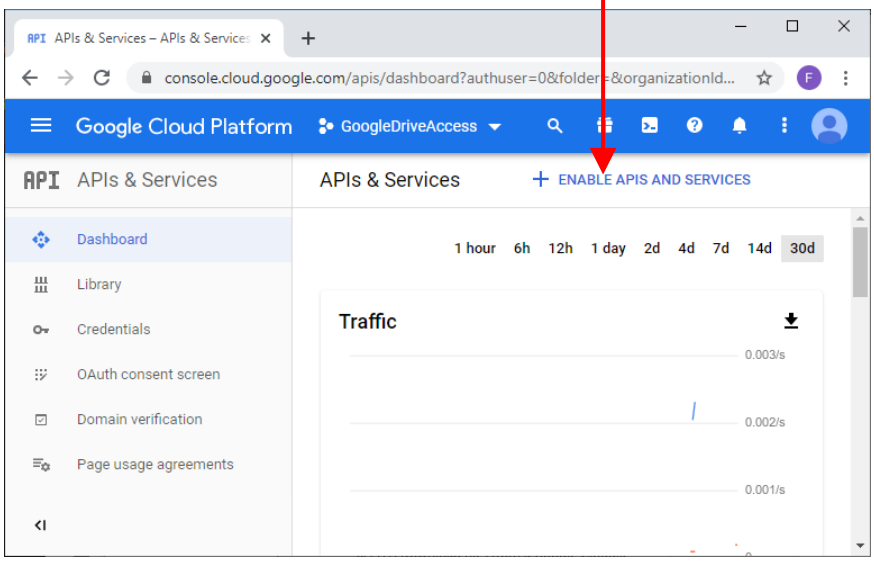
We will not configure the OAuth consent screen now because we first need to enable different google API and services beforehand. To do so, click on the "*Dashboard*" button:

4. We will now enable the "**Google Drive API**"

This API is required to be able to use the "Google Drive" Actions: These actions are:

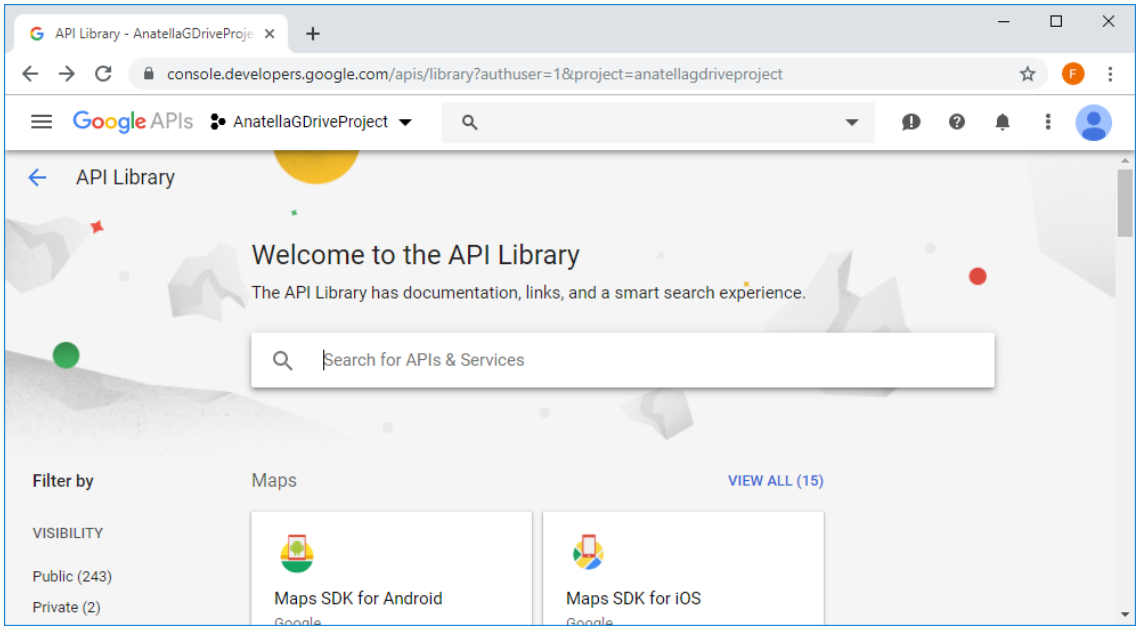
- List the files on a GDrive (section 5.23.12)
- Download files from a GDrive (section 5.23.13)
- Upload files to a GDrive (section 5.23.14)
- Delete files stored in a GDrive (section 5.23.15)

4.1. Click on the "Enable APIS and Service" button:

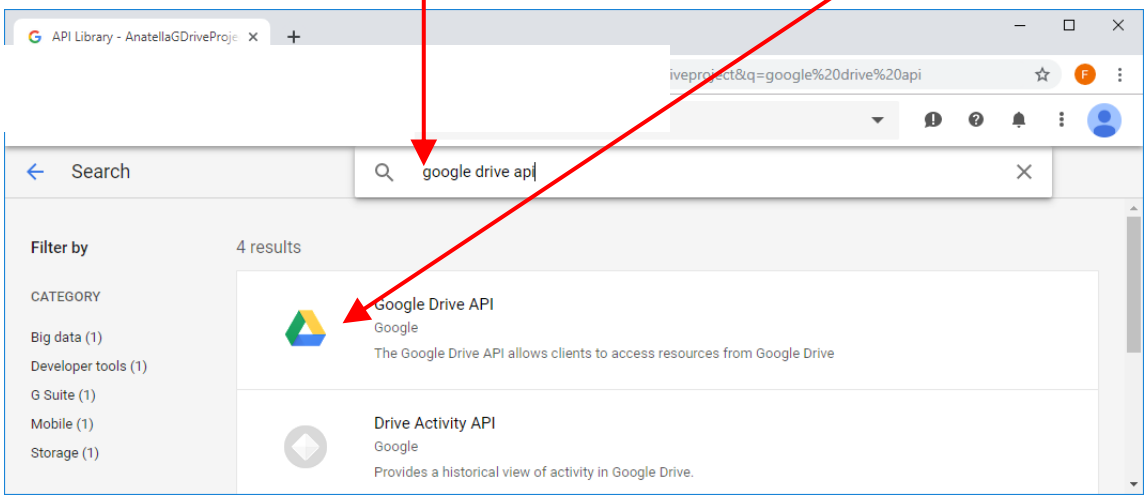


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

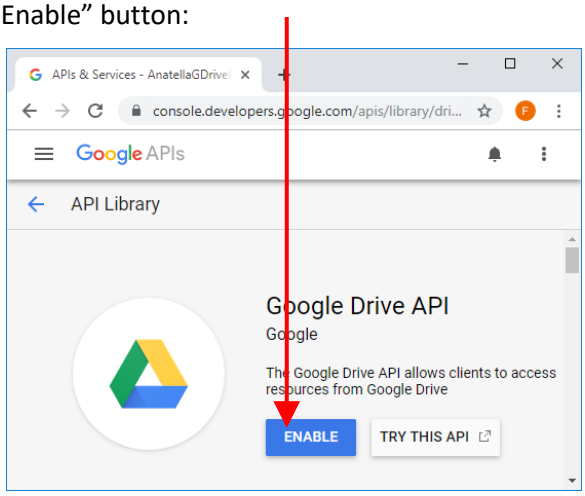
You arrive on this page:



4.2. Search for the “Google Drive API” and click on the first item found:

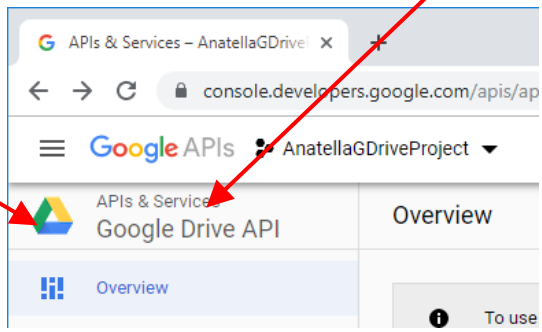


4.3. Click the “Enable” button:



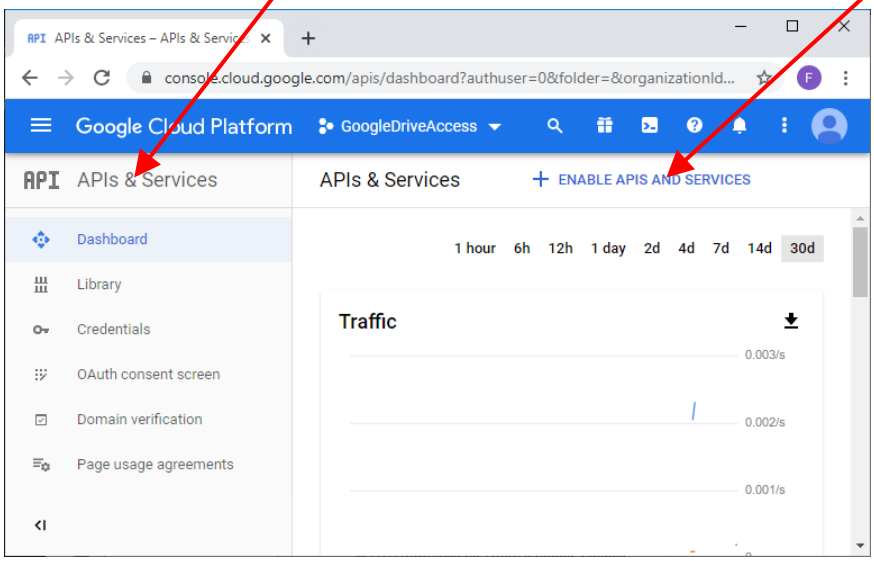
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4.4. We are back on the Project Selection screen but the Google Drive logo is now visible here: (and we don't want that). Click on the "APIs and Services" text here:



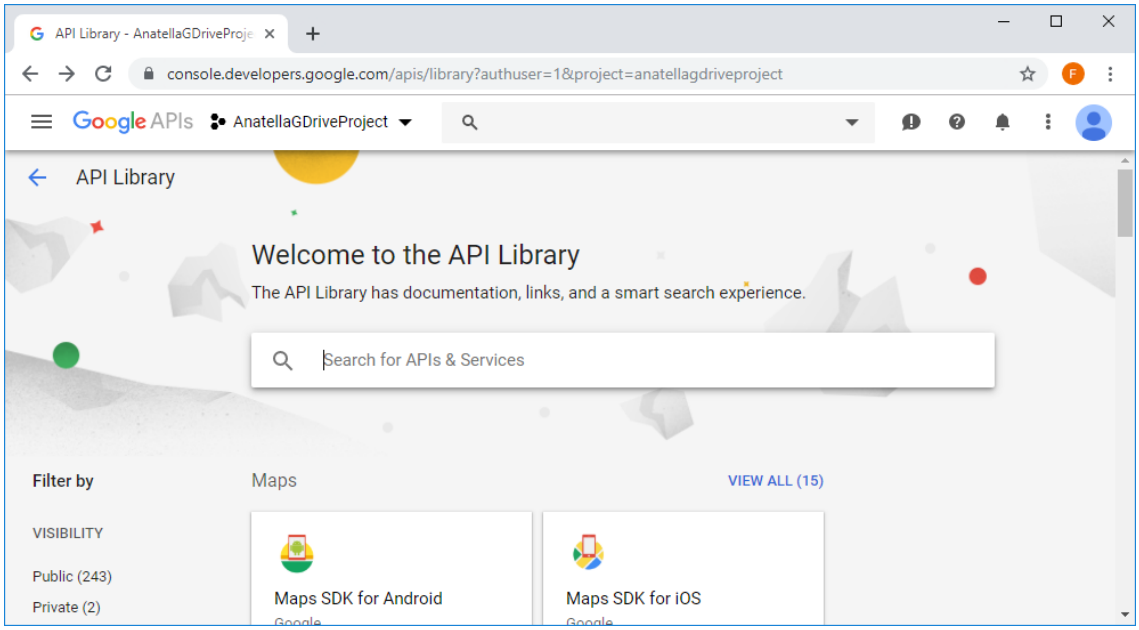
5. We will now Enable the "**Google Cloud Storage JSON API**".
This API is required to be able to use the "Google Cloud Storage" Actions: These actions are:
   - List the files on a Google Cloud Storage (section 5.23.16)
   - Download files from a Google Cloud Storage (section 5.23.17)
   - Upload files to a Google Cloud Storage (section 5.23.18)
   - List all your Google Cloud Storage Buckets (section 5.23.19)
   - Delete files from a Google Cloud Storage (section 5.23.20)

5.1. We are back on the Project Selection screen (and the GDrive logo is not visible anymore: it's written "APIs & Services" instead). Click on the "*Enable APIS and Service*" button:

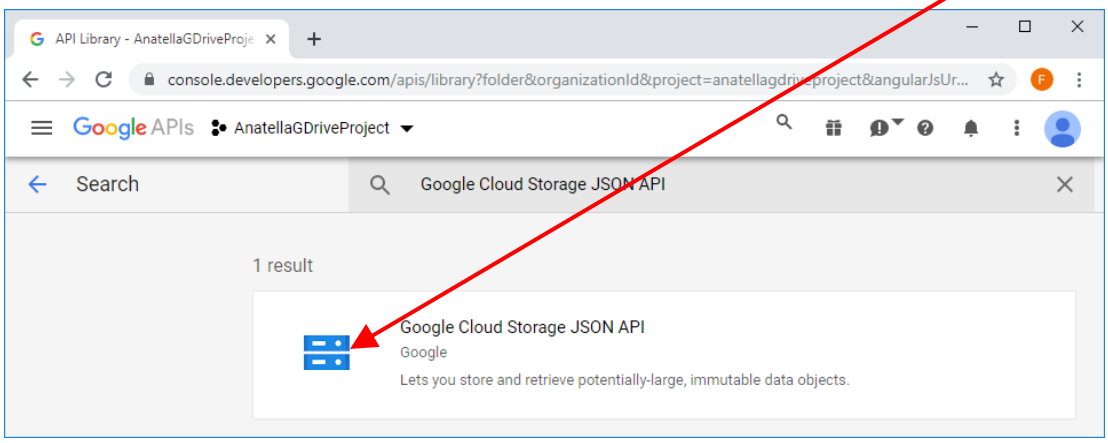


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

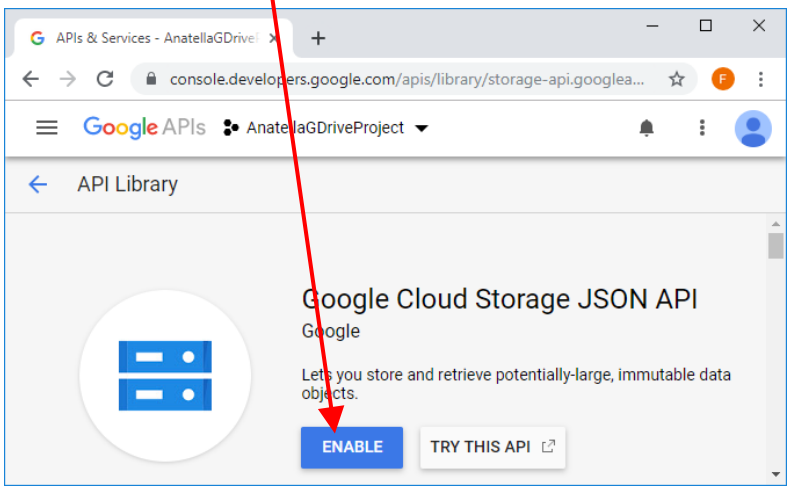
You arrive on this page:



5.2. Search for the “Google Cloud Storage JSON API” and click on the first item found:

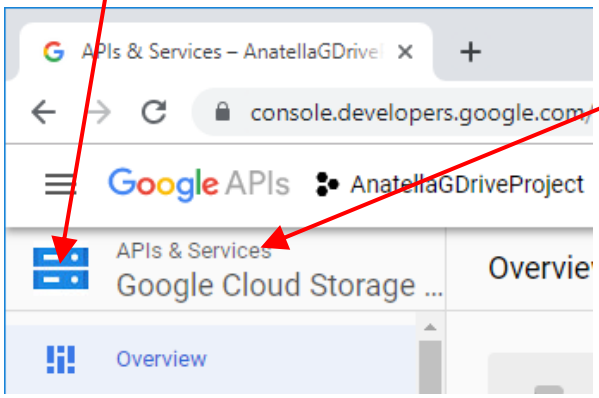


5.3. Click the “Enable” button:



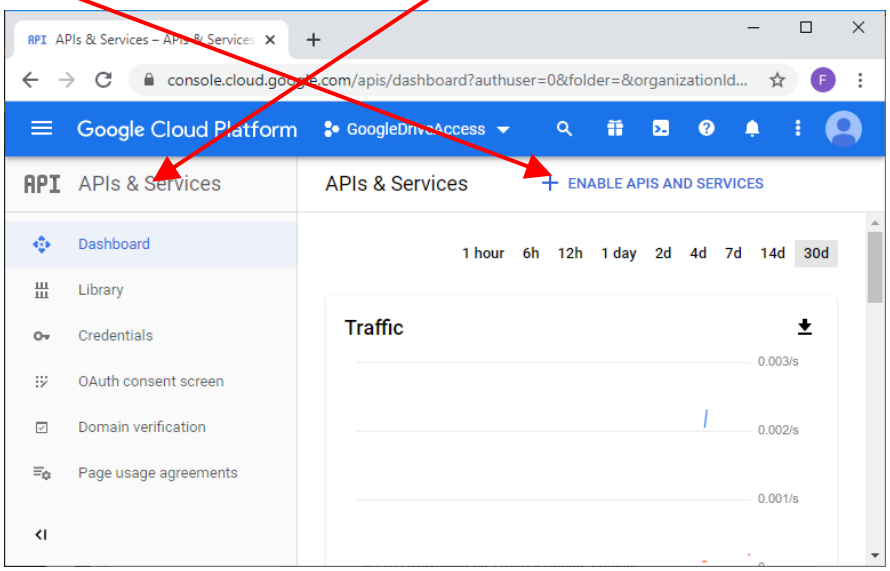
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.4. We are back on the Project Selection screen but the "Google Cloud Storage" logo is now visible here: (and we don't want that). Click on the "APIs and Services" text here:



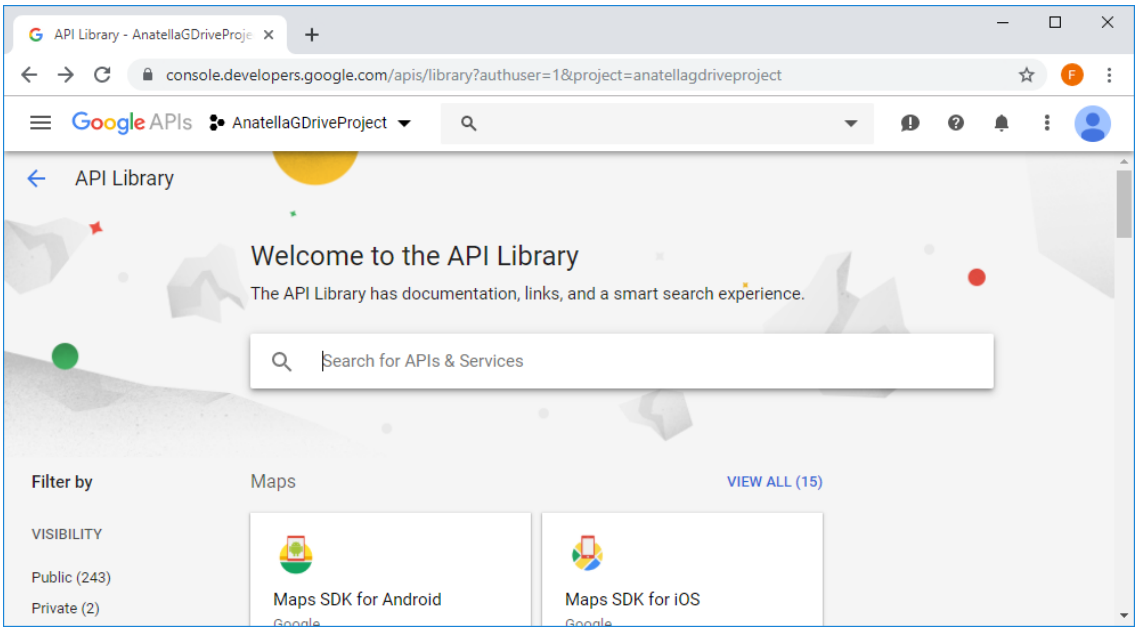
6. We will now Enable the "**BigQuery API**":
This API is required to be able to use all the "Google Big Query" Actions: These actions are:
   - Execute a SQL query on Big Query (section 5.23.21)
   - Upload CSV files to Big Query (section 5.23.22)

6.1. We are back on the Project Selection screen (and the "Google Cloud Storage" logo is not visible anymore: it's written "APIs & Services" instead:). Click on the "Enable APIS and Service" button:

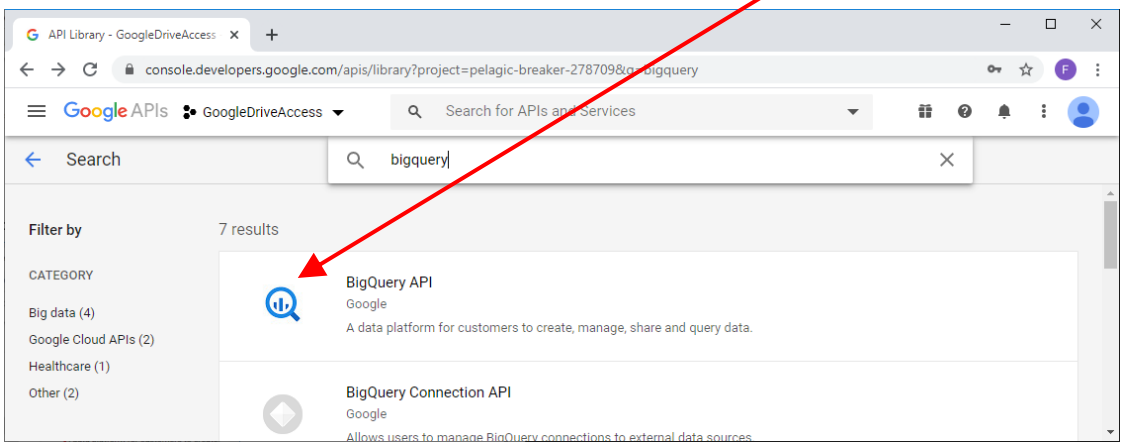


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

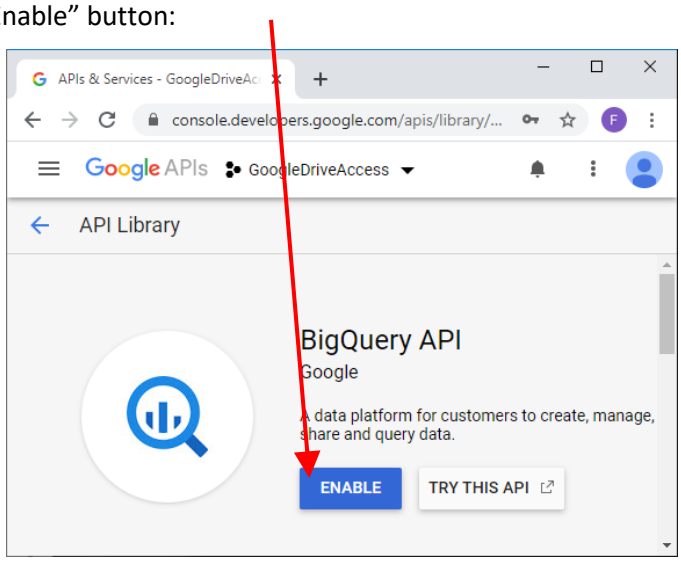
You arrive on this page:



6.2. Search for the “BigQuery API” and click on the first item found:

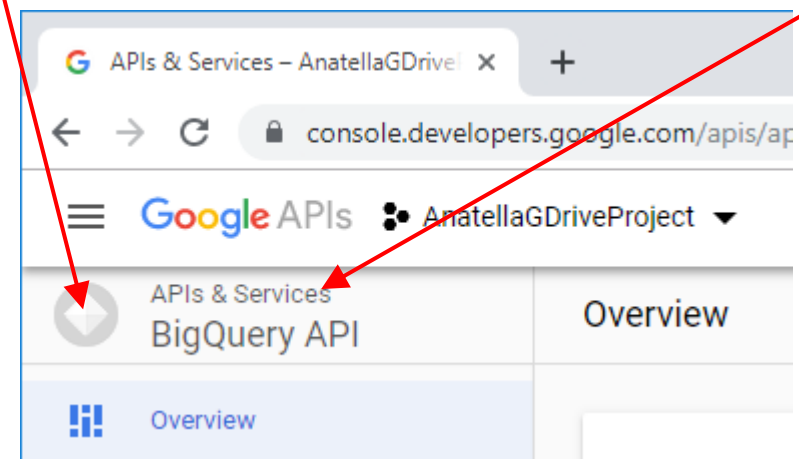


6.3. Click the “Enable” button:



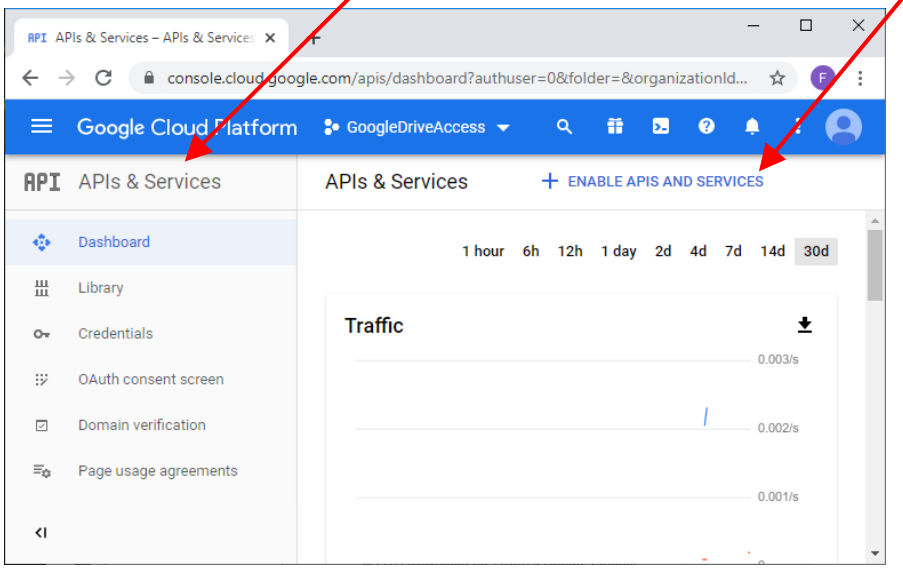
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6.4. We are back on the Project Selection screen but the "Big Query API" logo is now visible here: (and we don't want that). Click on the "APIs and Services" text here:



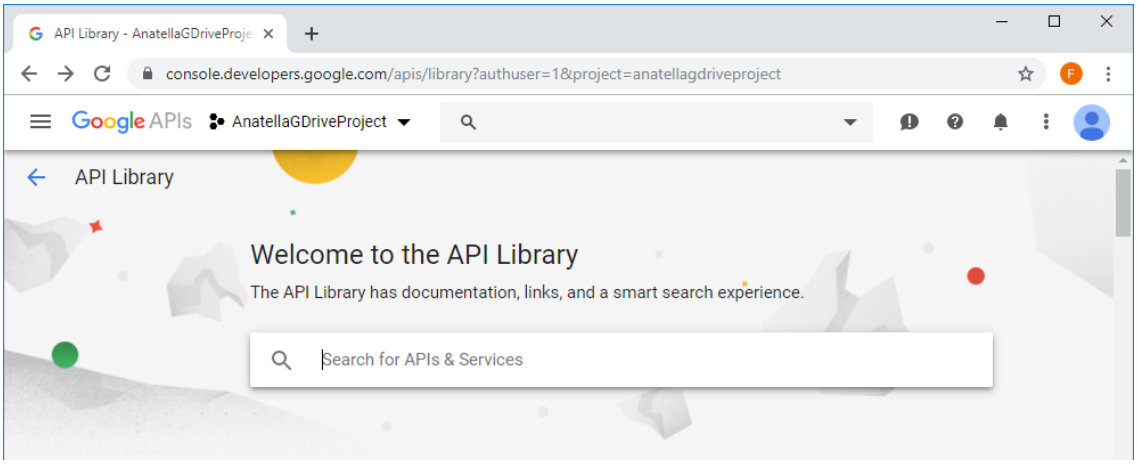
7. We will now Enable the "**Google Analytics API**":
This API is required to be able to use the "Google Analytics" Action from section 5.23.23.

7.1. We are back on the Project Selection screen (and the "BigQuery API" logo is not visible anymore: it's written "APIs & Services" instead:). Click on the "Enable APIS and Service" button:

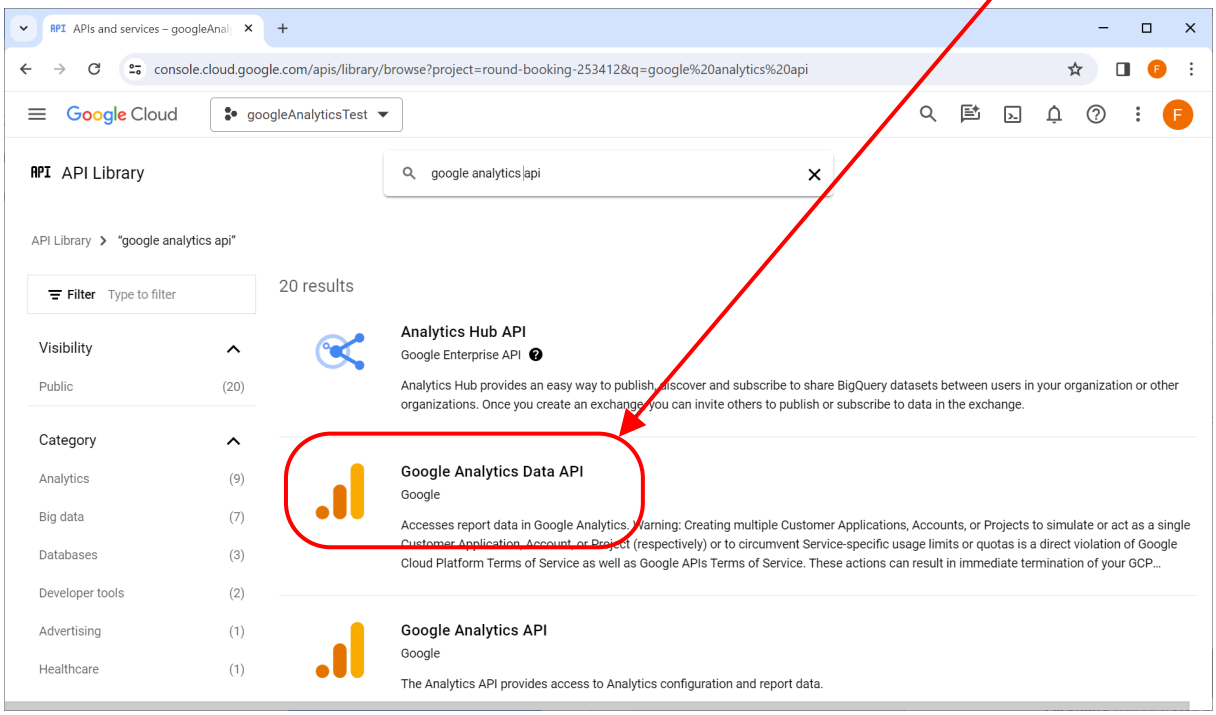


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

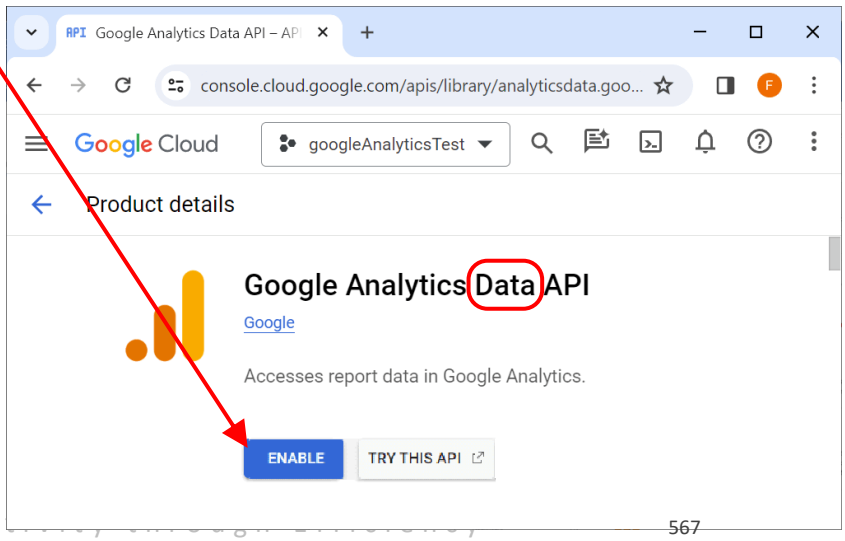
You arrive on this page:



7.2. Search for the “Google Analytics **DATA** api” and click on the second item found:

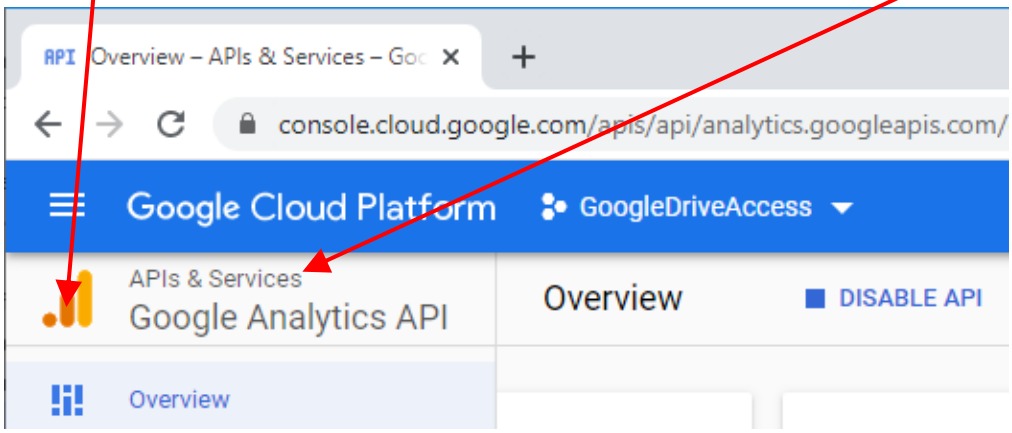


7.3. Click the “Enable” button:



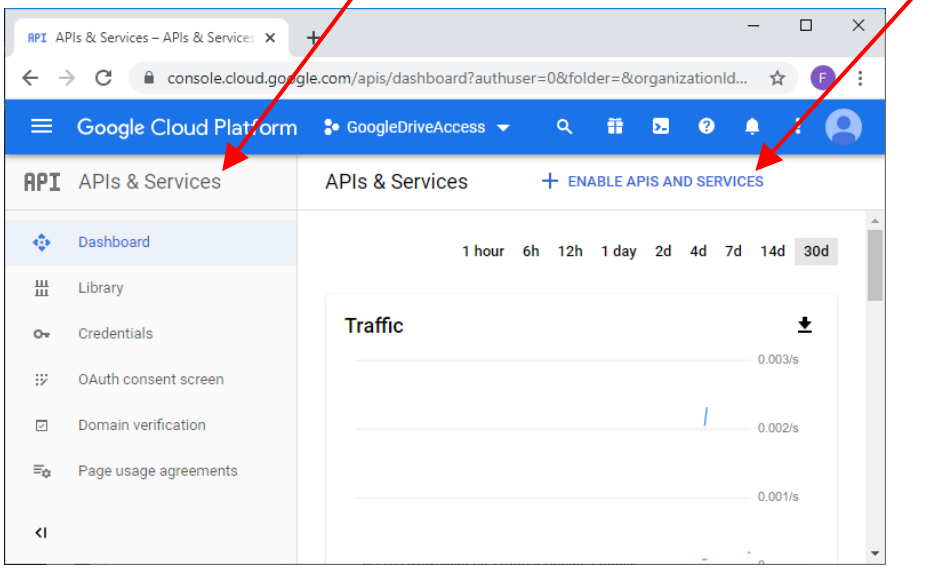
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

7.4. We are back on the Project Selection screen but the "Google Analytics **DATA** API" logo is now visible here: (and we don't want that). Click on the "APIs and Services" text here:



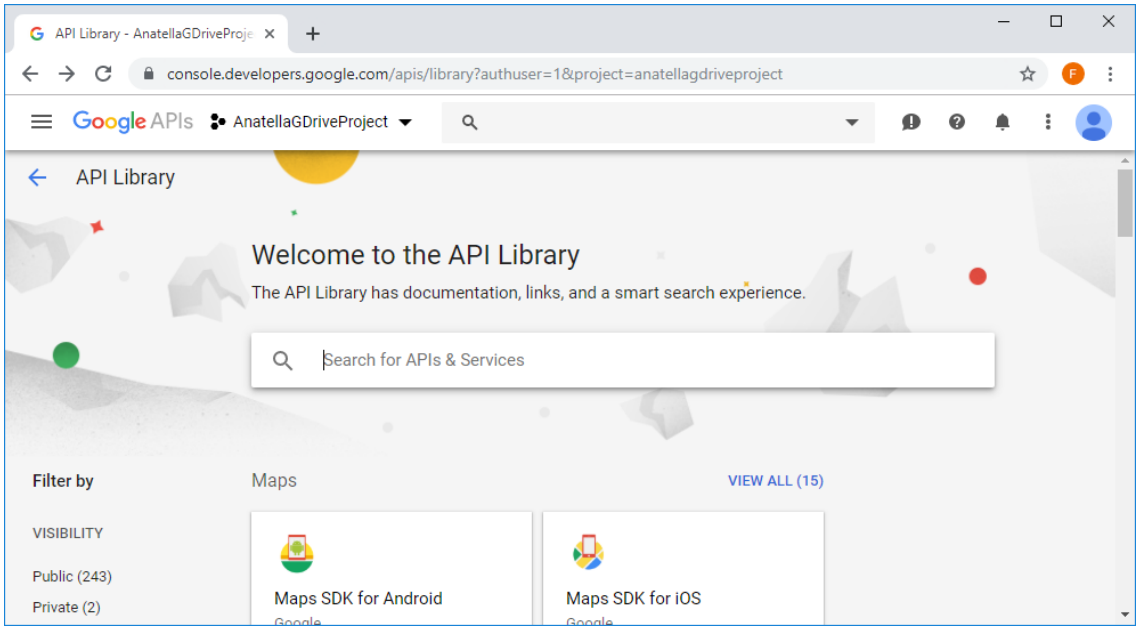
8. We will now Enable the "**Google Speech-to-Text API**":
This API is required to be able to use the "Google Speech-to-Text" Action from section 5.23.48.

8.1. We are back on the Project Selection screen (and the "Google Analytics API" logo is not visible anymore: it's written "APIs & Services" instead:). Click on the "Enable APIS and Service" button:

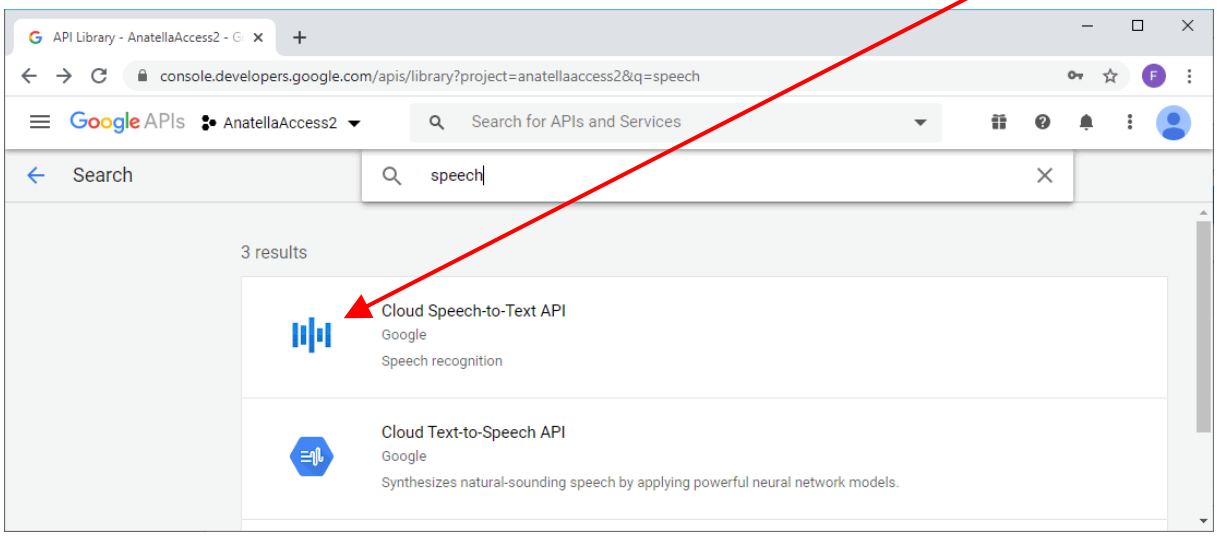


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

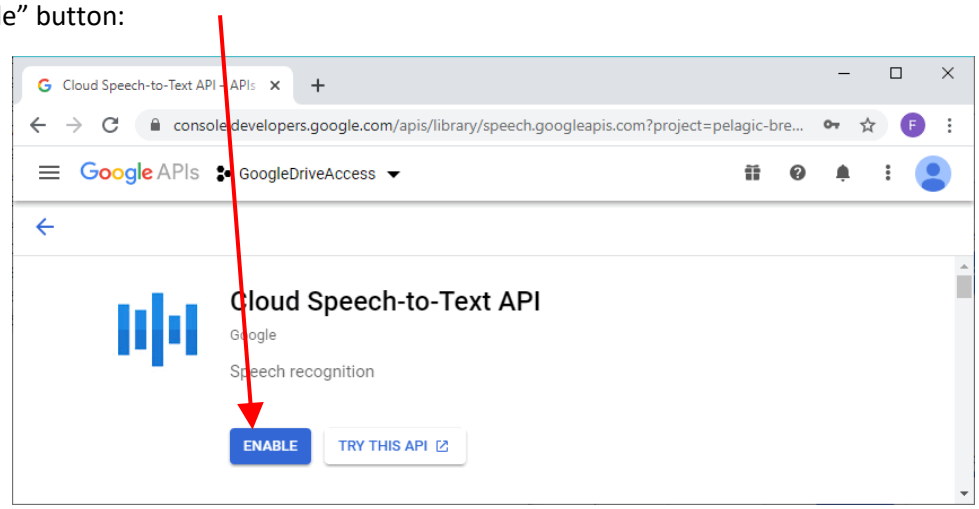
You arrive on this page:



8.2. Search for the “google speech to text” and click on the first item found:

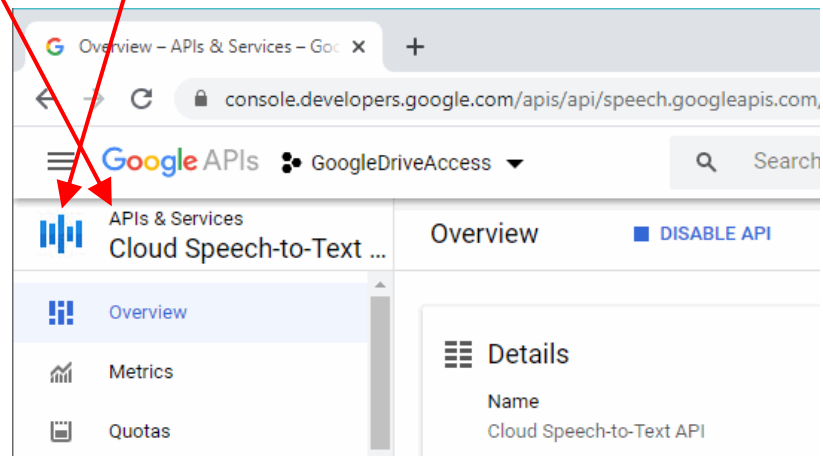


8.3. Click the “Enable” button:



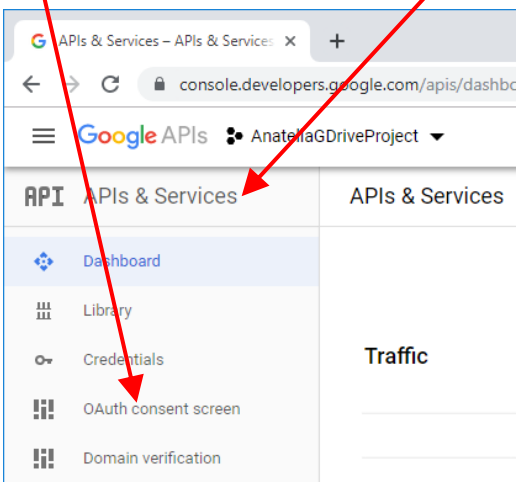
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

8.4. We are back on the Project Selection screen but the "Cloud Speech-to-Text" logo is now visible here: (and we don't want that). Click on the "APIs and Services" text here:

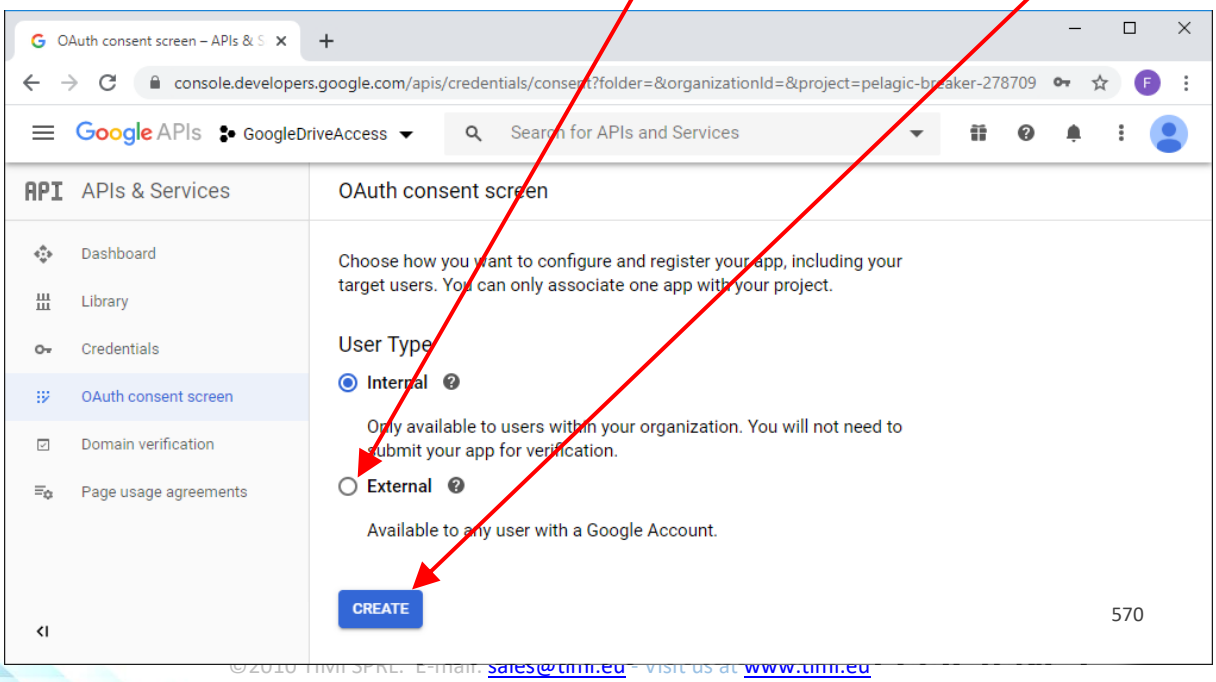


9. Configure OAuth consent Screen

9.1. We are back on the Project Selection screen (and the "Google Analytics API" logo is not visible anymore: it's written "APIs & Services" instead). Click on the "OAuth consent screen" button:



9.2. On the next screen you must select "External" and click the "create" button:

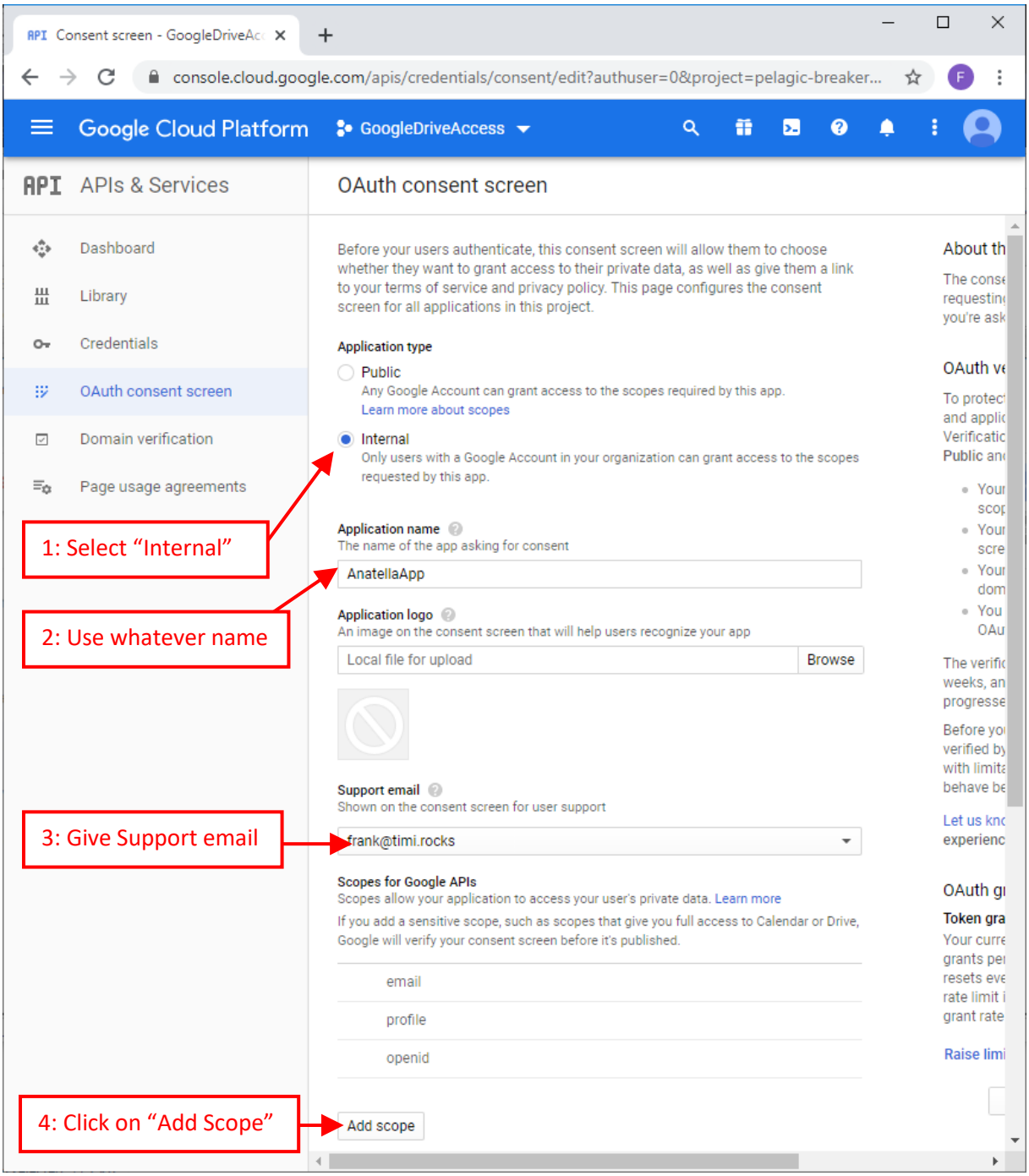


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here, it's easier to select the "External" option. The only downside of the "External" option is that you are limited to 100 different "test credentials" (but, hopefully, you need only ONE credential to automate everything with Anatella! 😉). You can also have more than 100 credential but then you need to publish your application to the general audience (and that's a little bit more work because you need to to make it validated by the google team).

The "Internal" option also works with Anatella but you'll have to pay.

9.3. On the next screen, we'll configure the "App Name" and the "Support Email".
Once this is done, click on the "Add Scope" to configure the scope of your app:

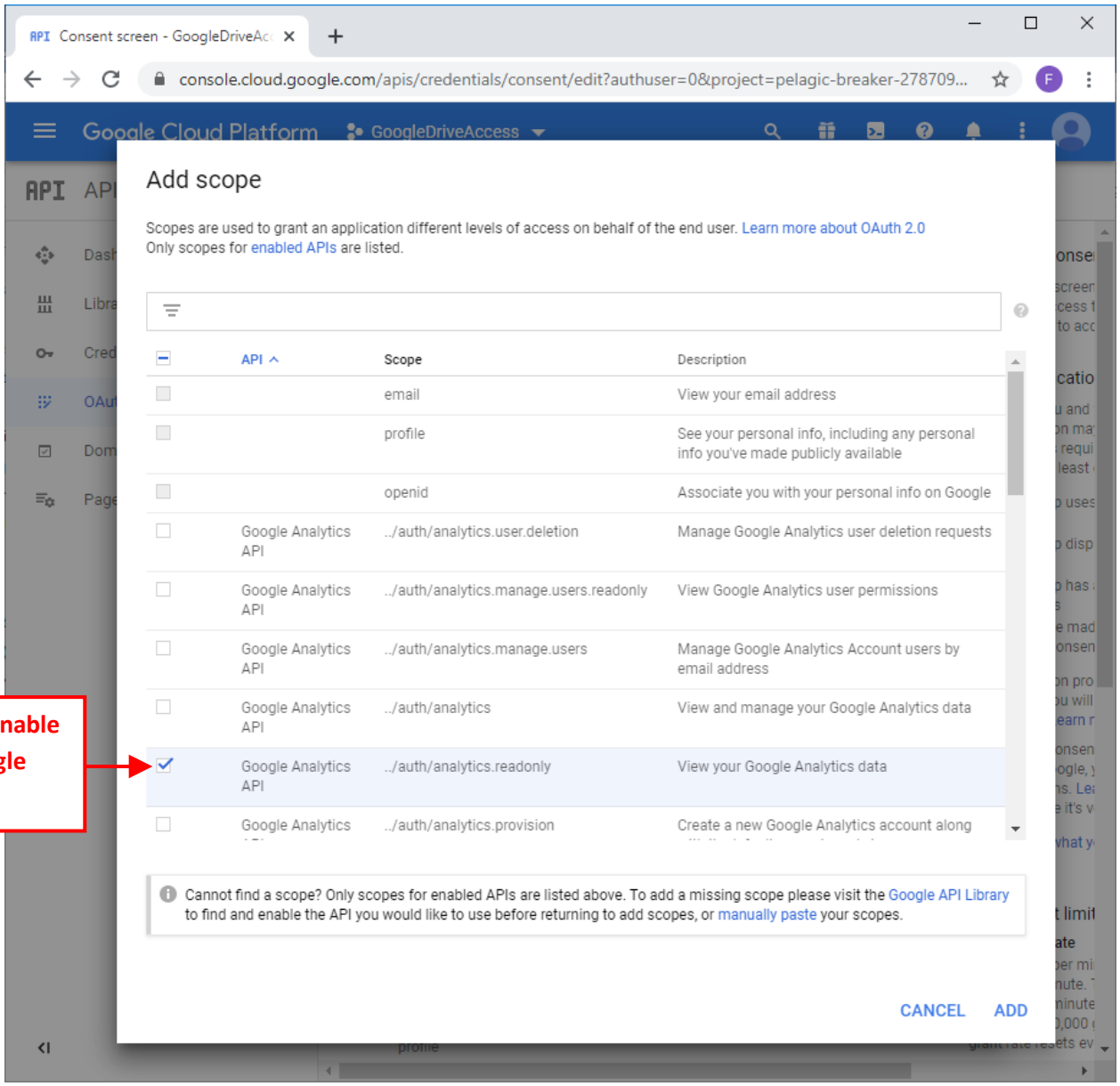


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.4. Enable the following scopes:

On the next configuration screen, pay attention to the "Scopes": For security reasons, you may want to limit the scopes to the minimum. For example, if you only intend to...

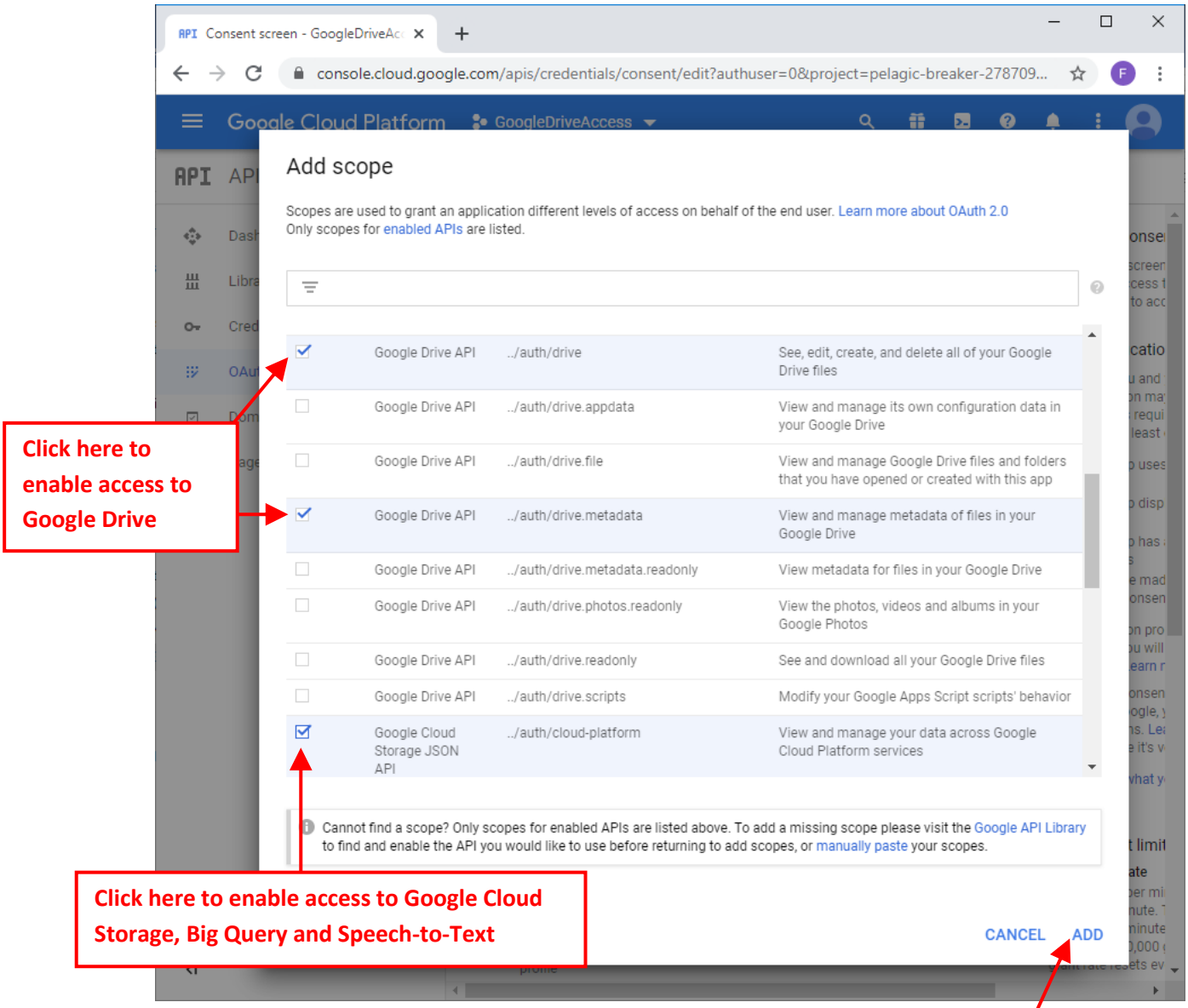
- ...use Google Analytics: Activate only the following scope:
  `../auth/analytics.readonly`
- ...use GDrive: Activate only the following 2 scopes:
  `../auth/drive.metadata`
  `../auth/drive`
- ...use Google Storage, Big Query or Speech-to-Text: Activate only the following scope:
  `../auth/cloud-platform`



Click here to enable access to Google Analytics

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

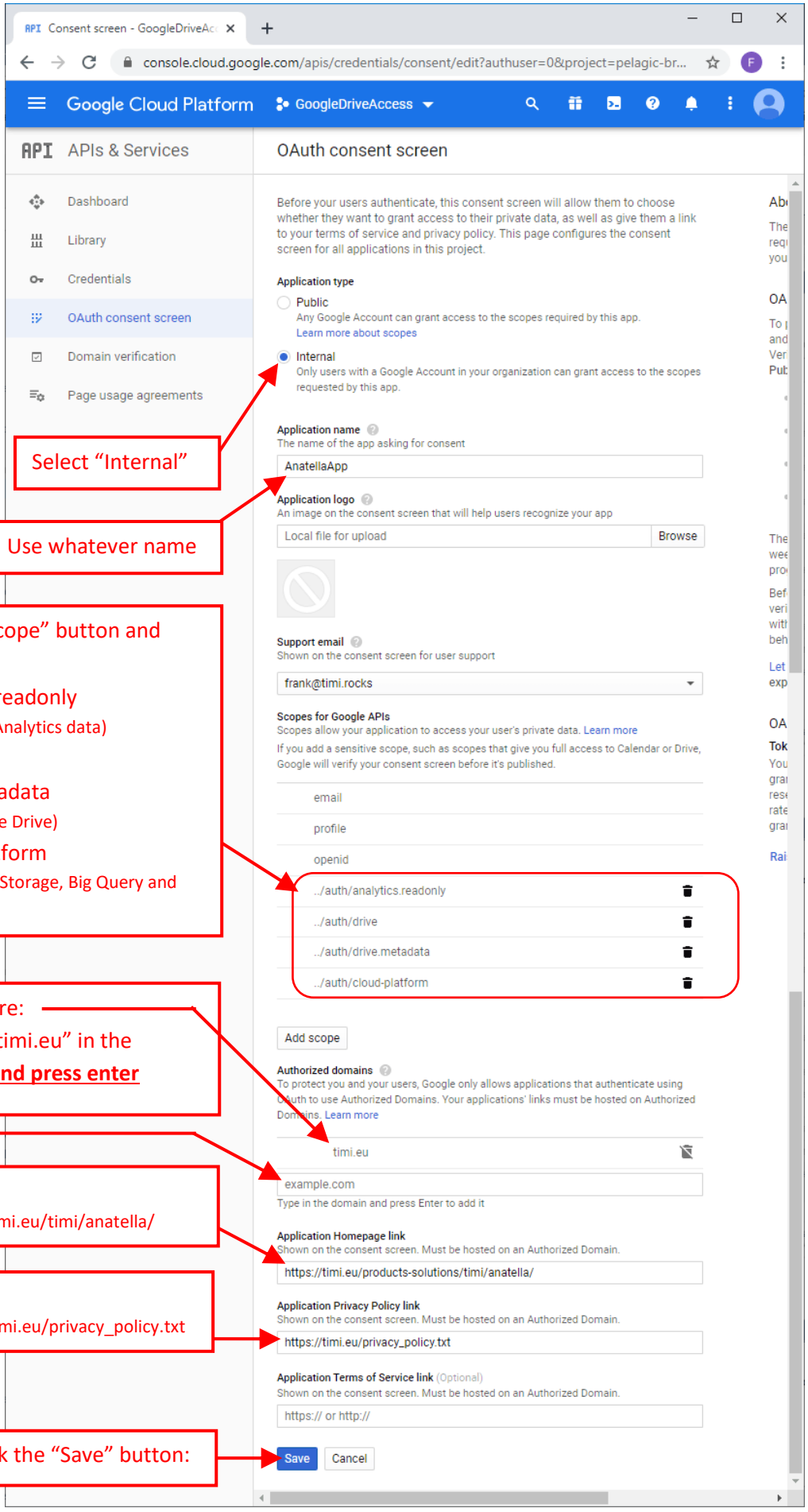
Scroll down a little:



When you have finished selecting your “scopes”, click on the “Add” button:

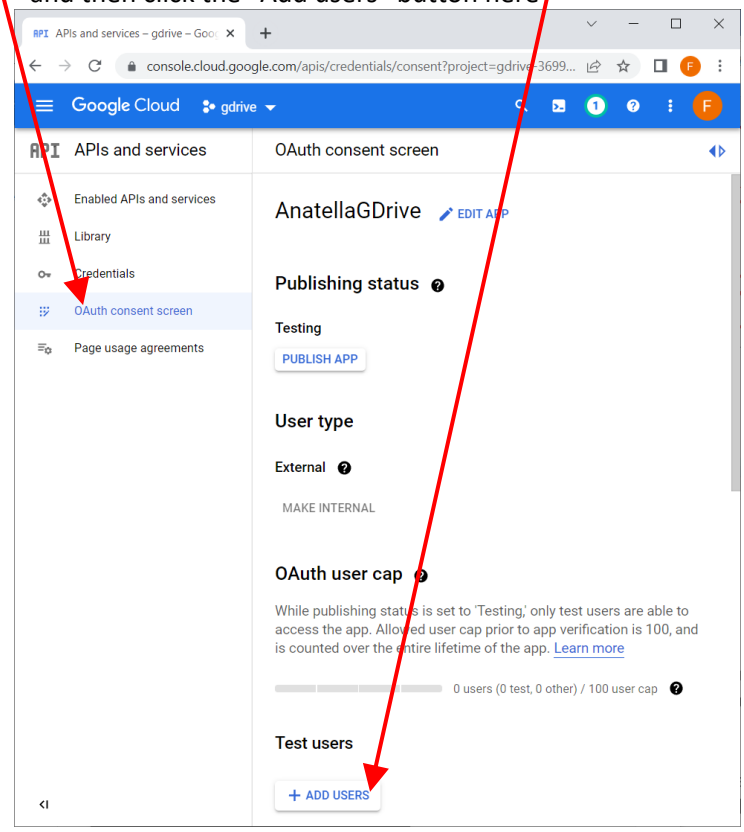
9.5. Finish configuring the OAuth consent Screen.
At the end, you should have something like this:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

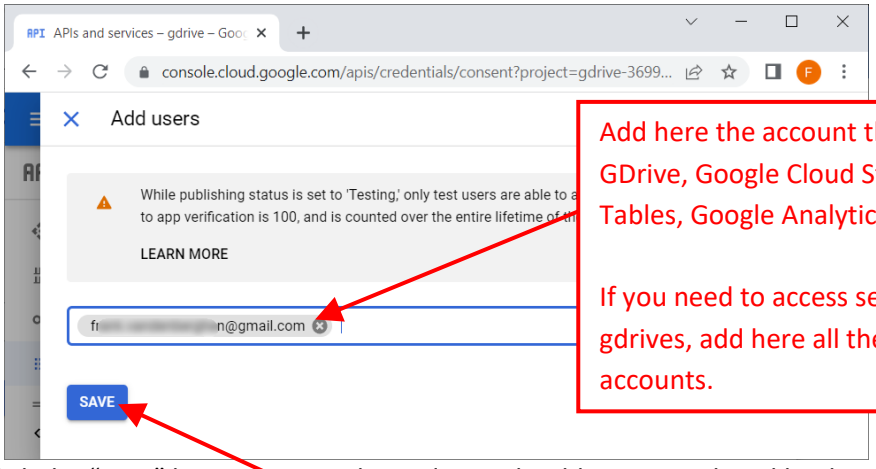


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

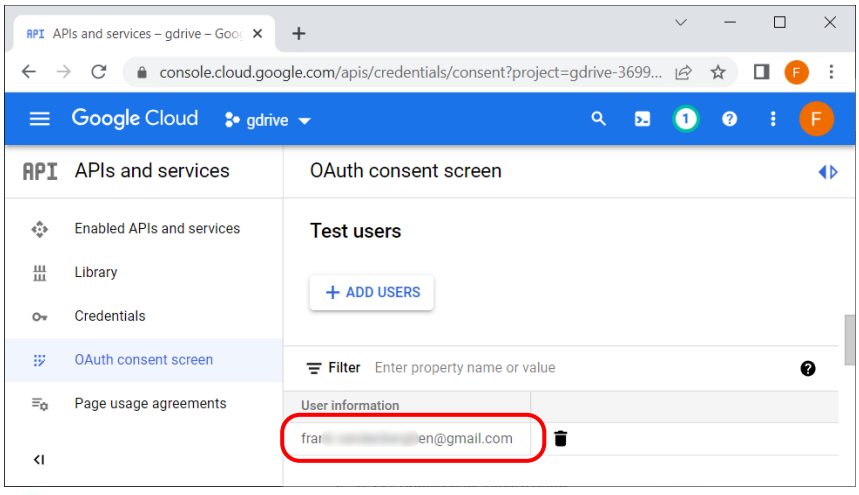
9.6. Add a test user: Click on “OAuth consent screen” in the left menu and then click the “Add users” button here



9.7. Add the email from your “test user” (that will use your application):

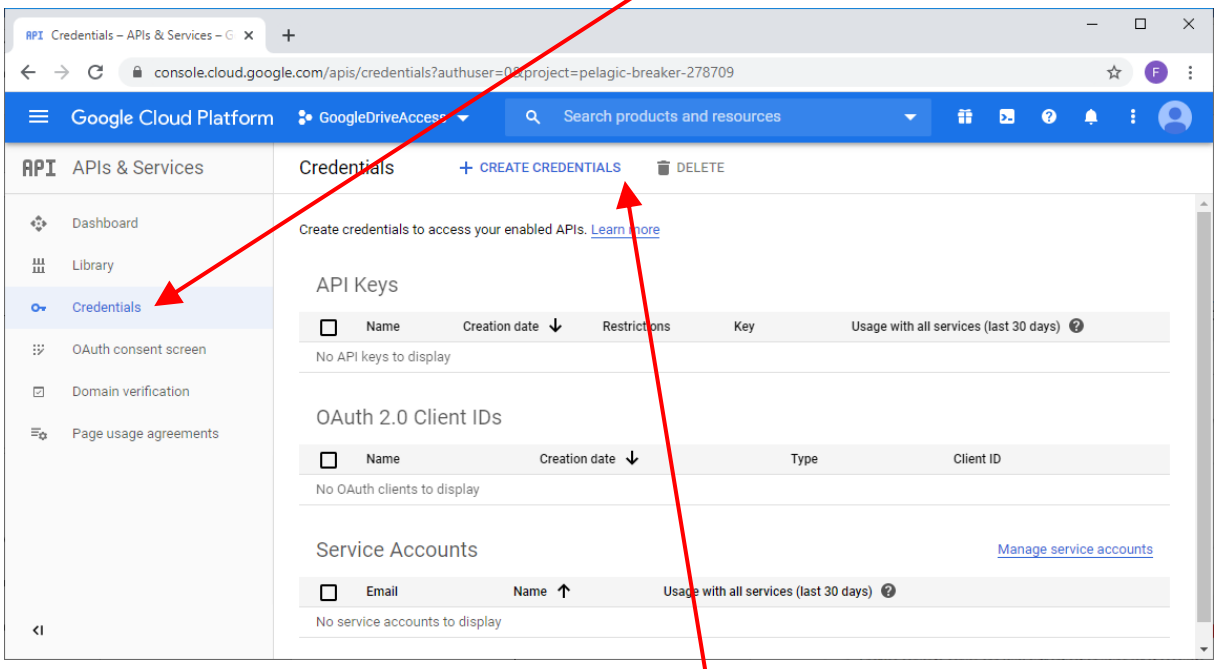


…and click the “save” button . At the end, you should see something like this:



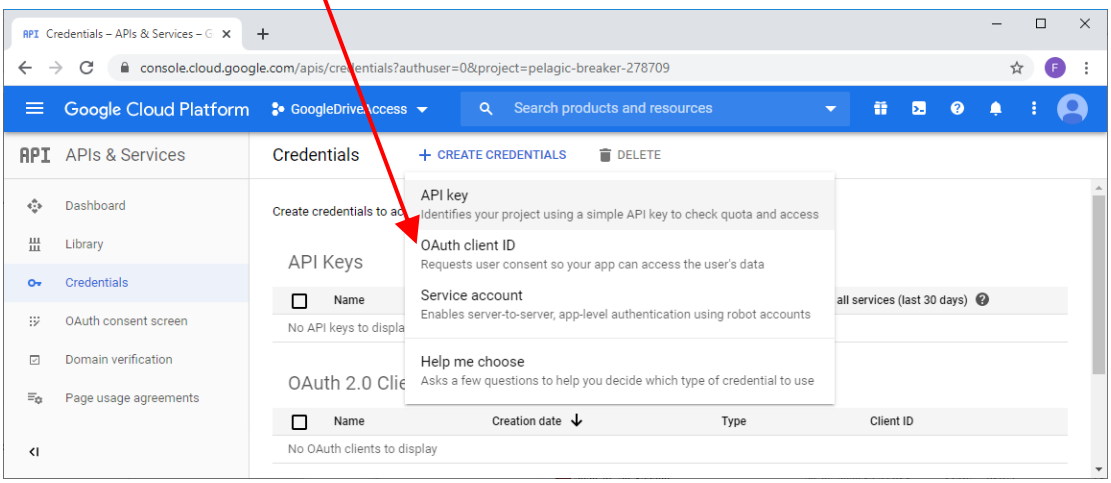
10. Get your Credentials

10.1. Click the “Credentials” option in the left pane:



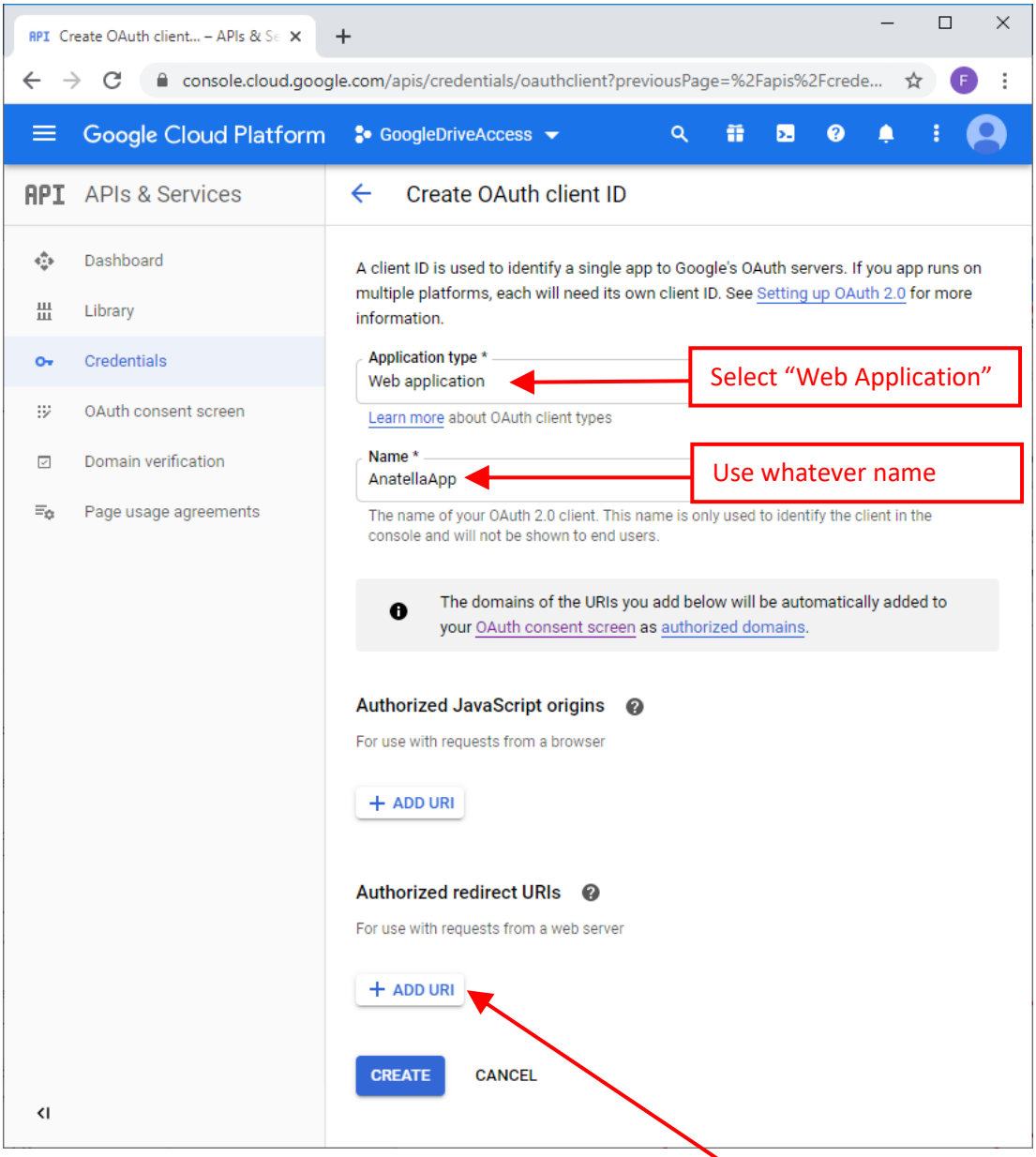
10.2. Click the “Create credentials” button in the right panel:

10.3. Select “OAuth Client ID”:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.4. Enter the following:



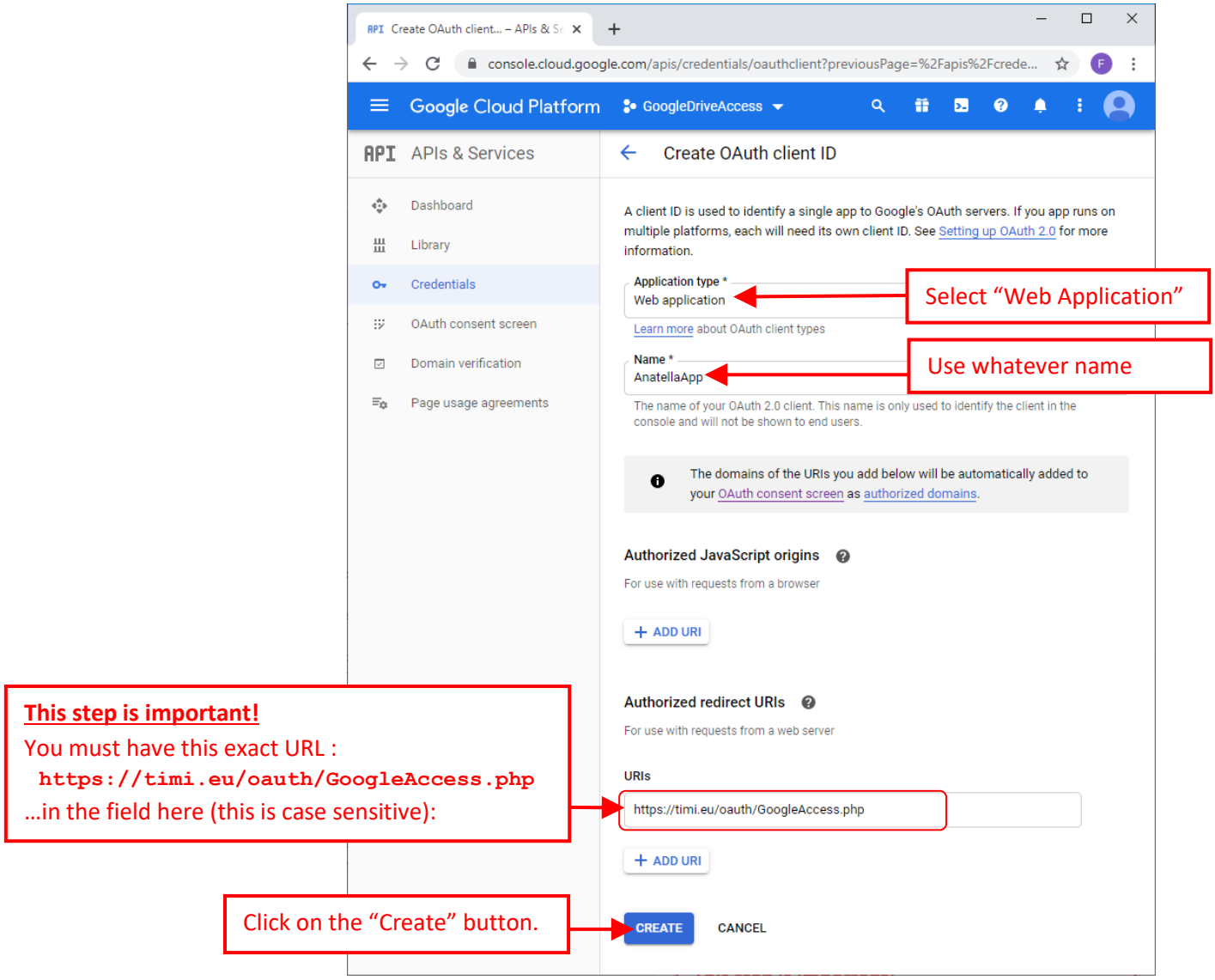
10.5. **This step is important!**

Click the "ADD URI" button in the "*Authorized redirect URIs*" section:

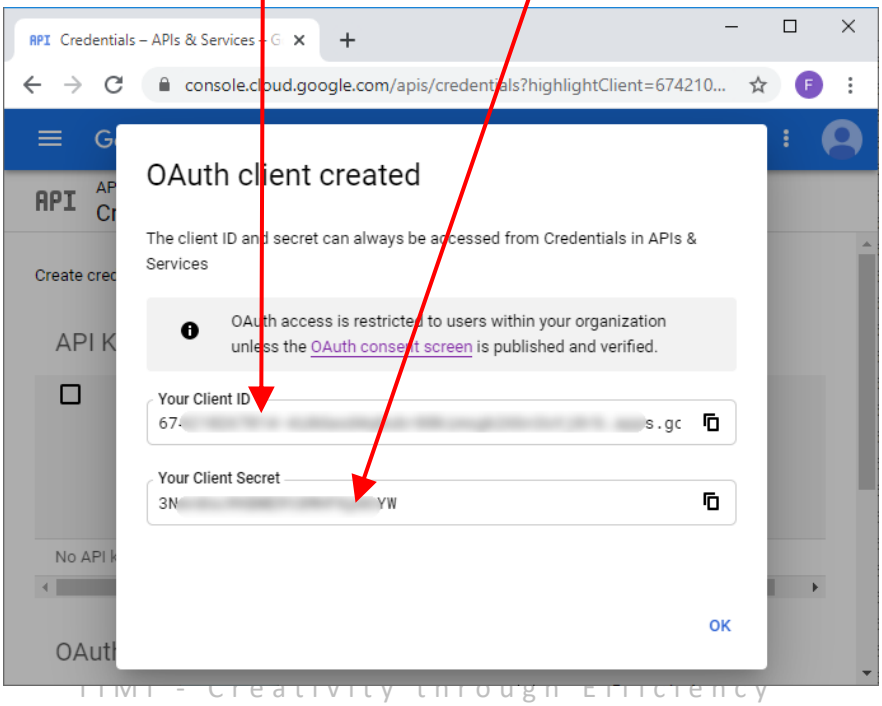
...and enter: `https://timi.eu/oauth/GoogleAccess.php`

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.6. At the end you should have something like this:

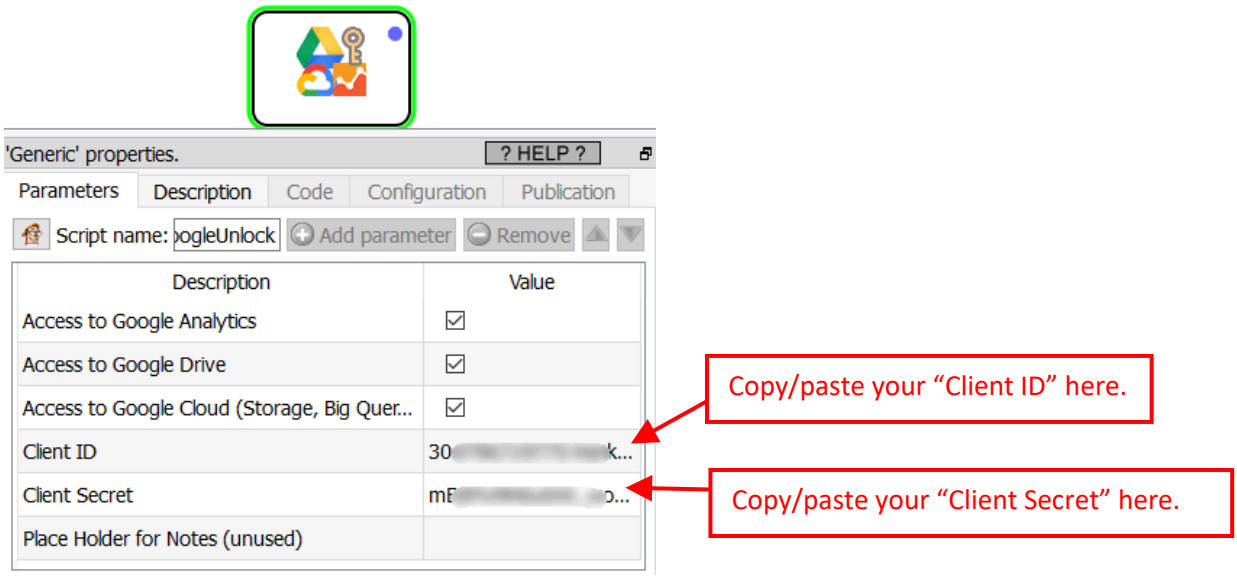


10.7. You receive your "Client ID" and "Client Secret":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

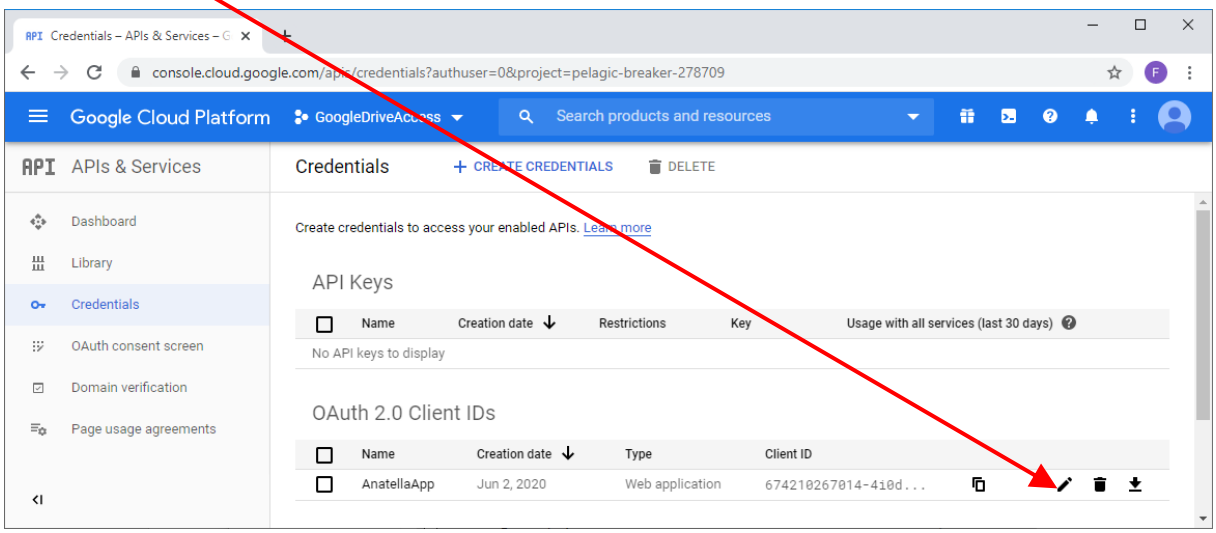
Copy/paste your your “Client ID” and “Client Secret” inside Anatella here:



Inside the Anatella action, the 3 parameters that are named “*Access to…*” must match the scopes that you selected at step 9.4. For security reasons, you may have limited the scopes to the minimum. For example, if you only intend to...

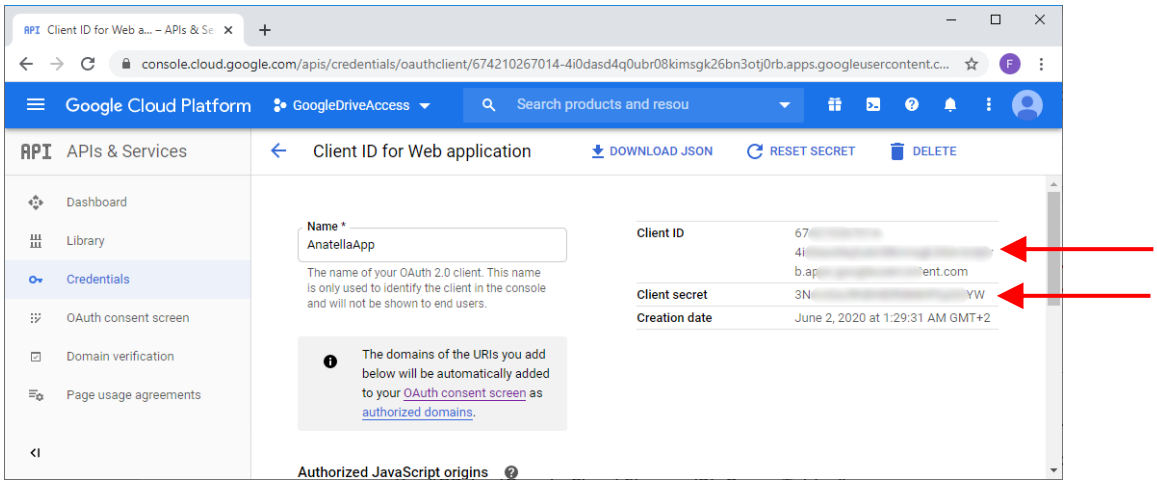
- ...use Google Analytics: You only activated the following scope:
  `../auth/analytics.readonly`
- ...use GDrive: You only activated the following 2 scopes:
  `../auth/drive.metadata`
  `../auth/drive`
- ...use Google Storage, Big Query or Speech-to-Text: You only activated the following scope:
  `../auth/cloud-platform`

Don’t worry if the Window in your browser closes before you can copy/paste your “Client ID” and “Client Secret”: You can always see again later these 2 informations by clicking the “edit” icon here:

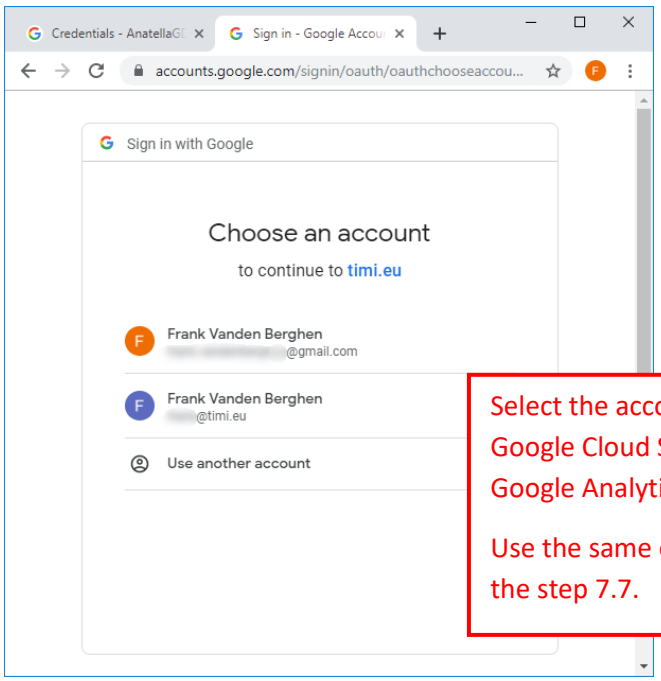


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

…Then, you'll again see your "Client ID" and "Client Secret" here:



10.8. Run (i.e. click the output pin) the UnlockGoogle Action in Anatella (with the correct "Client ID" and "Client Secret" obtained from the previous step): A web browser opens: Select the account that owns the "GDrive/Google Cloud Storage/BigQuery Tables":



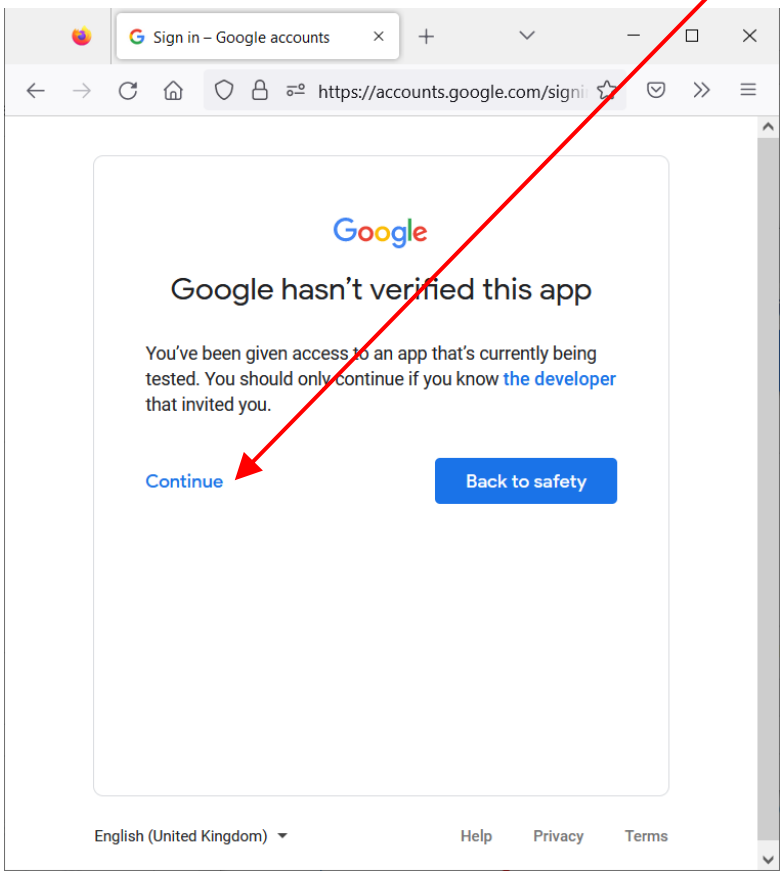
Select the account that owns the GDrive, Google Cloud Storage/BigQuery Tables, Google Analytics data.

Use the same email(s) as you used during the step 7.7.

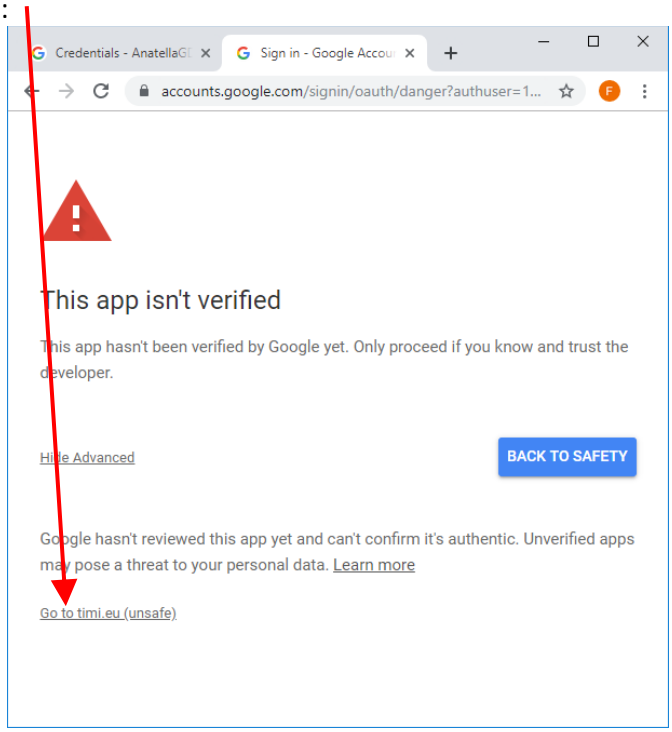
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.9. The following 3 confirmation screens do not appear all the time:

10.9.1. Since your (totally new) application isn't verified, you get a warning message. This is perfectly normal and expected. Just click on the "Continue" url link:



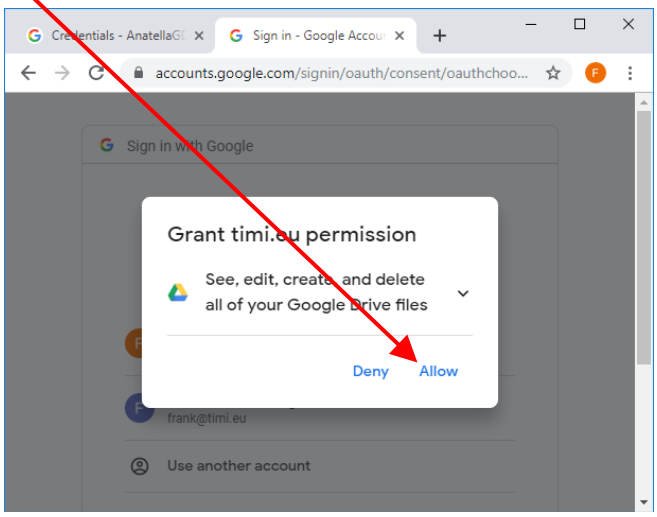
10.9.2. Confirm that you trust your own application by clicking the "Go to timi.eu (unsafe)" url link here:



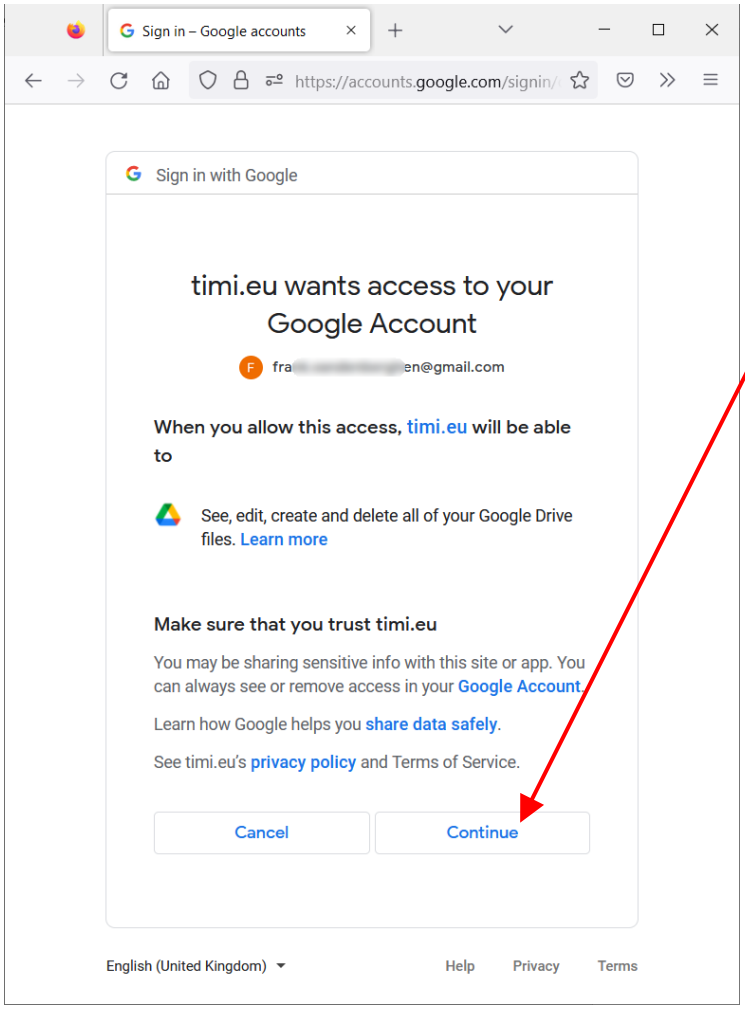
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.10. The following confirmation screens does not appear all the time:

10.10.1. Confirm that you want to access your gdrive files from Anatella: Click the “Allow” button:

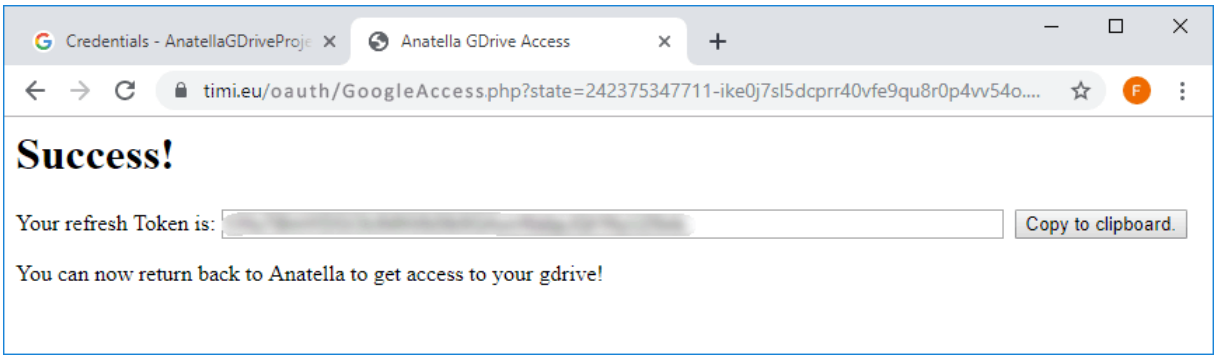


10.10.2. Sometime, an additional confirmation window opens. Just click the “Continue” button:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.11. Finally, you receive your “Refresh Token”!

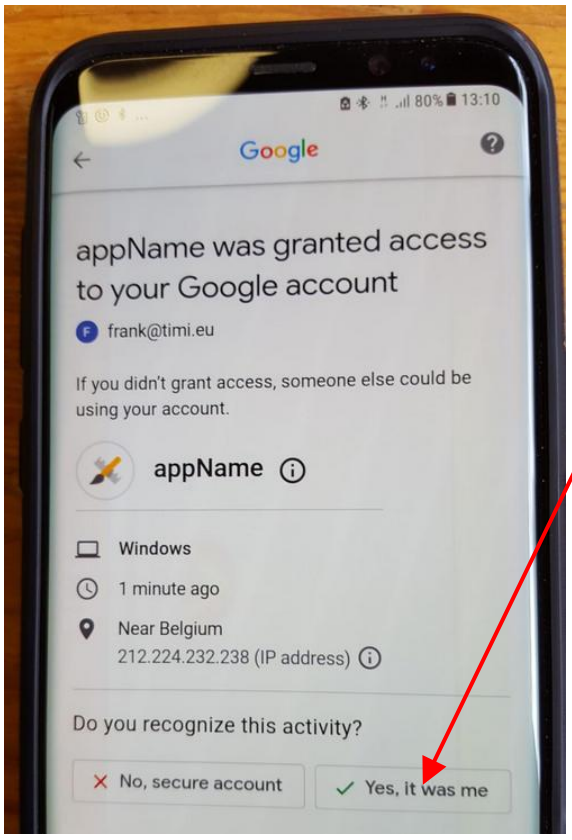


If you don’t see any “Refresh Token” (i.e. the “Refresh Token” field is empty), then you’ll need to redo the whole procedure starting from the step 10.1.

Please write your “Refresh Token” in a safe place. There is no way to retrieve your “Refresh Token” in anyway: If you lose it, you’ll need to redo the whole procedure starting from the step 8.1. You now have your complete credentials:

- A “Client ID” (obtained at step 8.7)
- A “Client Secret” (obtained at step 8.7)
- A “Refresh Token” (obtained at this step 8.11)

11. After a few seconds, you should also receive on your mobile phone a security message that warns you that someone accessed your Google Account. Just click the “Yes, it was me” button:



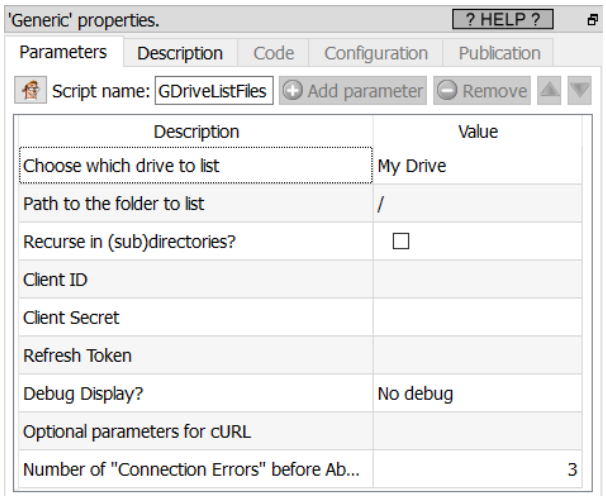
12. Save your “Refresh Token” in a safe place.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.12. List the files on a GDrive

Icon:

Property window:



Short description:
Lists the files on a Google Drive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to use this Action, you need to get these 3 parameters from Google:

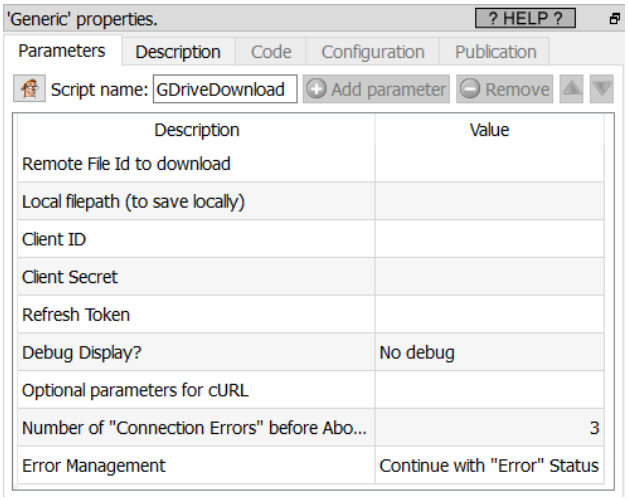
- your “Client ID”
- your “Client Secret”
- your “Refresh Token”

To get these 3 parameters, you must use the “Unlock Google Services” action detailed in section 5.23.11.

5.23.13. Download files from a GDrive

Icon:

Property window:



Short description:
Download files from a Google Drive

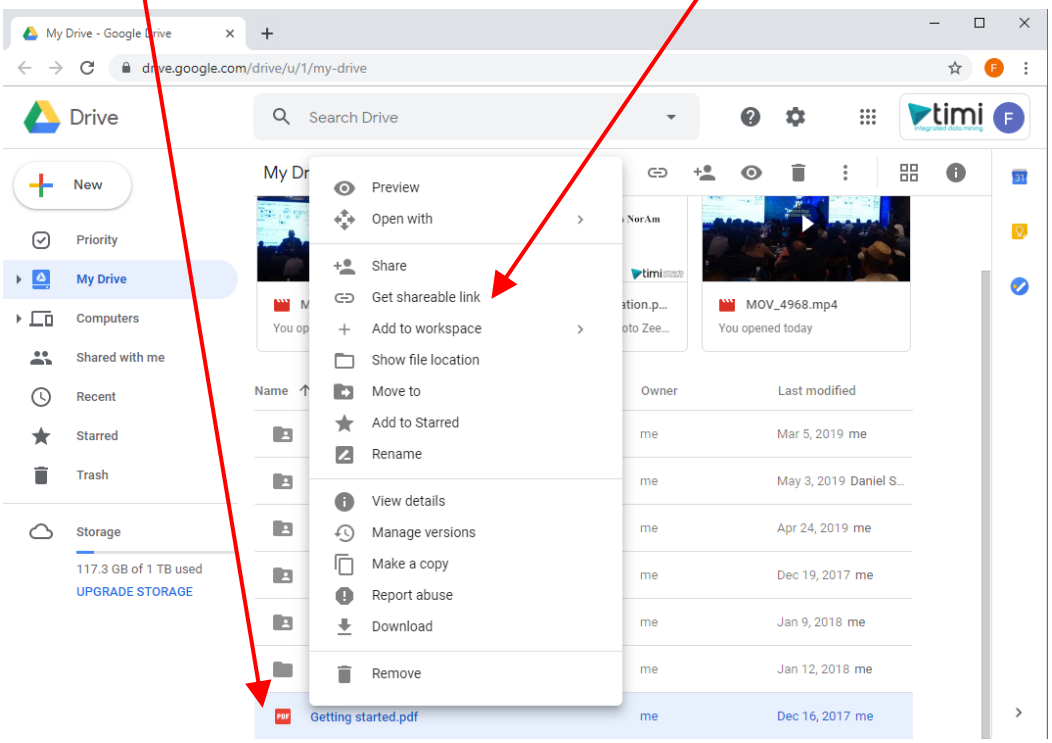
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to use this Action, you need to get these 3 parameters from Google: (1) your “Client ID”, (2) your “Client Secret”, (3) your “Refresh Token”. To get these 3 parameters, you must use the “Unlock Google Services” action detailed in section 5.23.11.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The GDriveDownload Action expects a table with (at least) 2 columns in input:

1. A column containing the "Remote File ID to download": All the files in a gdrive are uniquely identified by their "File ID" (and not by their filepath). So, to download a file from your GDrive, you first need to get its "File ID". There are two ways to get a "File ID":
   - Use the GDriveList Action from the previous section (section 5.23.11): This action gives as output a column named "FileID" that contains thea "File ID" that you want.
   - Navigate on your GDrive (using a web-browser) to the file that you want to download and right-clik it: . Then, select the "Get shareable link" option:



   You receive an URL inside your clipboard that looks like this:
   https://drive.google.com/open?id=0BwGsipwcl6WCc3RhcnRlcl9maWxlX2Rhc2hlclYw

   The "File ID" (that you are searching for) is the "id" parameter inside the url.
   For example, in the above url, your "File ID" is:
   0BwGsipwcl6WCc3RhcnRlcl9maWxlX2Rhc2hlclYw

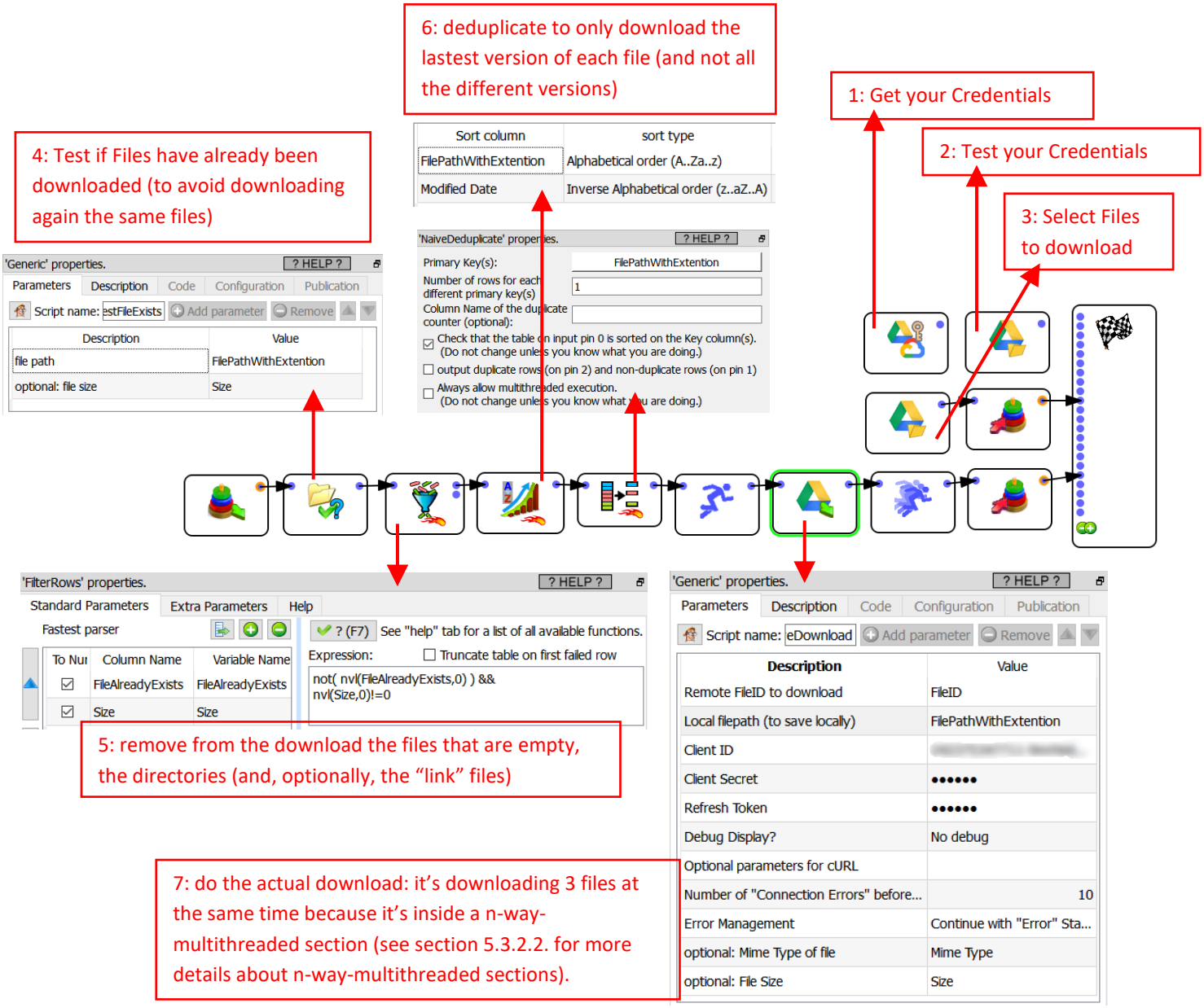
2. A column containing the local filepath (i.e. the file path + the file name + the file extention) of the file(s) to create locally.

   One option to create this second required column is to use the column named "Filename" originating from the GdriveList Action. Unfortunately, this won't always work properly because, for example, the "Google Spreadsheet files" do not have any extention (i.e. they should have a .xlsx extention but it's missing). So, for your convenience, the GdriveList Action also outputs another column named "FilePathWithExtention" that, basically, adds the missing file extentions to the "Google Spreadsheet" files, the "Google document" files, etc.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.13.1. Synchronizing all local files from a source Google Drive.

Let's use the GDriveDownload Action to make a copy on your local hard drive of the content of your Google Drive.



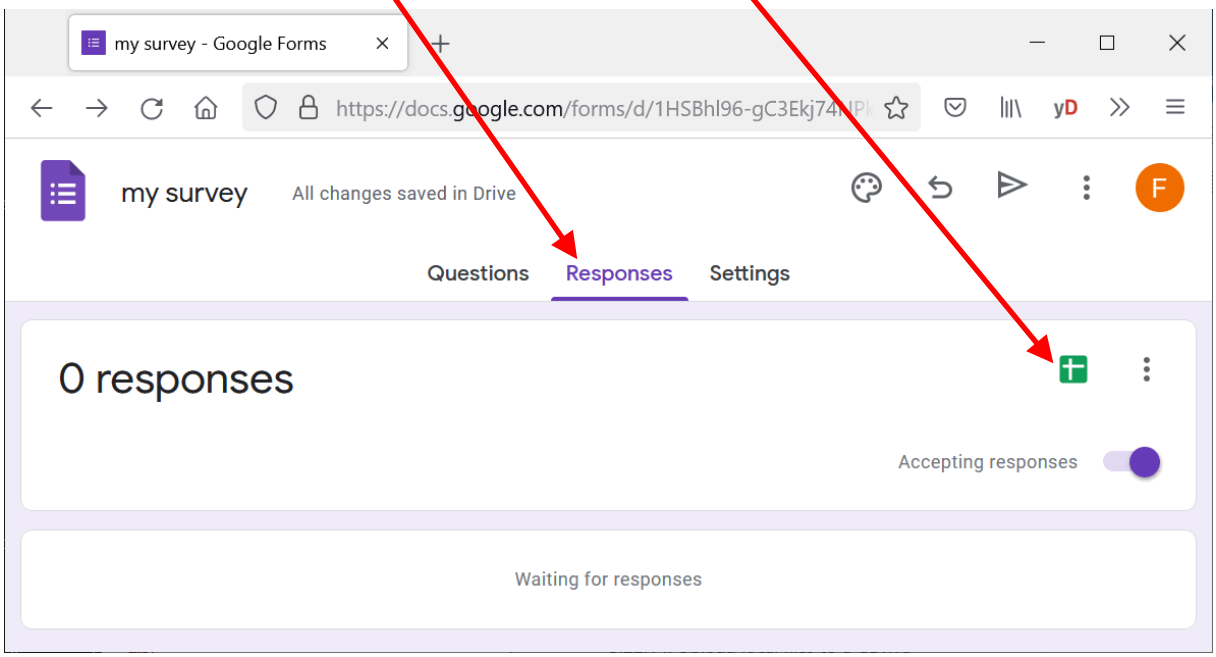
In opposition to the official Google Drive Syncrhonisation tool, the download speed of the GDriveDownload Action is not limited to a very small speed (typically around 100KB/sec). The GDriveDownload Action is using all the bandwidth of your internet connection (typically around 10MB/sec) so that your files are synchronized much faster.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

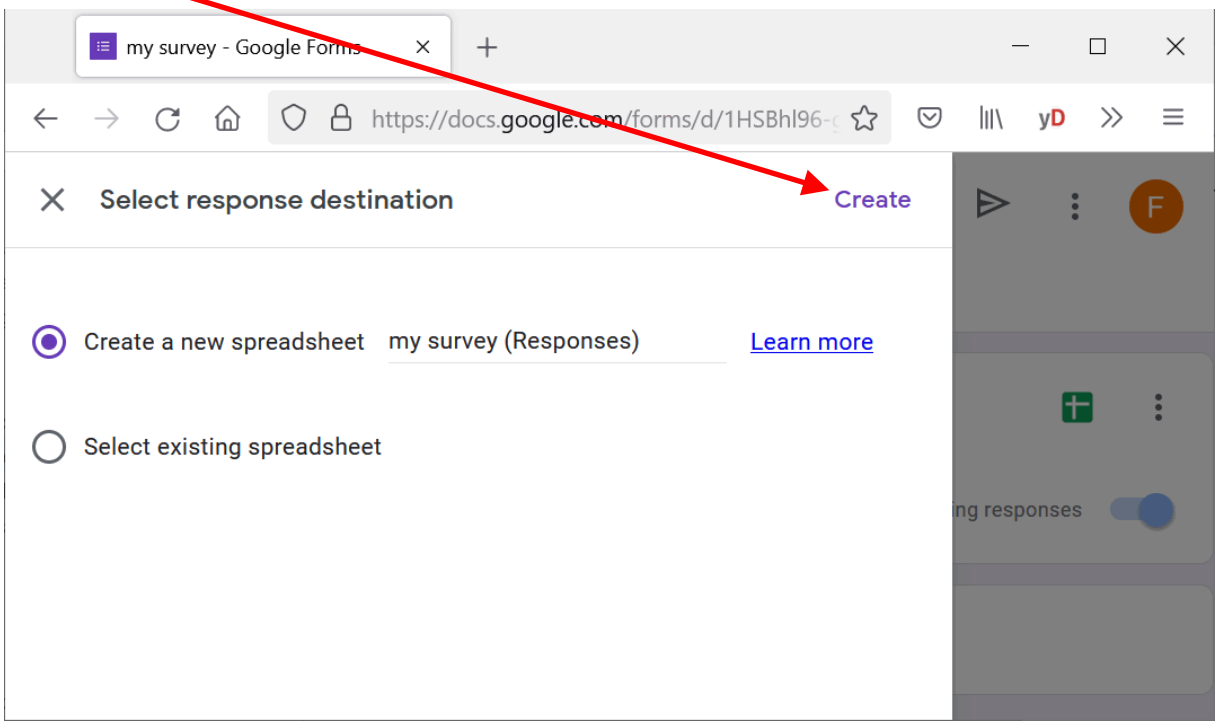
5.23.13.2. Download the responses to a Survey made with Google Forms

Here is the procedure to download the responses to a Survey made with Google Forms:

1. Open the URL https://docs.google.com/forms and navigate to the setup page of your survey.
2. Go to the "Responses" panel: and click on the icon:



3. Click on the "Create" button to create a Spreadsheet that contains all the responses to your survey:



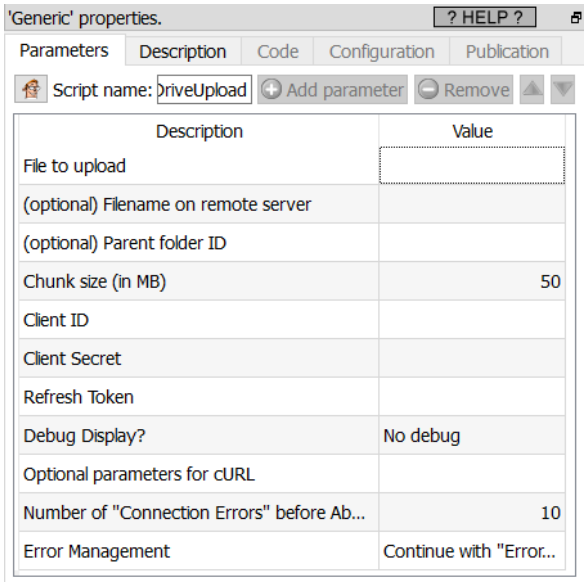
4. You can now download the new Spreadsheet with the GDriveDownload Action to get all the responses to your survey.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.14. Upload local files to a GDrive

Icon:

Property window:



Short description:
Upload local files to a Google Drive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to use this Action, you need to get these 3 parameters from Google:

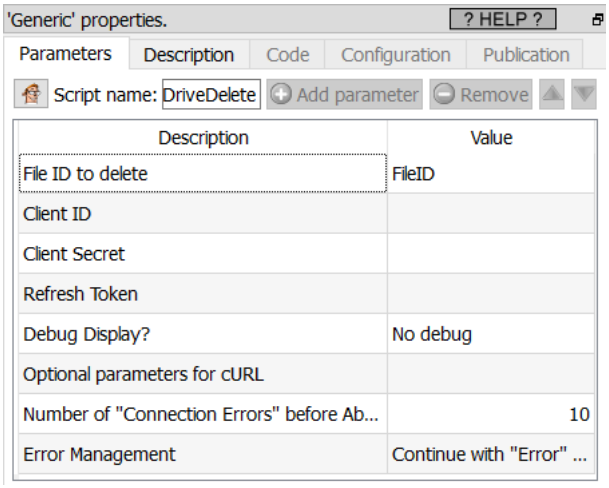
- your “Client ID”
- your “Client Secret”
- your “Refresh Token”

To get these 3 parameters, you must use the “Unlock Google Services” action detailed in section 5.23.11.

5.23.15. Delete files stored on a GDrive

Icon:

Property window:



Short description:
Delete files stored in a Google Drive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to use this Action, you need to get these 3 parameters from Google:

- your “Client ID”
- your “Client Secret”
- your “Refresh Token”

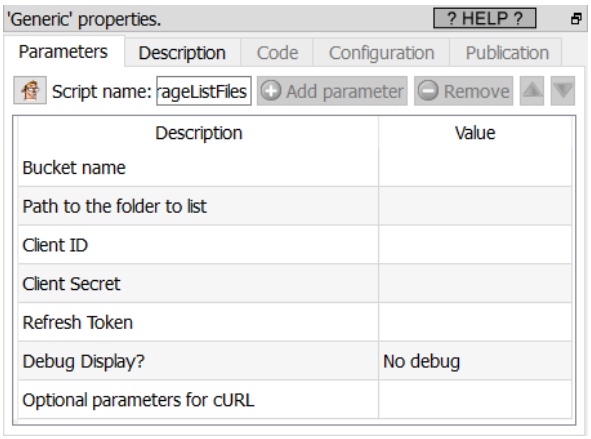
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

5.23.16. List the files on a Google Cloud Storage

Icon:

Property window:



Short description:
List the files on a Google Cloud Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

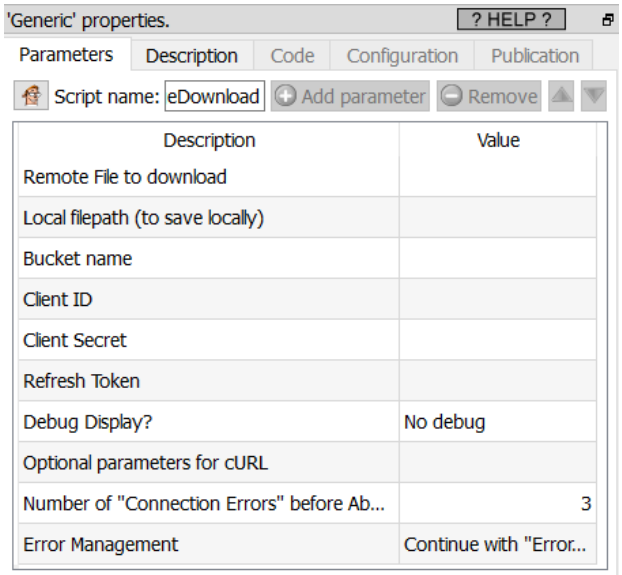
To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

You can get the names of all your accessible "Buckets" using the GoogleStorageListBuckets action detailed in section 5.23.20.

5.23.17. Download files from a Google Cloud Storage

Icon:

Property window:



Short description:
Download files from a Google Cloud Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

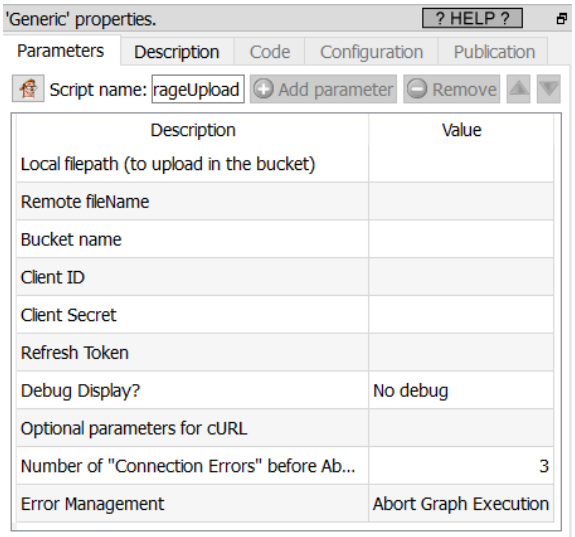
To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

You can get the names of all your accessible "Buckets" using the GoogleStorageListBuckets action detailed in section 5.23.20.

5.23.18. Upload files to a Google Cloud Storage

Icon:

Property window:



Short description:
Upload files to a Google Cloud Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

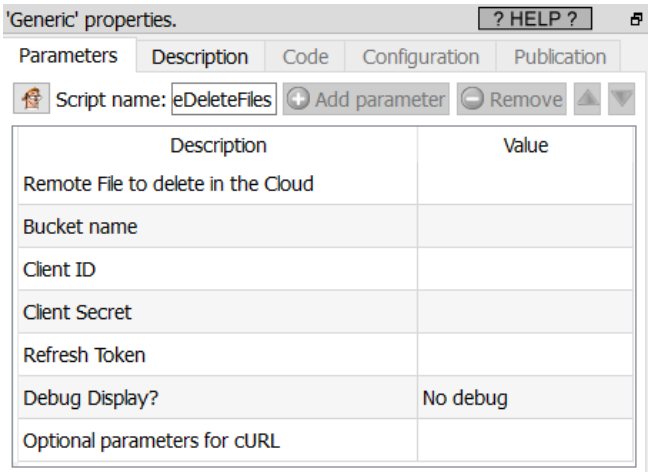
To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

You can get the names of all your accessible "Buckets" using the GoogleStorageListBuckets action detailed in section 5.23.20.

5.23.19. Delete files from a Google Cloud Storage

Icon:

Property window:



Short description:
Delete files from a Google Cloud Storage

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

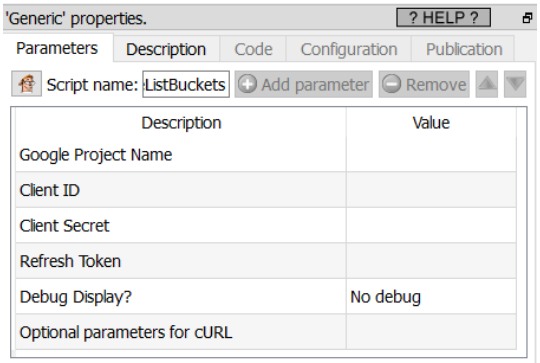
To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

You can get the names of all your accessible "Buckets" using the GoogleStorageListBuckets action detailed in section 5.23.20.

5.23.20. List the buckets on a Google Cloud Storage

Icon:

Property window:



Short description:
List the buckets on a Google Cloud Storage

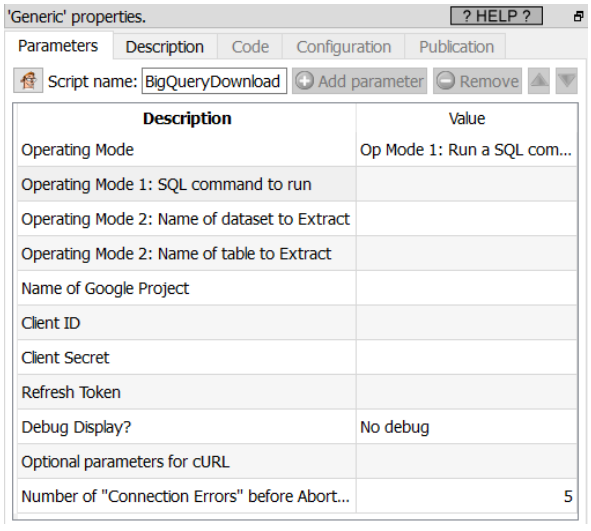
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

5.23.21. Download/Query a table on Google Big Query

Icon:

Property window:



Short description:
Export a complete table from Google Big Query

OR

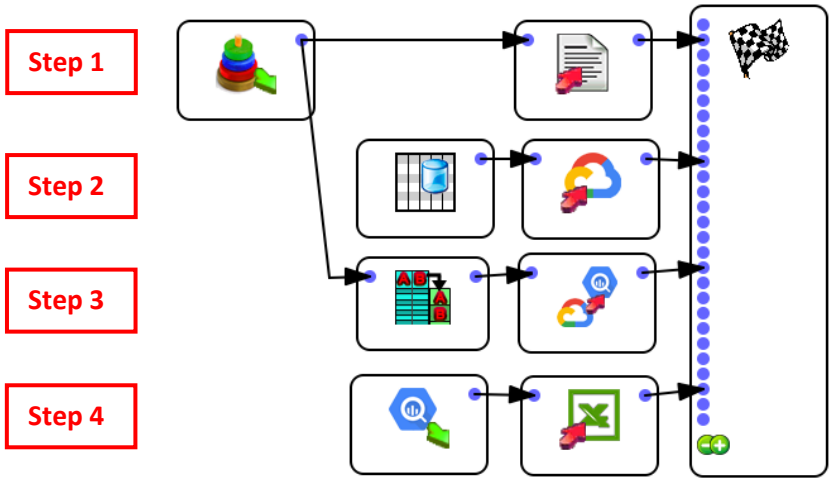
Run any SQL command on Google Big Query

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

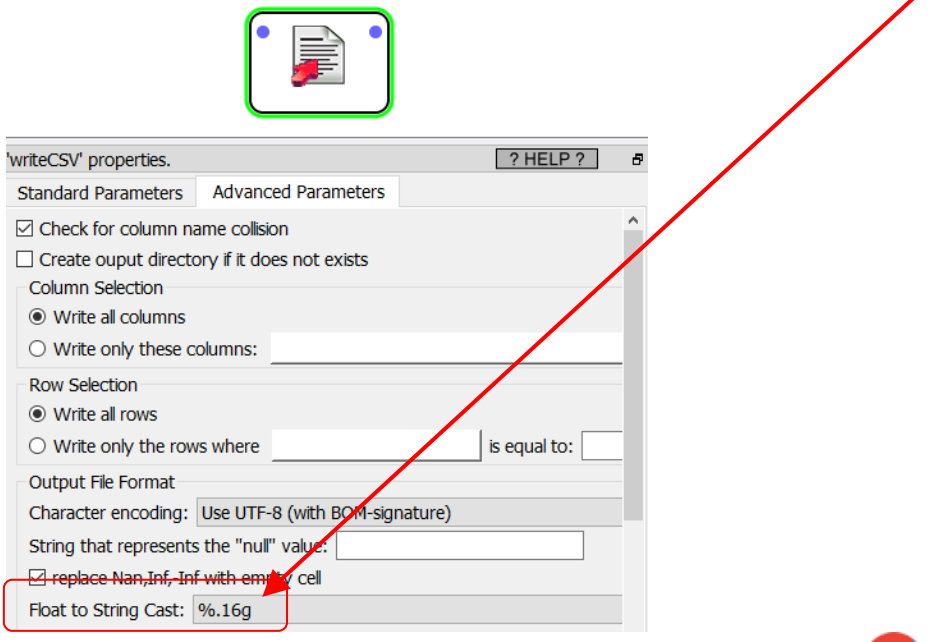
To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

Let's assume that I have some data table on my local server (in a .gel_anatella file) and I now want to play with this data using Google Big Query. To do so, I will follow the following steps:



Here are more details on the 4 steps illustrated in the above Anatella graph:

- **Step 1:** Export your data table to a simple .csv file using the writeCSV action. To avoid any loss of accuracy, please use the "%.16g" option when exporting your table to a .csv file:



- **Step 2:** Copy your .csv file inside your Google Cloud Storage using the GoogleStorageUpload Action detailed in section 5.23.18.

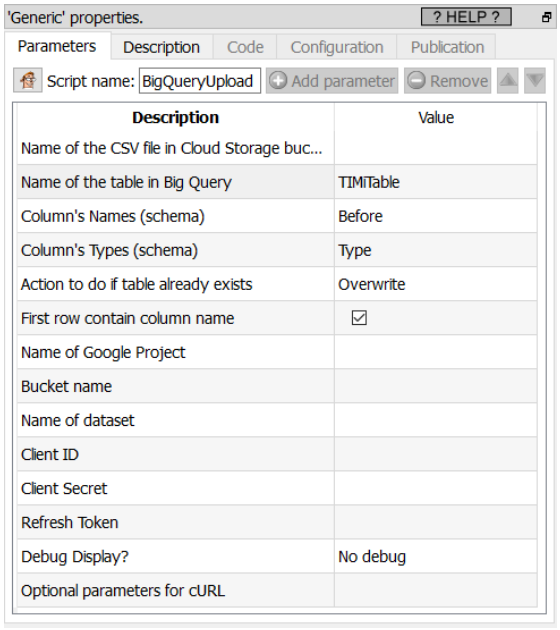
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- **Step 3:** Import your .csv file from your Google Cloud Storage into your Big Query infrastructure using the BigQueryUpload Action detailed in the next section 5.23.22.
- **Step 4:** Run different queries on Google Big Query using the BigQueryDownload Action detailed in this section (5.23.21).

5.23.22. Upload a table to Google BigQuery

Icon:

Property window:



Short description:
Upload a table to Google BigQuery from a Google Cloud Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

The previous section (5.23.21) contains an example of usage.

To be able to use this Action, you need to get these 3 parameters from Google: (1) your "Client ID", (2) your "Client Secret", (3) your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

This action expects as input a table with two columns (by default, these two columns are named "Before" and "Type"):

- The first column (i.e. the column named "Before") contains the column's names of the table to import inside BigQuery.
- The second column (i.e. the column named "Type") contains the column's types of the table to import inside BigQuery: More precisely, we will have:

Content of the column "Type" inside the input table of the BigQueryUpload Action	Type of the column when imported inside Google Big Query (i.e. how the column is declared inside the schema of the imported table inside BigQuery)
K	INTEGER
F	FLOAT
U (i.e. any other value than 'K' or 'F')	STRING

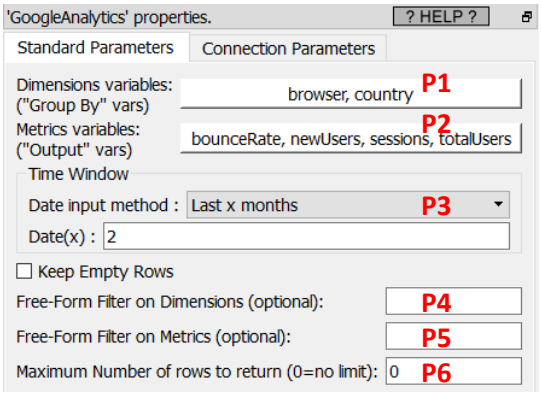
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

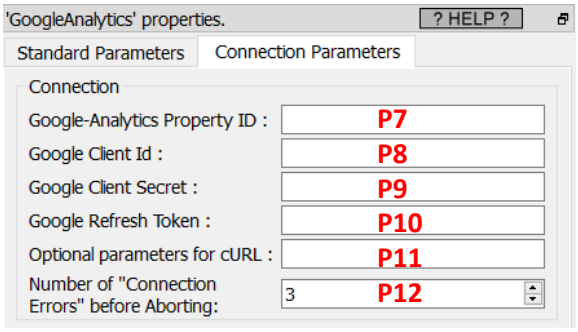
The Get-Meta-Data action (see section 5.5.14) returns the exact table required as input table to the BigQueryUpload Action: See the previous section (5.23.21) for an example of usage.

5.23.23. Google Analytics

Icon:

Property window:





Short description:
Download data from Google Analytics

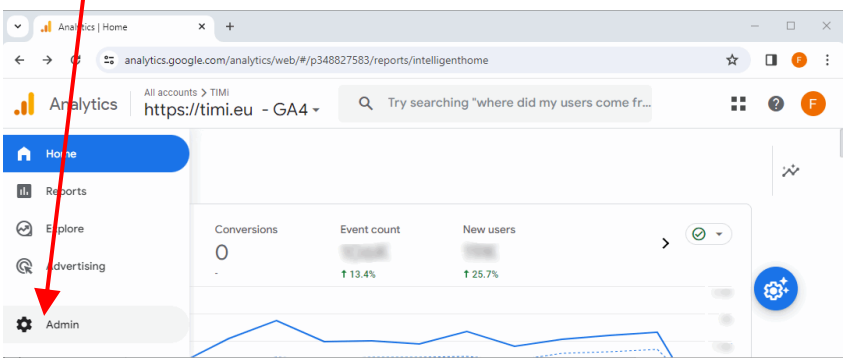
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on this subject.

At the end of the year 2023, Google changed the way to access the data from Analytics. You need Anatella v3.29 or above to access the new Google Analytics v4 API.

To be able to use this Action, you need to get these 3 parameters from Google using the "Unlock Google Services" action detailed in section 5.23.11.:
(1) Parameter **P8**: your "Client ID",
(2) Parameter **P9**: your "Client Secret",
(3) Parameter **P10**: your "Refresh Token".

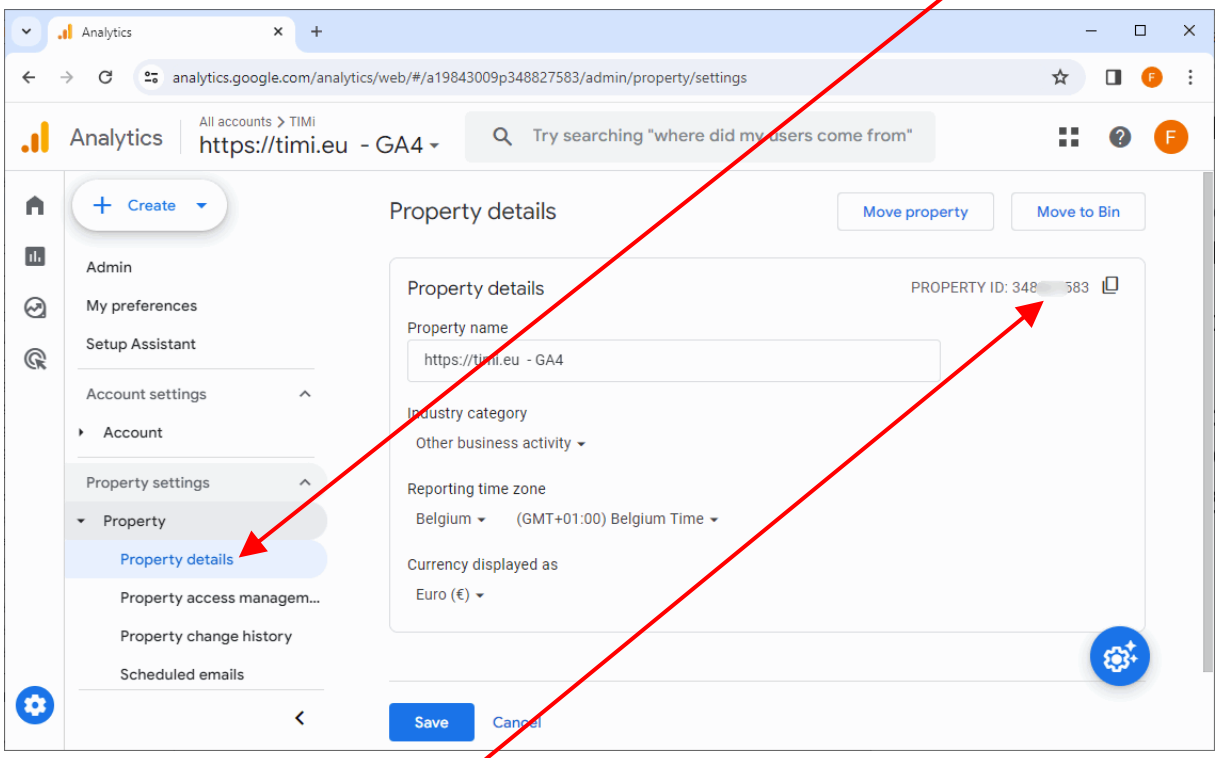
To get your "Google-Analytics Property ID" (Parameter **P7**):

1. Visit Google Analytics at https://analytics.google.com/
2. Select Admin:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Select your Property, Inside the “Property Settings”, select the “Property details”:



4. Your “Property ID” is visible here:

If you get the following error message:

"User does not have sufficient permissions for this profile."

...this means that your "Property ID" is incorrect.

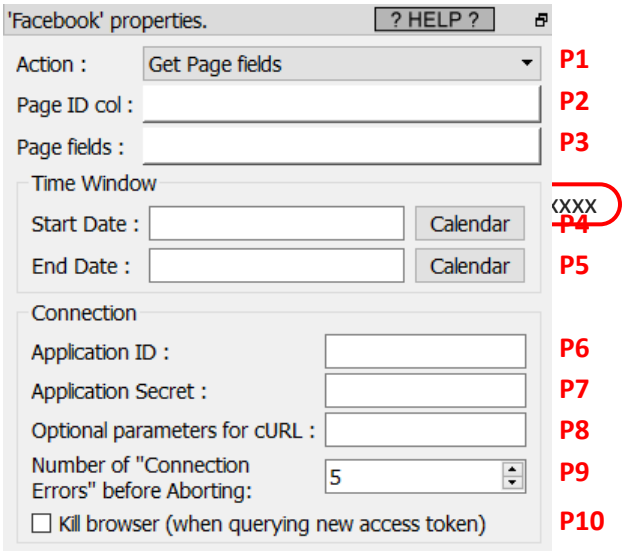
If you created custom, used-defined Metrics/Dimensions inside Google Analytics, you’ll find them directly inside the Metrics/Dimensions windows (warning: their name will start with “custom-…”).

5.23.24. Facebook

Icon:

Property window:

Short description:
Download data from Facebook Pages



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from Facebook:

- your "Application ID" (parameter **P6**) and "Application Secret" (parameter **P7**), so that your application can login into facebook: see the next section (section 5.23.24.1) to get these 2 parameters.
- the "Page ID" (the parameter **P2** is the name of the column of the input table that contains all the required "Page ID's") of all the pages that you want to query with Anatella: see the section 5.23.24.2 on how to get this parameter.

Once you have completed the "setup process" described in the next two sections (i.e. in the section 5.23.24.1 and the section 5.23.24.2), you can use the parameters **P1**, **P3**, **P4**, **P5** to select the data to extract from your Facebook pages. More precisely, you can extract from your Facebook pages all the "Page fields", all the "Page Post fields" and all the "Page Post Comments fields".

After a few months (usually every 3 months), Facebook forces you to re-authenticate yourself before you can get access to any data again (i.e. your "User Access Token" becomes expired). When this happens, Anatella automatically open a browser that will automatically re-new your authentification token (i.e. Anatella automatically gets a fresh "User Access Token"). No worries: The whole procedure is fully automated (unless you deleted your Facebook cookies from your browser, in which case, you need to manually login yourself, using your keyboard): i.e. you'll only see a browser that opens with the word "Success" displayed. You can check the parameter **P10** to automatically **close *ALL*** the running internet browser after the authentification procedure is complete (Warning: This will close all the currently running internet browser!).

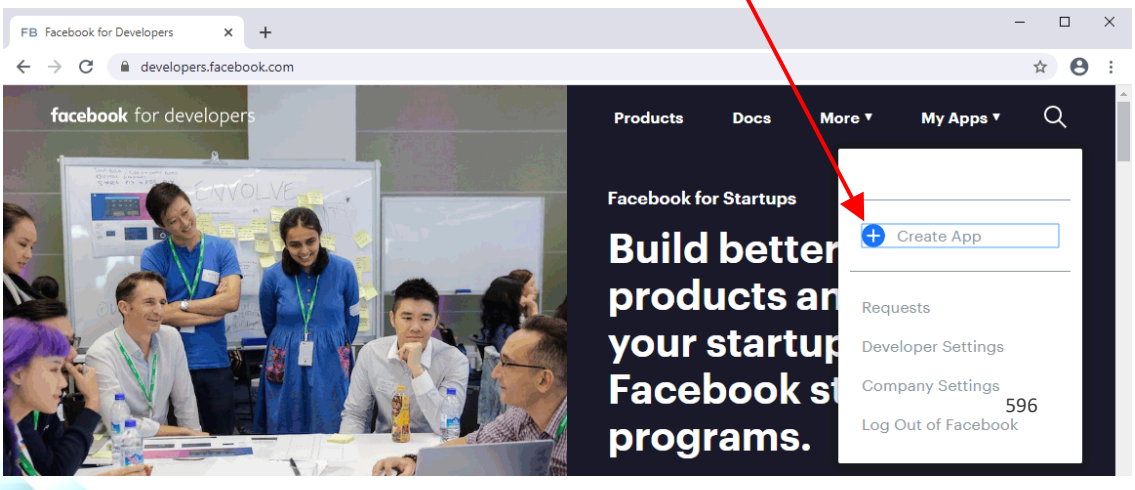
The automated authentification procedure (described in the above paragraph) also happens the first time that you run the Facebook Action.

5.23.24.1. Facebook Action: First time setup

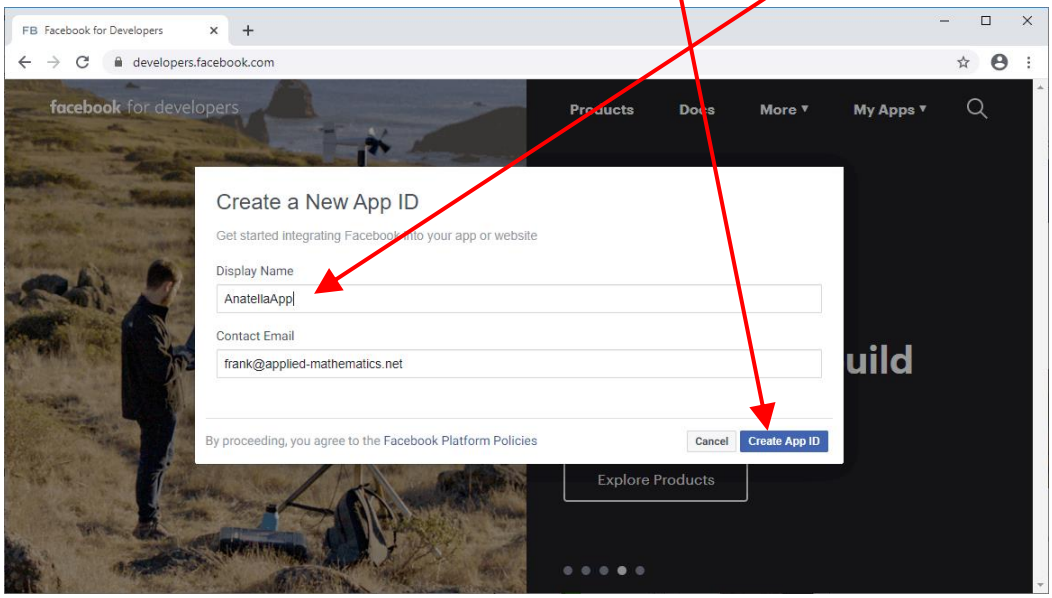
Before getting data from Facebook, you need to create a new Facebook application.
Here are the steps:

1. Open the url https://developers.facebook.com/ inside your browser and manually login into Facebook (if required).

2. Open the "My Apps" menu and click on the "Create App" button:



3. Fill-in the form: Enter a name for your app (you can use any name: e.g. “AnatellaApp”) and give your email. Once finished, click the “Create App ID” button:

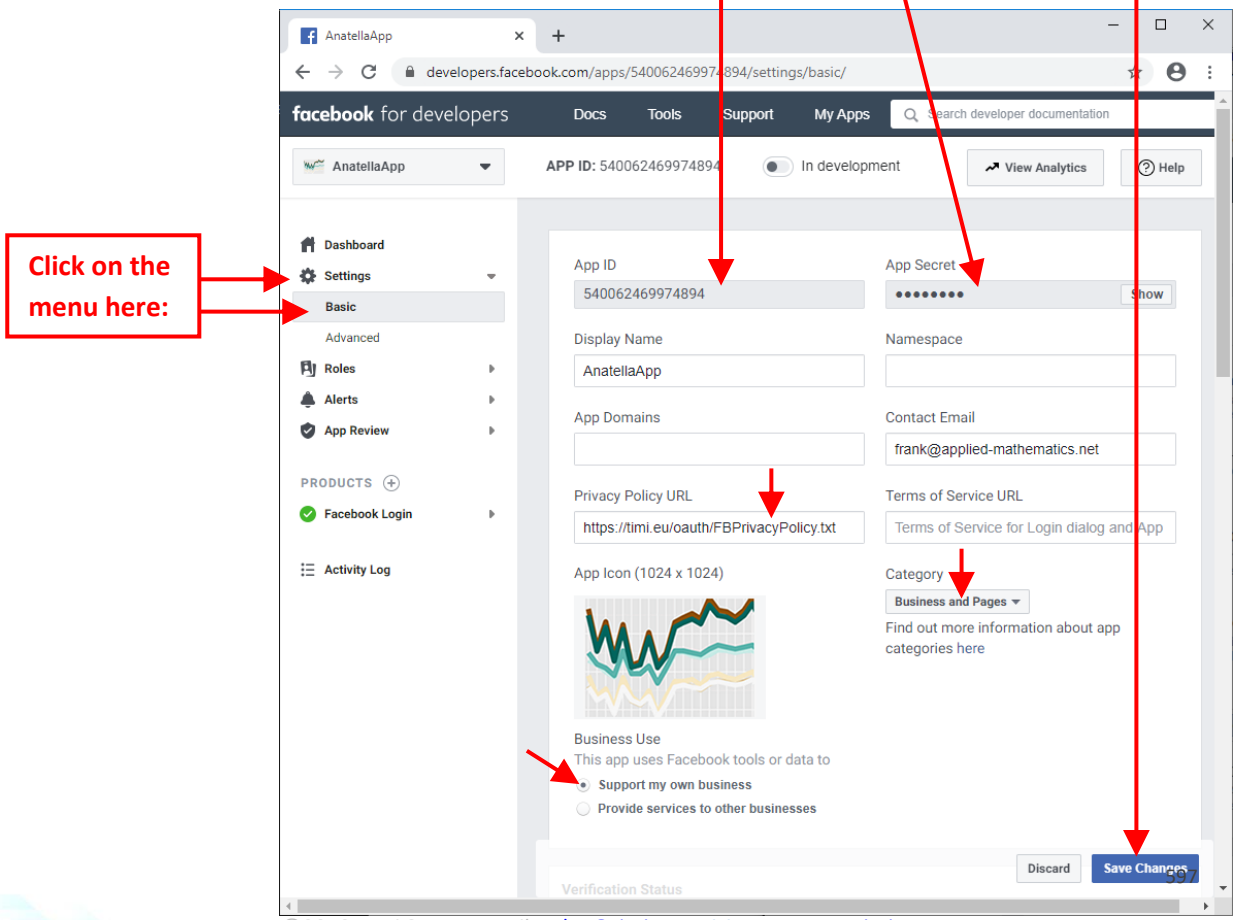


4. In the new webpage, in the left menu, select “Settings”, “Basic” and fill-in the following fields:
   - Privacy Policy URL: you can use: https://timi.eu/oauth/FBPrivacyPolicy.txt
   - Category: select “Business and Pages”
   - Business Use: select “Support my own business”

   Click the “Save changes” button at the bottom right of the screen to save your changes:

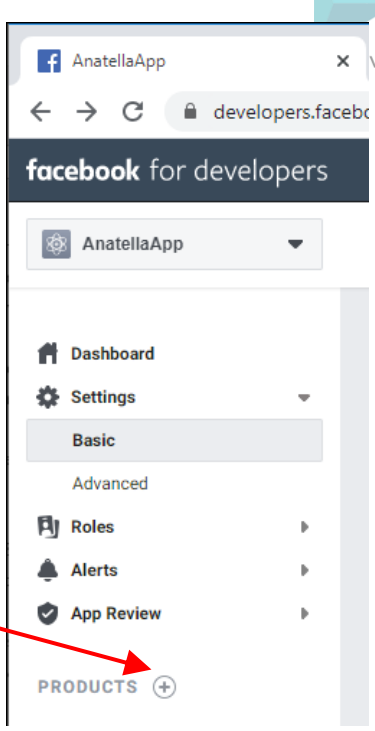
   Please also copy, from the same webpage, your “Application Secret” (parameter **P7**) and your “Application ID” (parameter **P6**).
   You must enter these 2 parameters inside Anatella.

   At the end, you should have something like this:

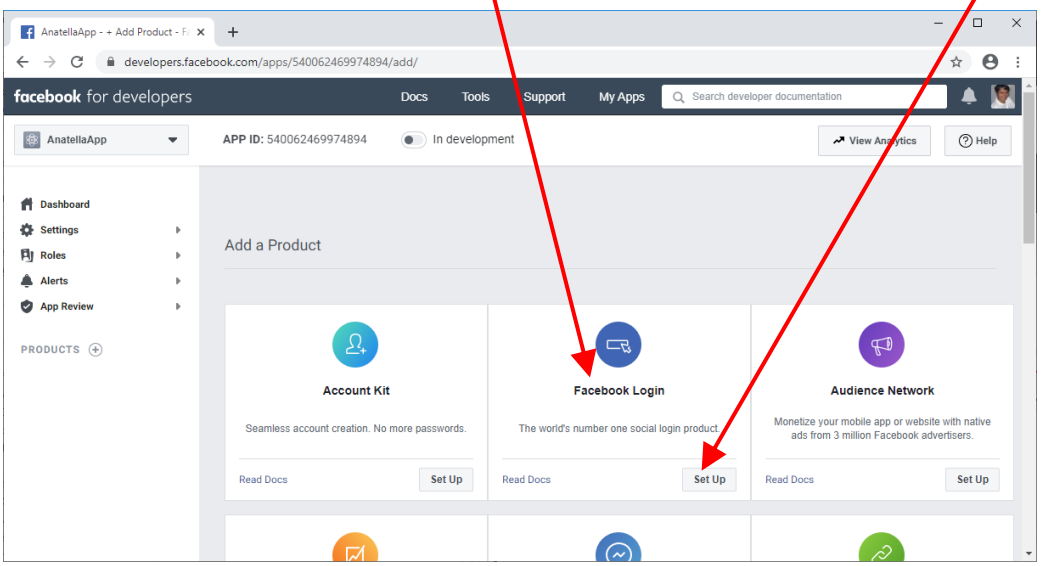


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5. Click the ⊕ icon to open the "Products" page:



6. On the "Product" page, inside the "Facebook login" box, click the "Set Up" button:



7. **This is important!**

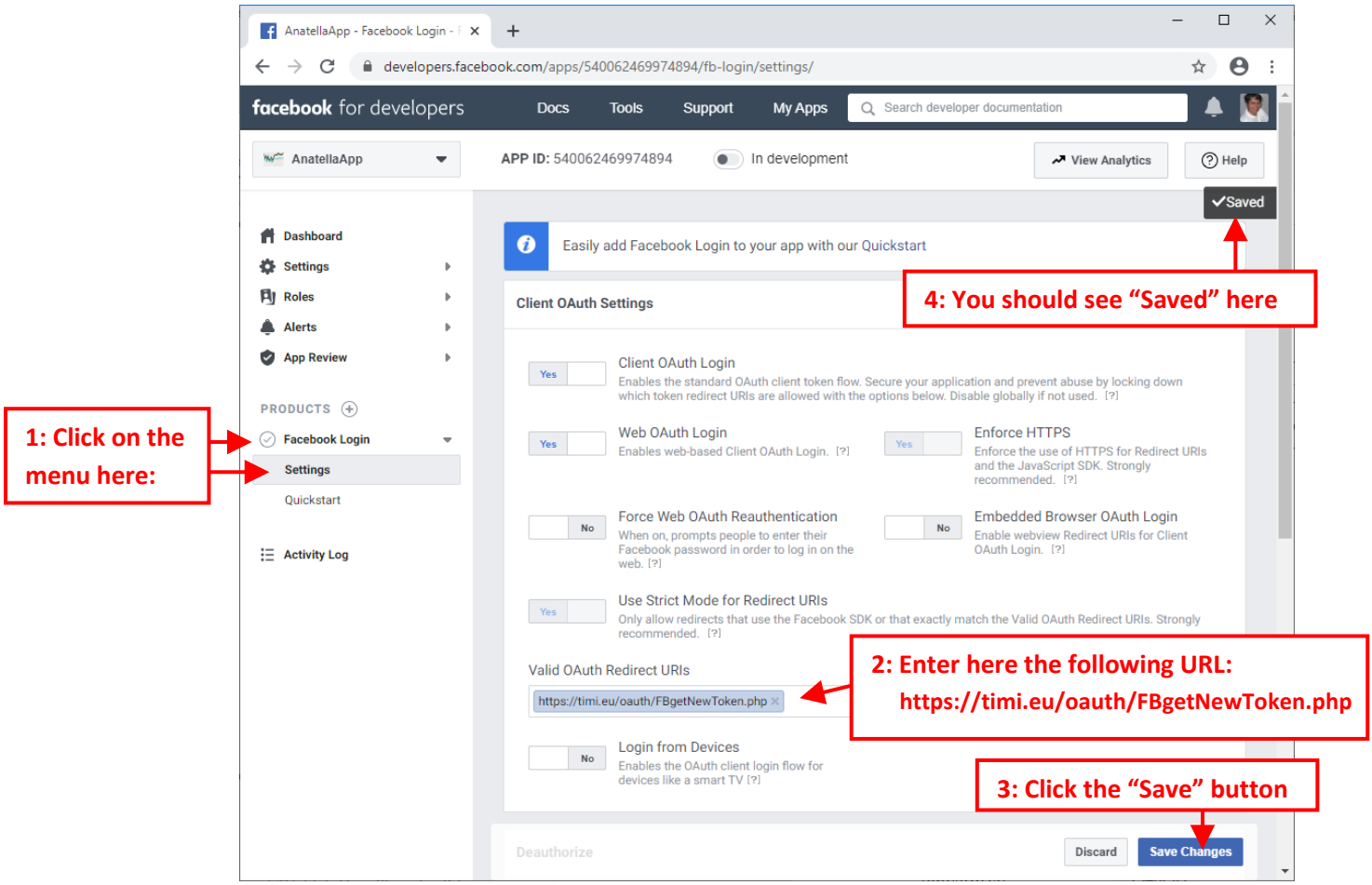
On the menu on the left, click directly the "Settings" button inside the "Facebook Login" menu (see the illustration on the next page). Inside this "Settings" webpage, add inside the parameter named "*Valid OAuth Redirect URIs*" the following URL:

```
https://timi.eu/oauth/FBgetNewToken.php
```

Click the "Save changes" button at the bottom right of the screen to save your changes.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

At the end, you should have something like this:

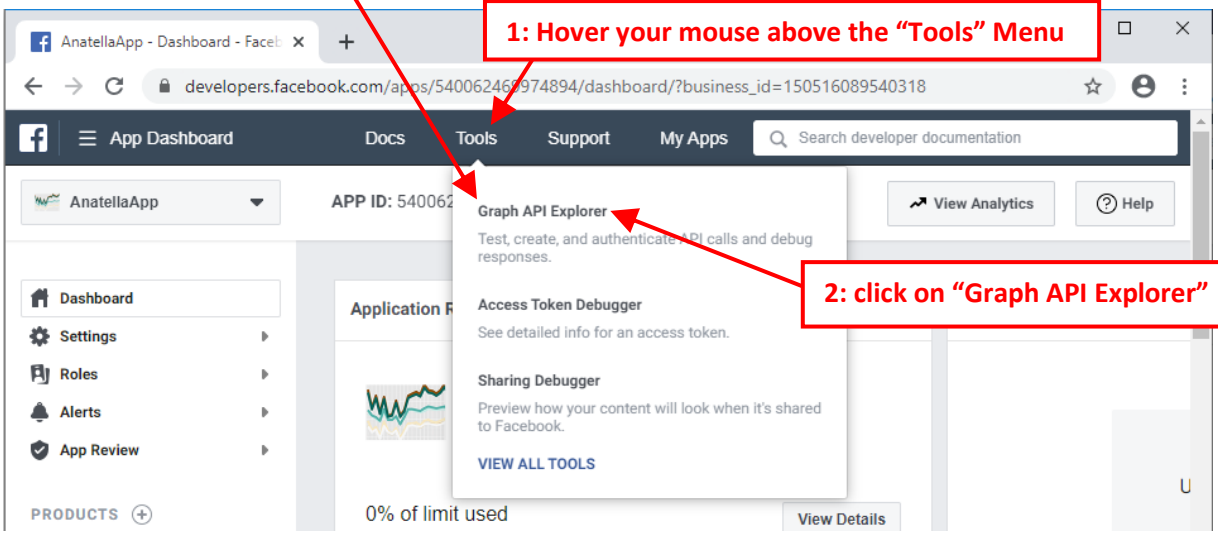


5.23.24.2. How to get the Facebook PageID's?

Here are the steps:

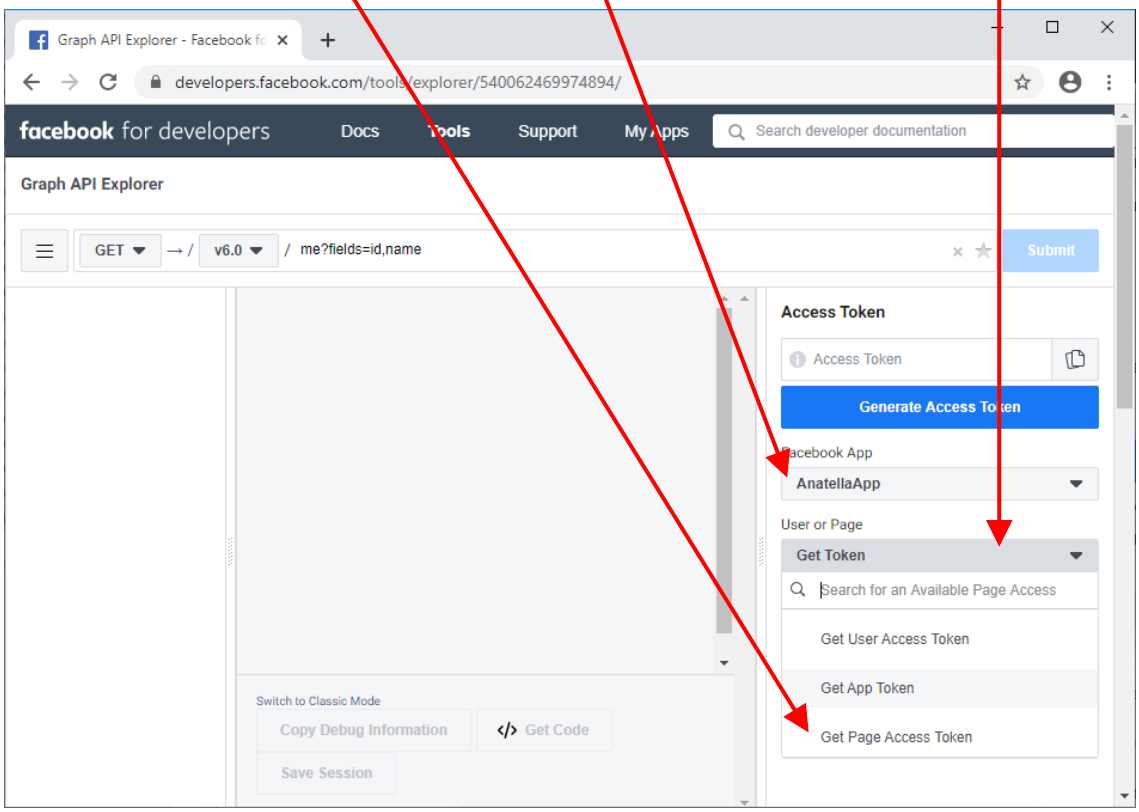
1. You need to grant to your new App the access to all the required pages. To do so:

1.1. Open the Facebook Explorer tool:
Open in your browser this URL: https://developers.facebook.com/tools/explorer
You can also click here:

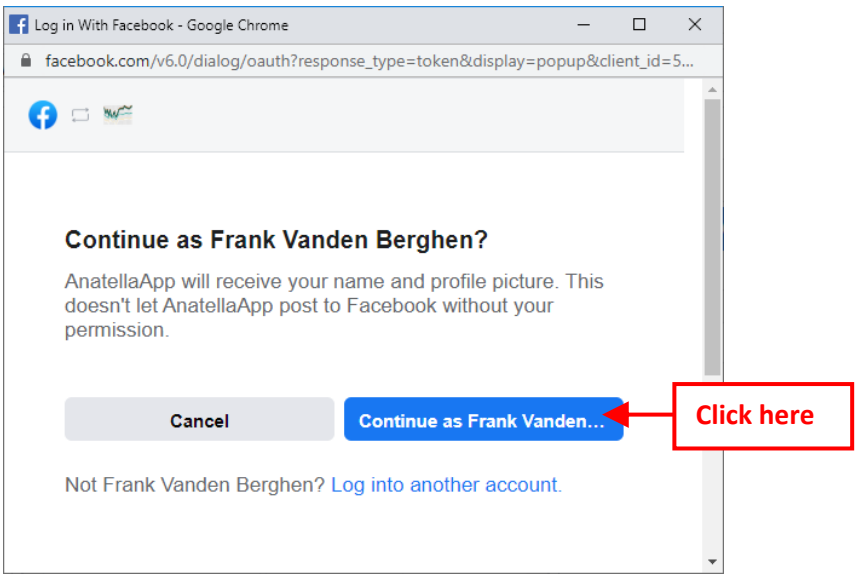


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1.2. Inside the “Facebook Explorer tool”, select your App, open the “User or Page” combobox and select “Get Page Access Token”:

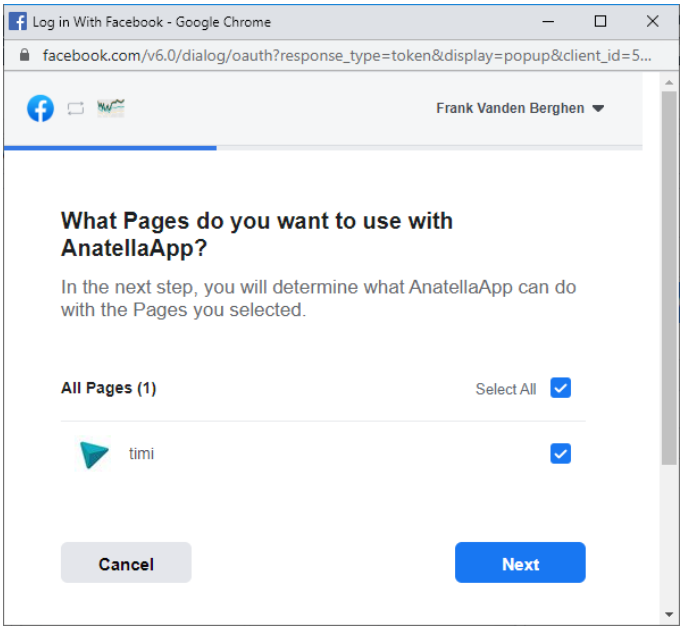


1.3. Confirm your login:

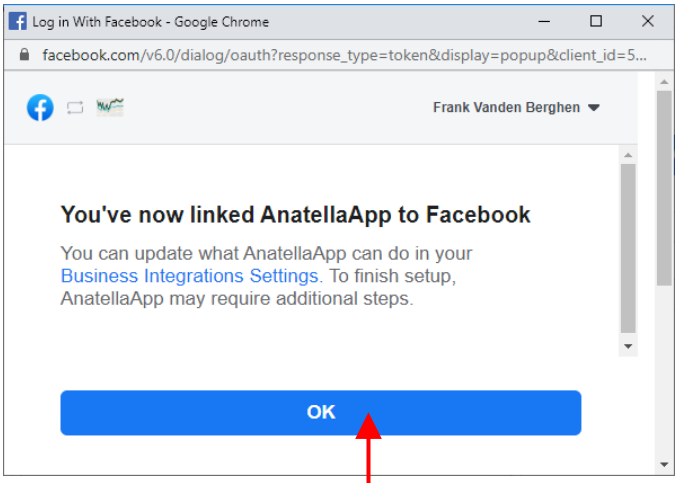


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1.4. Select all the Pages that you want to access using Anatella and click the “Next” button:

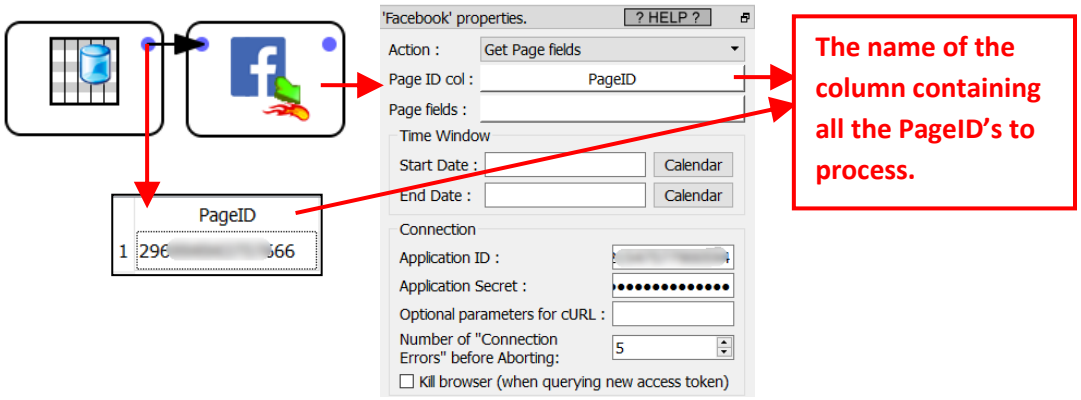


1.5. You should now see this:



1.6. Click the ok button to close this window.

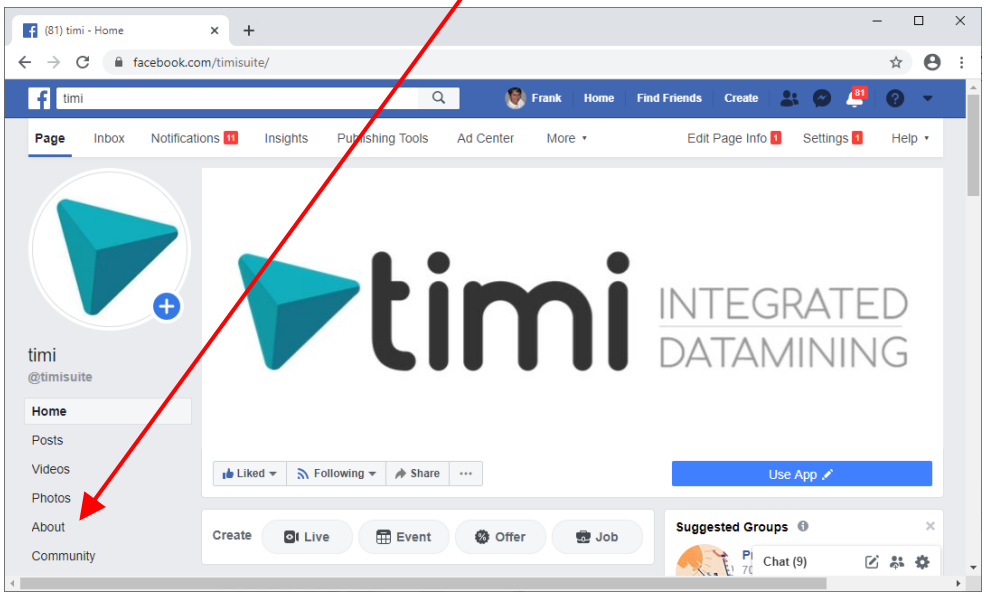
2. The Facebook Action inside Anatella works this way:



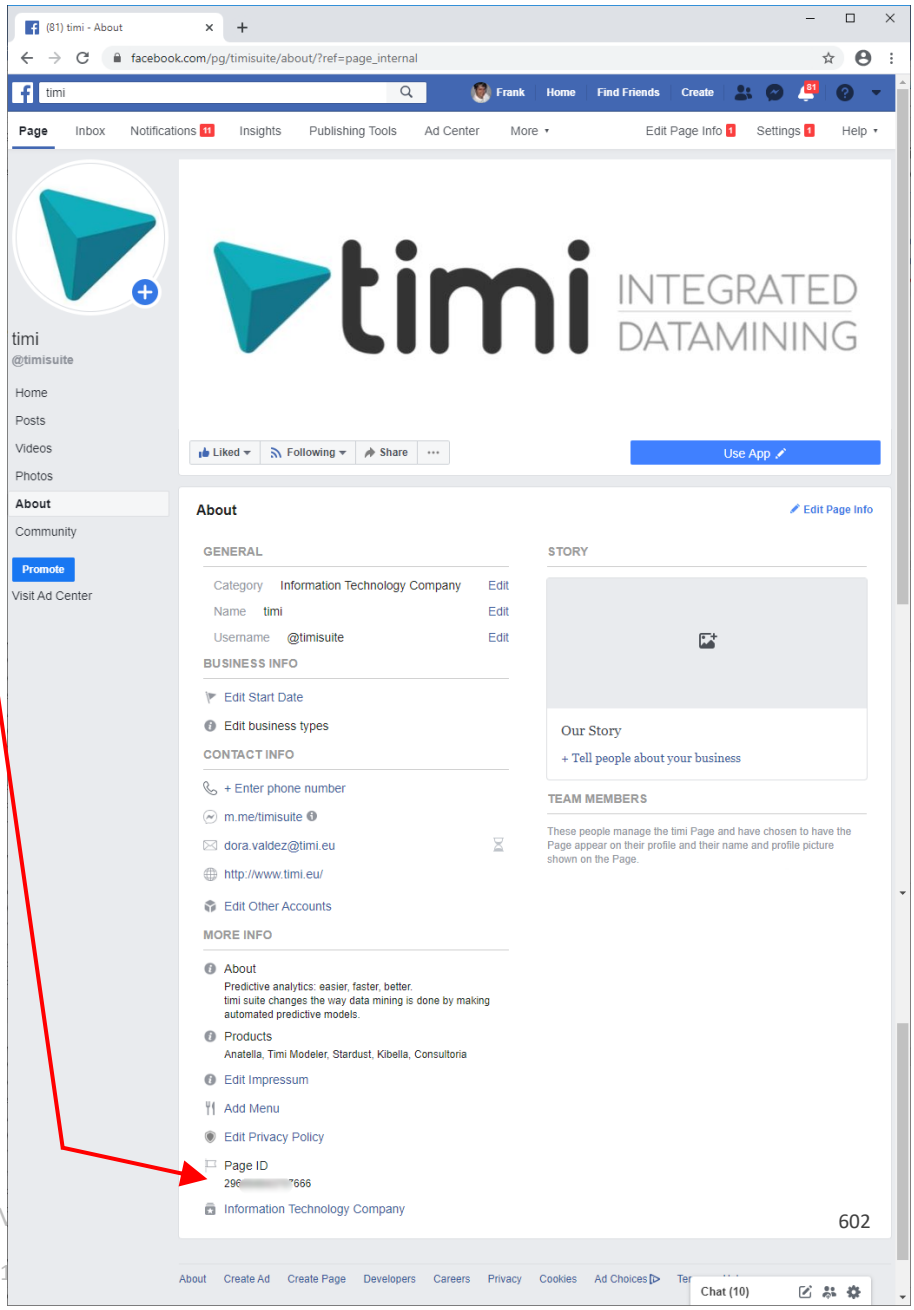
The Facebook Action requires as input a table with the PageID’s of all the Facebook pages that you want to analyze with Anatella. To get the PageID of a Facebook page, open the desired Facebook

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

page in a browser and click the “About” link in the column on the left: For example, for the “TIMi Suite” facebook page, you click the “About” link here:



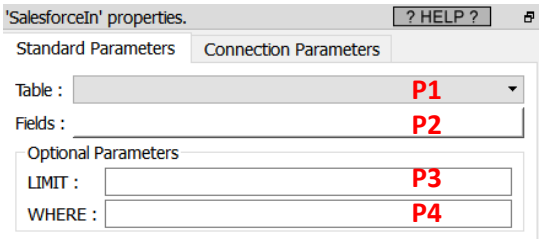
The required “Page ID” (that you need to copy in Anatella) is at the bottom of the “About” WebPage:

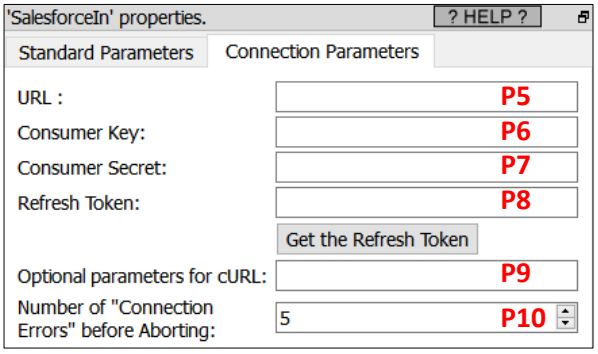


5.23.25. SalesForce Download

Icon:

Property window:





Short description:
Download data from Salesforce

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from Salesforce:

- your "URL" (parameter **P5**)
- your "Consumer Key" (parameter **P6**)
- your "Consumer Secret" (parameter **P7**)
- your "Refresh Token" (parameter **P8**)

See the next section (section 5.23.25.1) on how to get these 3 parameters.

Once you have completed the "setup process" described in the section (i.e. in the section 5.23.25.1), you can use the parameters **P1**, **P2**, **P3**, **P4** to select the data to extract from your Salesforce system.

The parameter **P3** is a number that defines how many rows of data will be downloaded from Salesforce (this is mostly usefull only for testing, the first time that you connect to Salesforce).

The parameter **P4** is the WHERE Clause of the SOQL statement. For more details about SOQL statements, see here:
https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_sosl_intro.htm

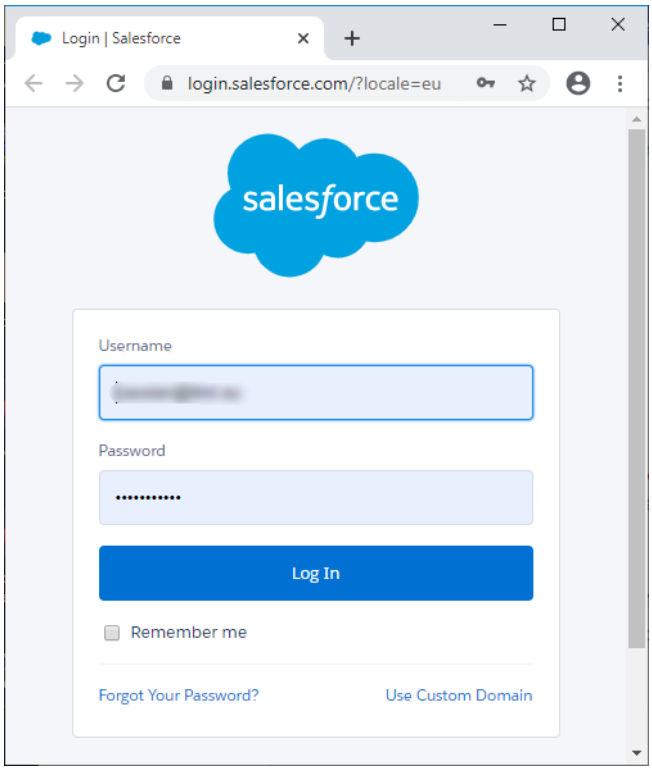
If the parameter **P4** is left empty, Anatella downloads all the records (i.e. there is no filter).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

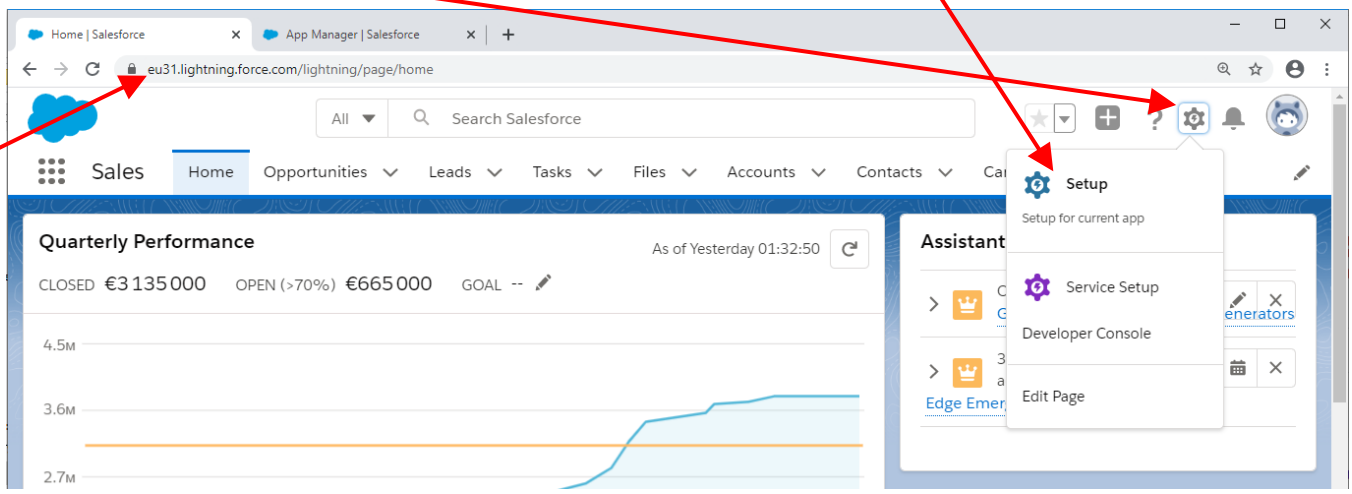
5.23.25.1. Salesforce Download/Upload: First time setup

Before downloading data from Salesforce or before uploading data to Salesforce, you need to create a new Salesforce application. Here are the steps:

1. Open the URL https://login.salesforce.com/?locale=eu and log-in into your Salesforce account:



2. Click the gear icon and click "Setup" in the drop-down menu:

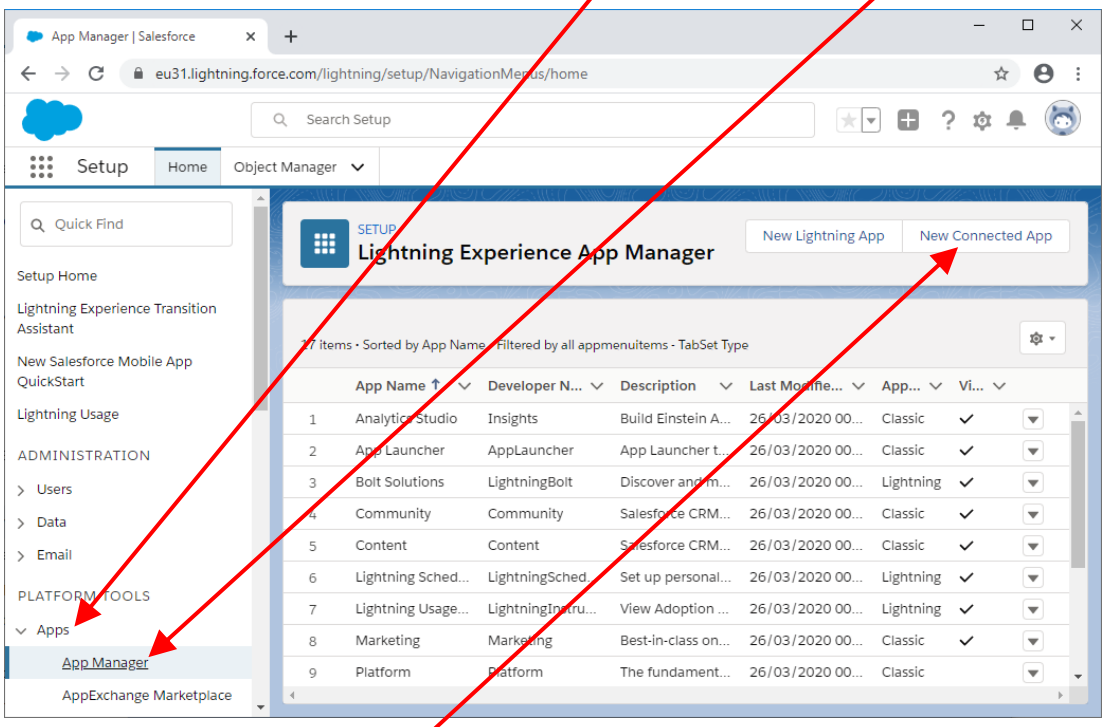


3. The Anatella parameter **P5** (i.e. the "URL") is based on the first characters of the current salesforce URL. For example, in the screenshot here above, the current salesforce URL starts with "https://eu31…", so the Anatella parameter **P5** is: https://eu31.salesforce.com

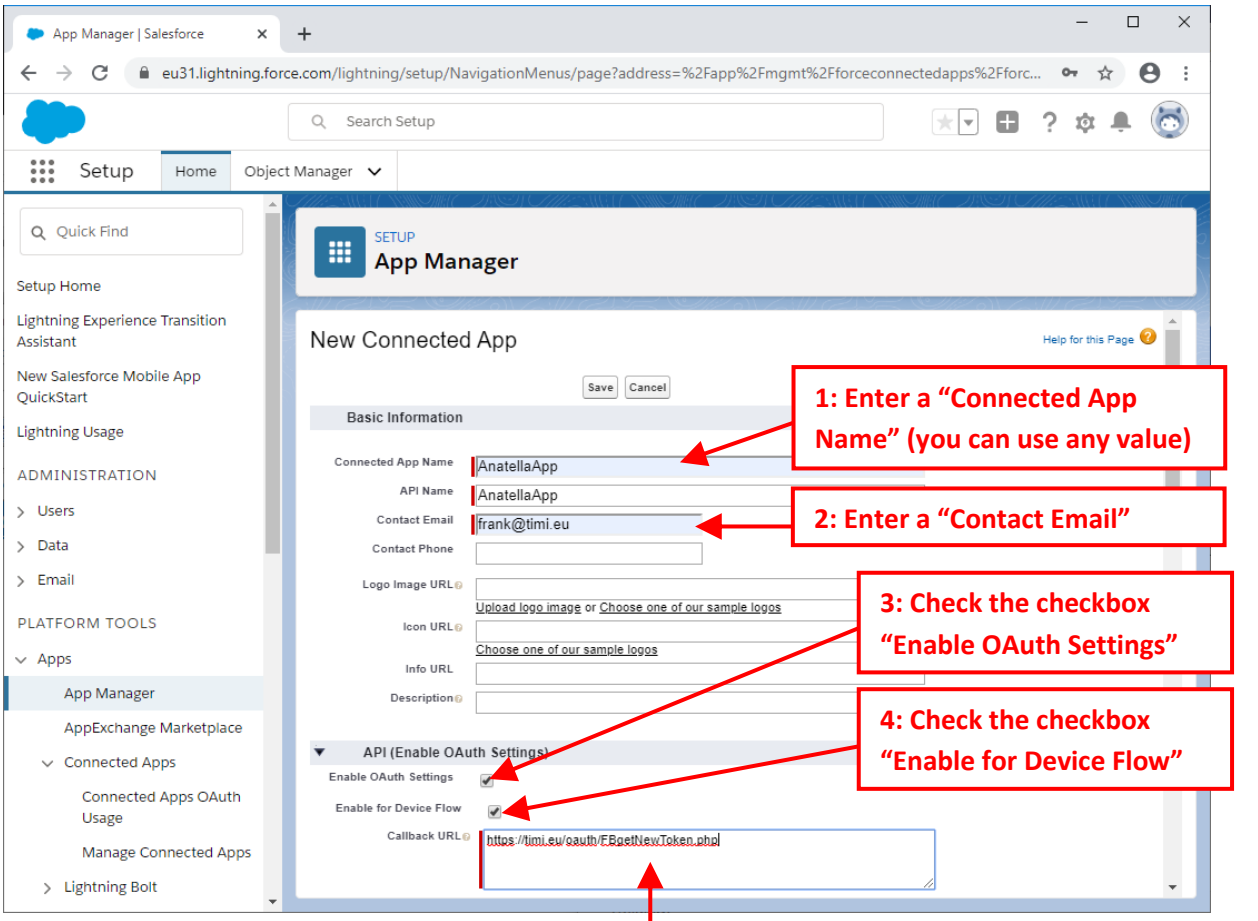
   IMPORTANT: The parameter **P5** ends with "salesforce.com" and not with "lightning.force.com".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. In the left column, in the “PLATFORM TOOLS” section, click “Apps” and then click “App Manager”.



5. Click the “New Connected App” button:
6. Fill-in the required fields inside the “New Connected App” setup page:



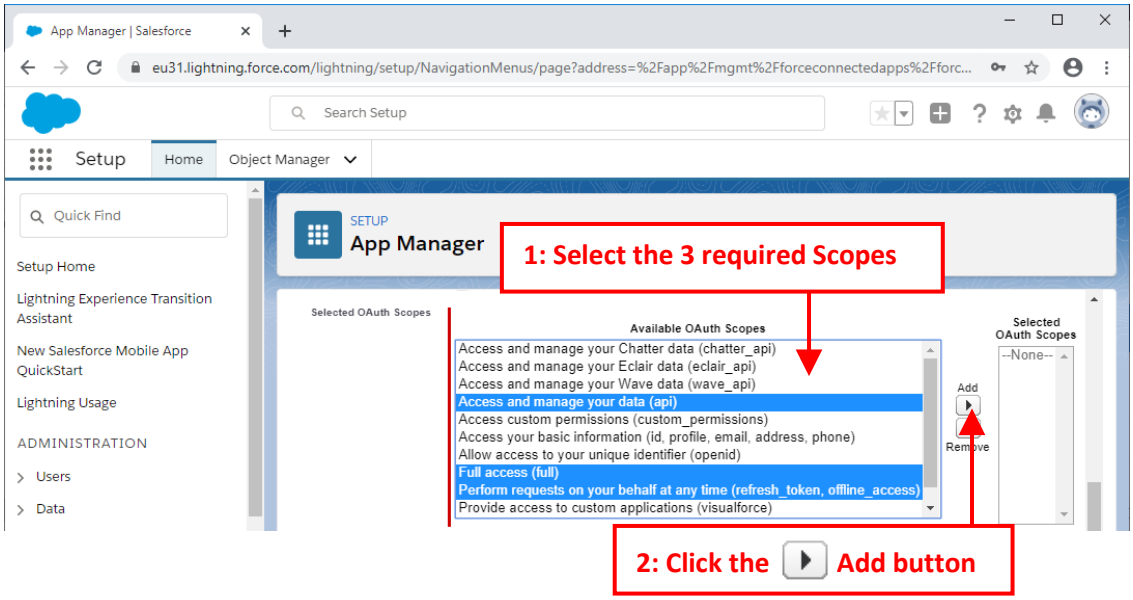
©2010 TIMI SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

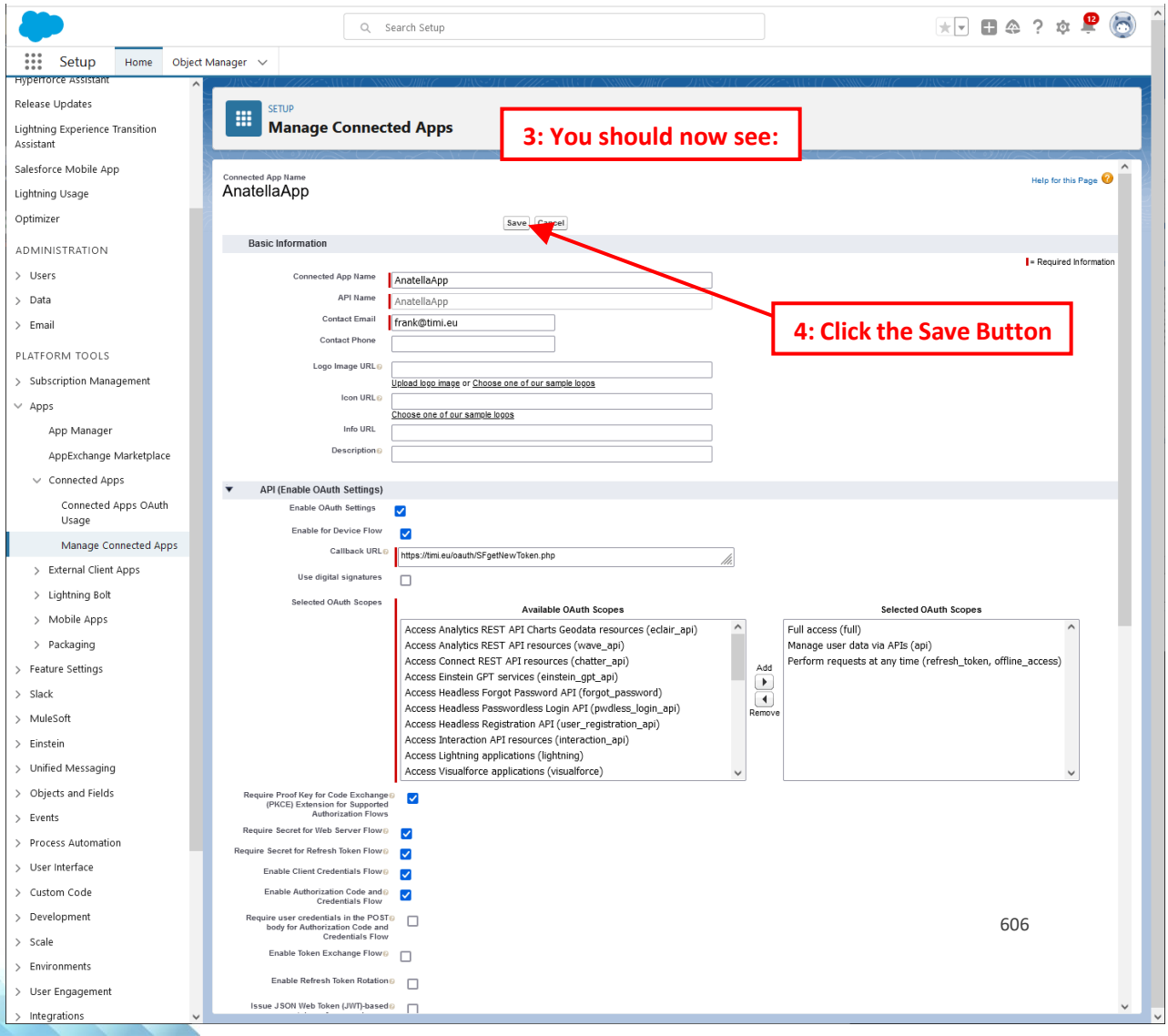
IMPORTANT: For the "Callback URL" parameter, you must use (this is case sensitive):

https://timi.eu/oauth/SFgetNewToken.php

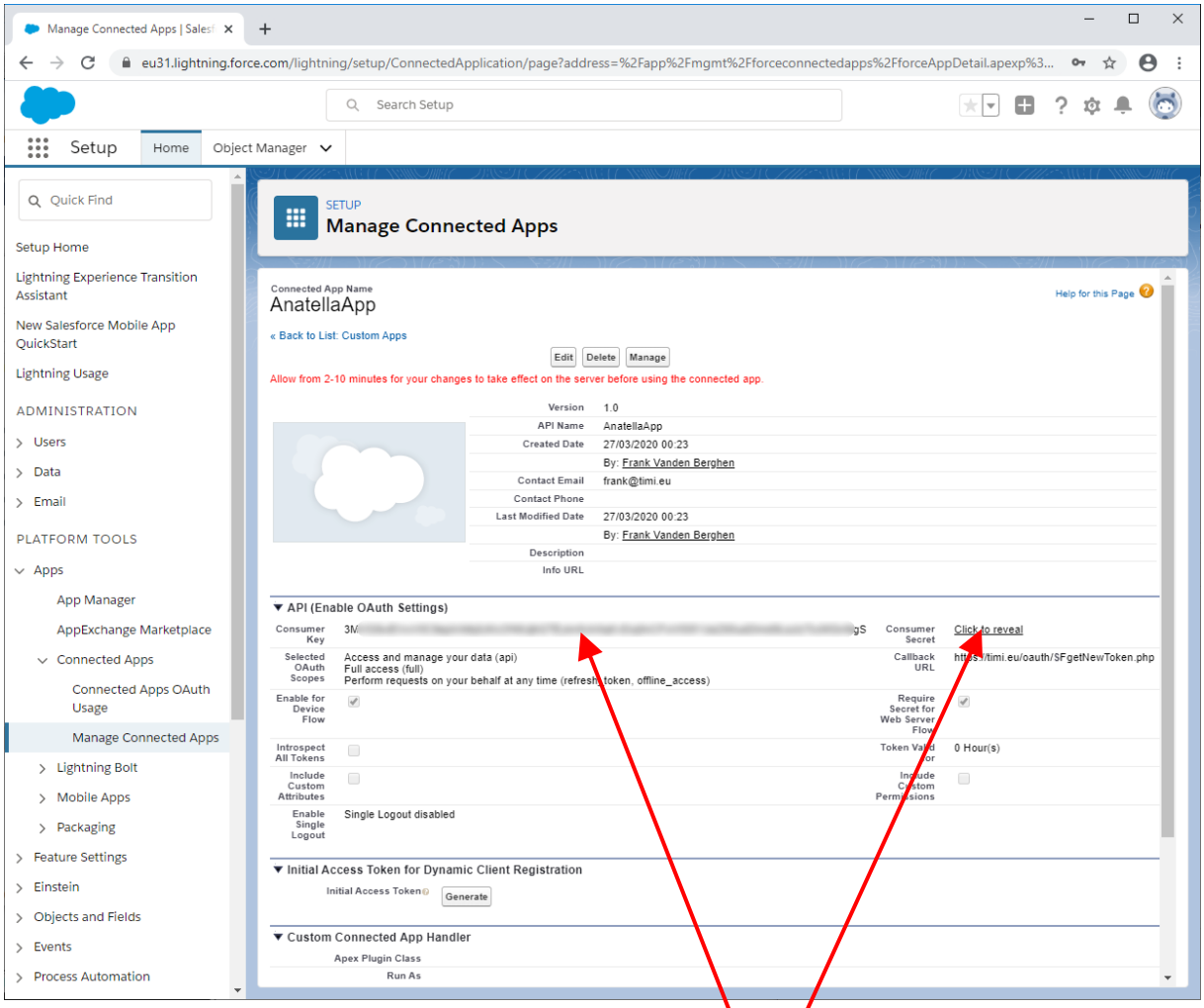
If you miss this step, you'll get later this error "redirect url must match".

7. Continue to fill-in the required fields inside the "New Connected App" configuration page:
   For the parameter "Selected OAuth Scopes", you must select at least these 3 scopes:
   - Full access (full)
   - Manage user data via APIs (api)
   - Perform requests at any time (refresh token, offline access)



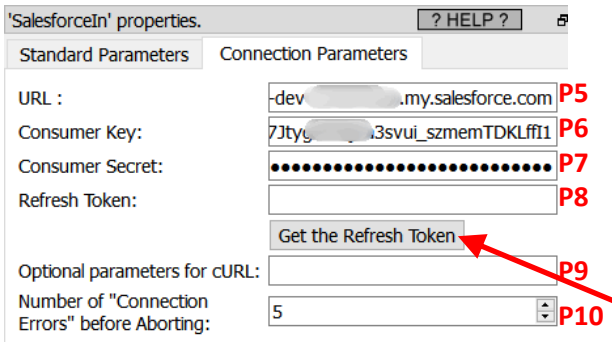


8. After clicking the “Save” button (at the bottom or at the top of the page), you should see this summary screen:



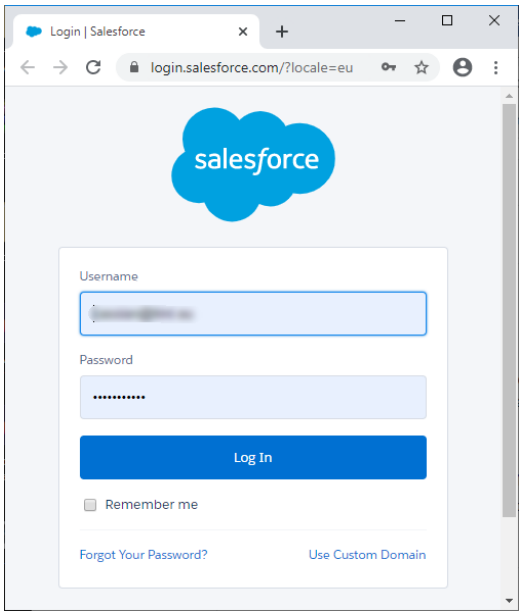
The Anatella-parameter **P6** (i.e. your “Consumer Key”) is here:
The Anatella-parameter **P7** (i.e. your “Consumer Secret”) is here:
Please, copy-paste these 2 parameters into Anatella.

9. Open Anatella. The parameters **P5**, **P6**, **P7** should now be filled-in:



Let’s now get parameter **P8**! Click on the button “Get the Refresh Token” : Your browser opens and shows you a Salesforce Login page:

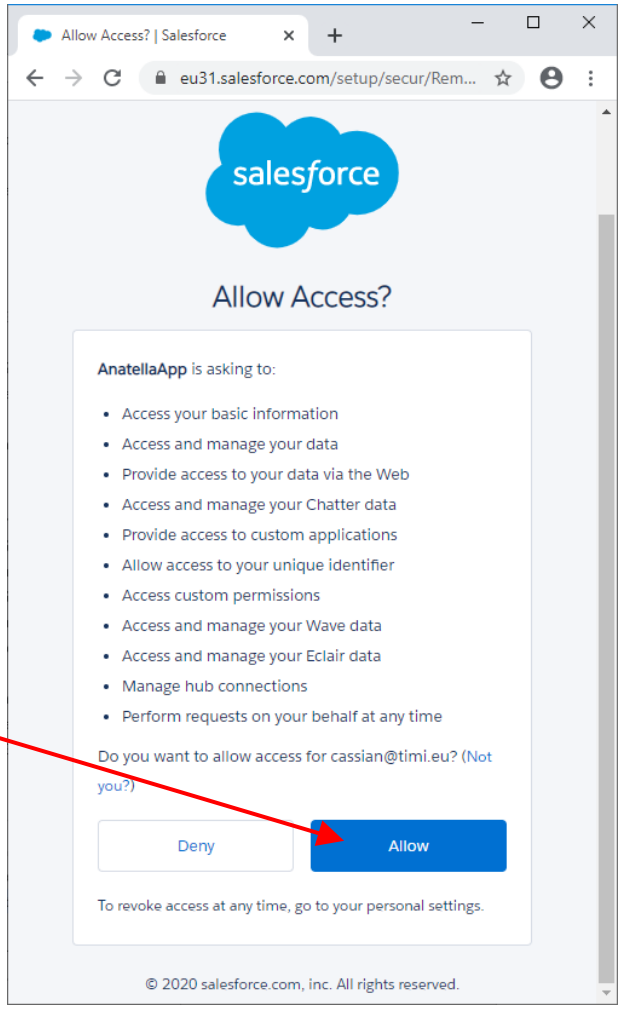
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



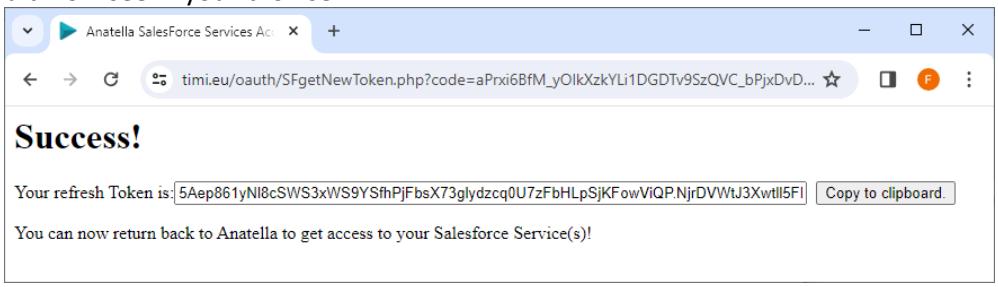
Please, log-in into Salesforce.

If you don't see the Salesforce login page: No Worries! Just read the next section (5.23.23.2), for different ways of troubleshooting the connection.

10. Click the "Allow" button:



11. You should now see in your browser:



Copy/paste your new Refresh Token inside the parameter **P8** From Anatella: i.e. click the button "Copy to clipboard" in the browser, select the parameter **P8** in Anatella and press CTRL+V.

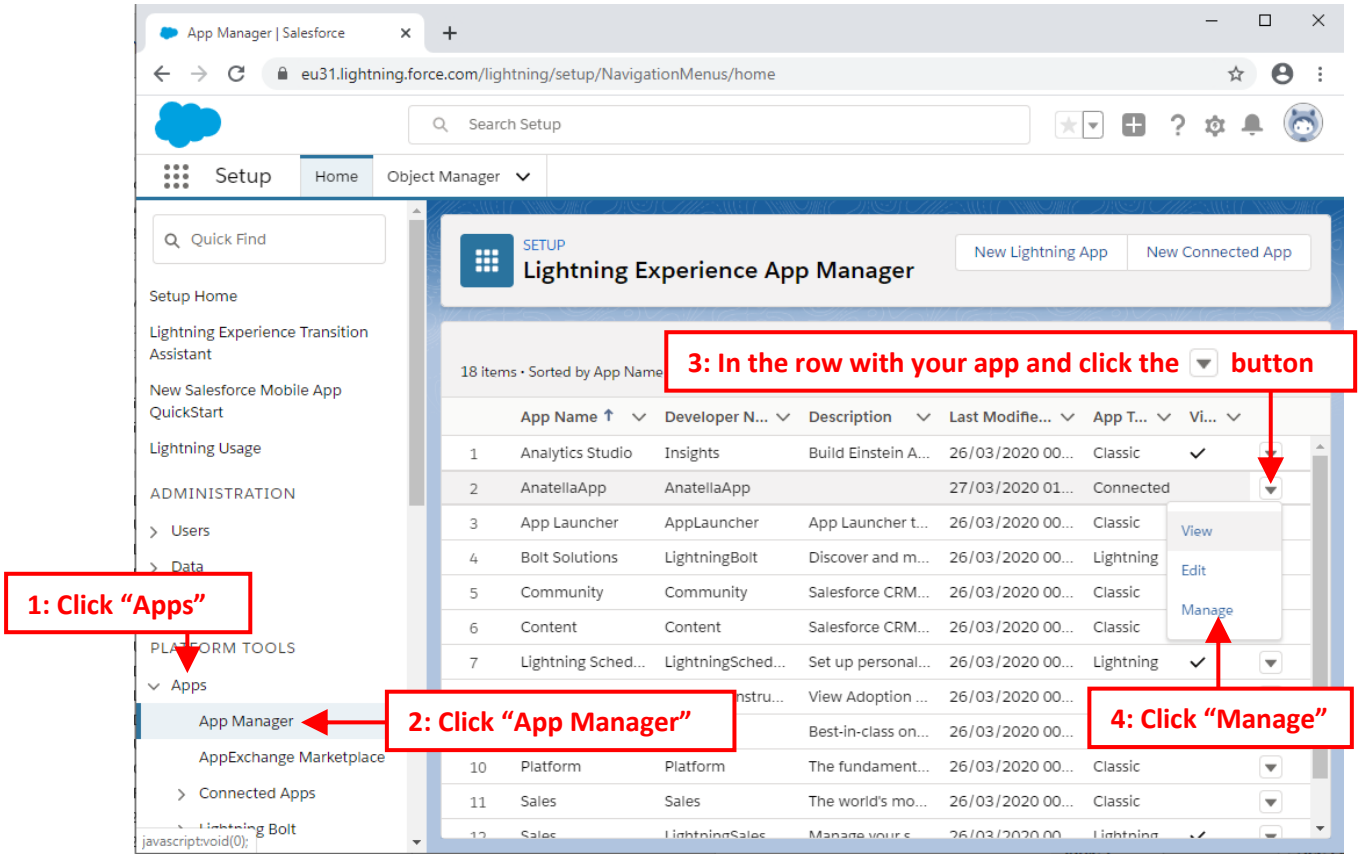
If you don't see "Success" in your browser, it most certainly means that your parameter **P5** (i.e. the "URL") is wrong: Please refer to the step 3 to get the correct parameter **P5**.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.25.2. Troubleshooting the Salesforce connection

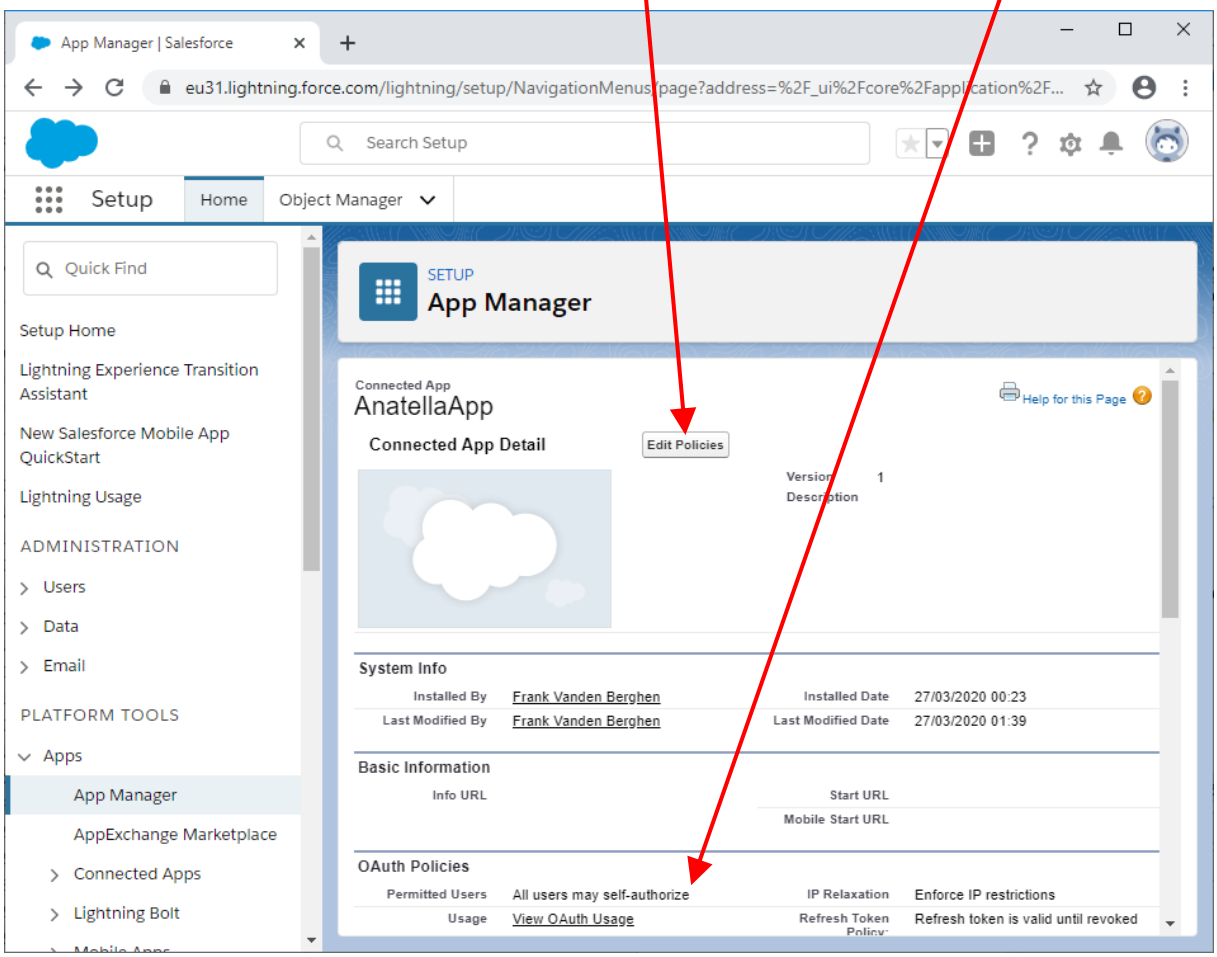
If you cannot see the Salesforce login page at step 10 of the previous section, it usually means that you need to follow this procedure:

1. Open the "Manage Connected Apps" page for your new Application:

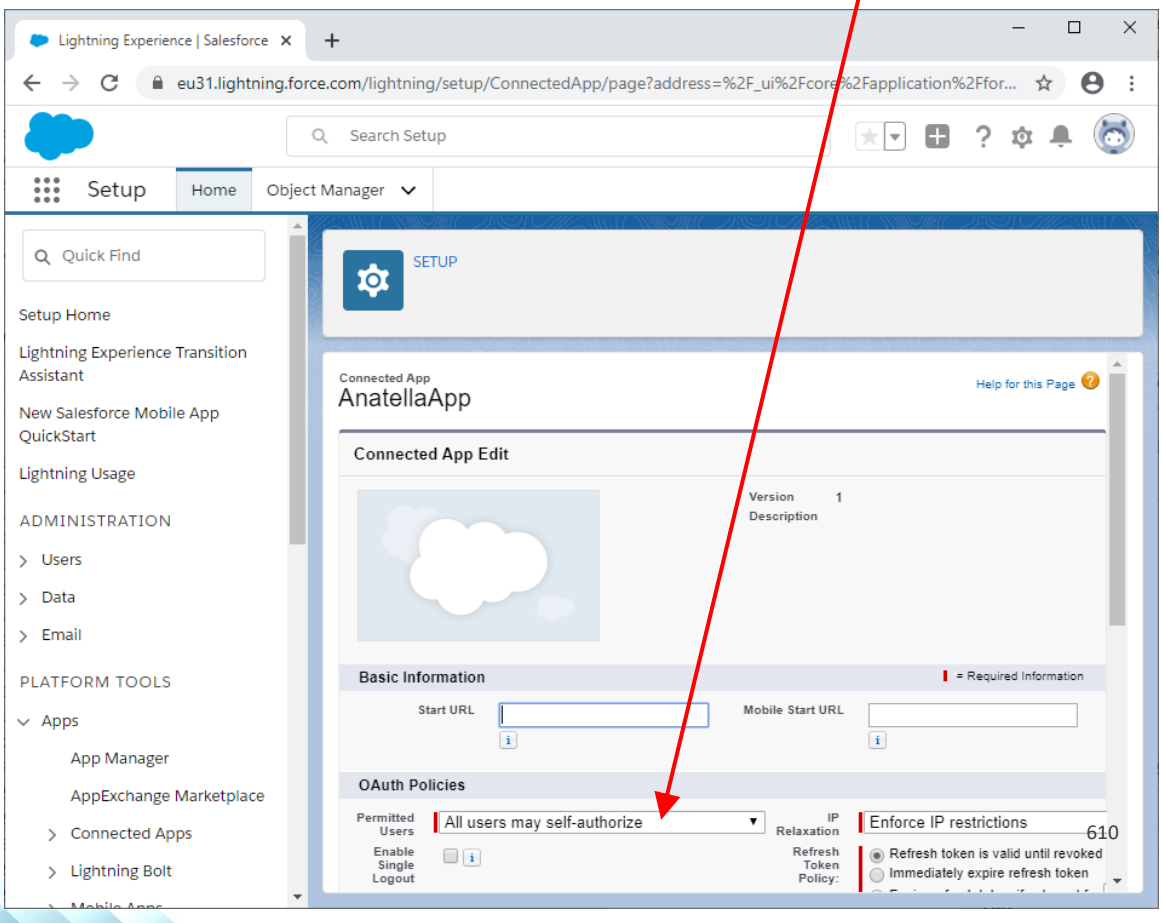


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Make sure that the “Permitted Users” option is set to “All users may self-authorize”: If that’s not the case, click the “Edit Policies” button:

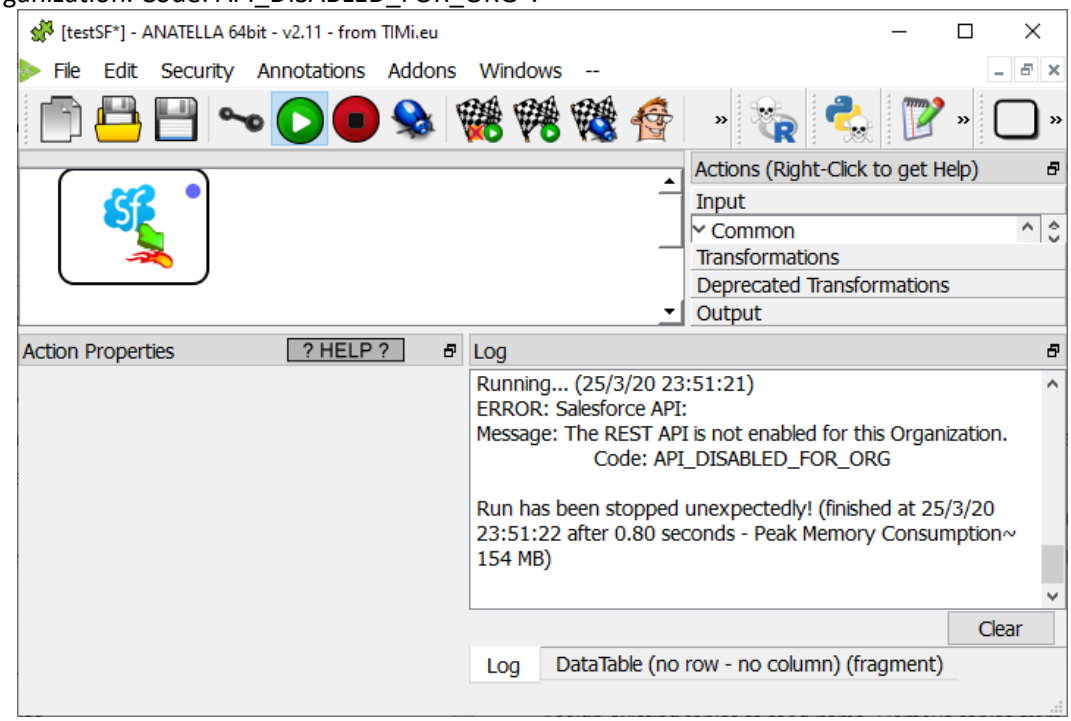


3. Change the “Permitted Users” option to “All users may self-authorize”:



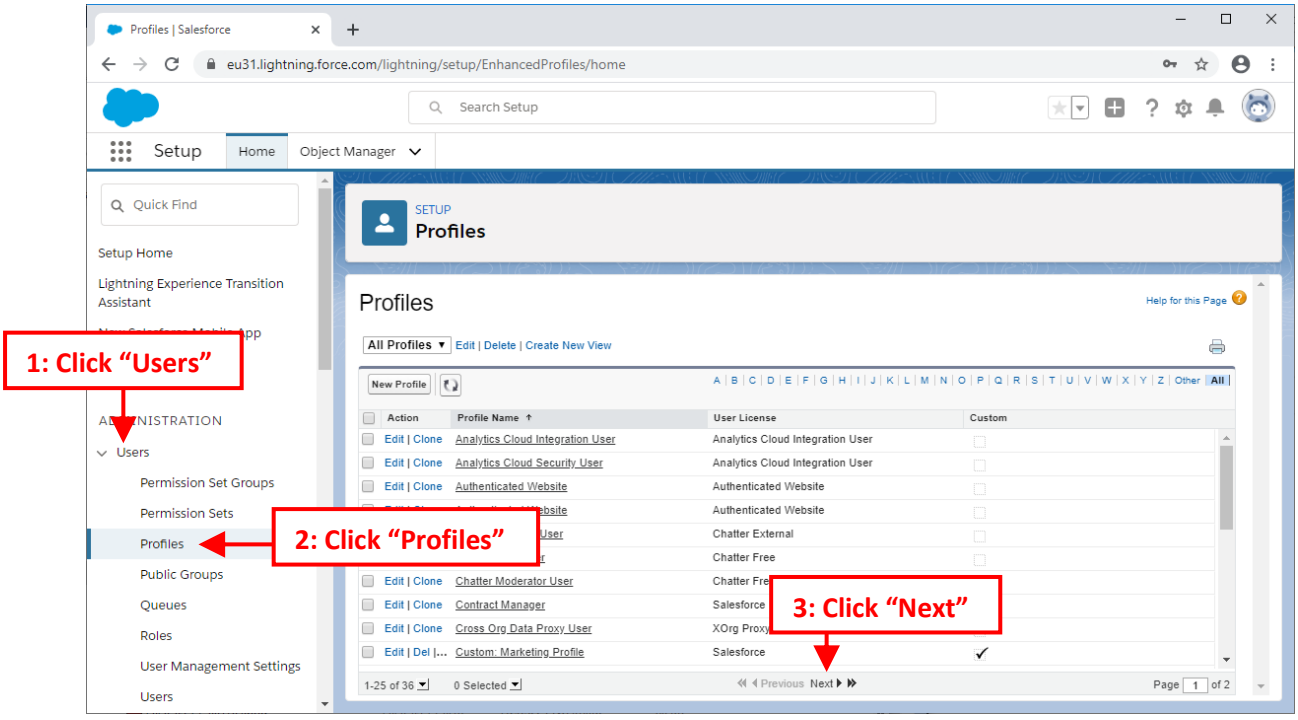
4. Click the “save” button at the end of the page.

If you see inside the Anatella log window the following error message “The REST API is not enabled for this Organization. Code: API_DISABLED_FOR_ORG”:

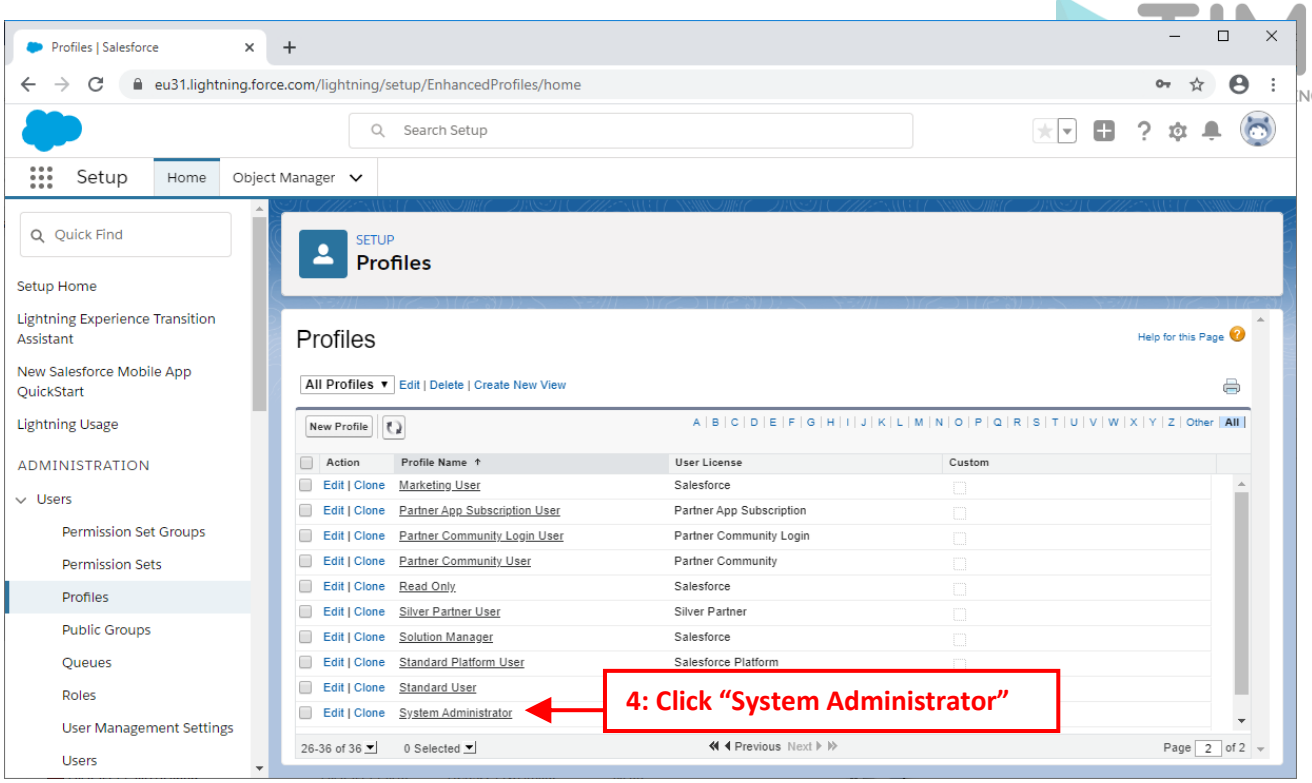


...this usually means that you are still using the TRIAL version of Salesforce. You need the paid version of Salesforce to be able to use the Saleforces Action inside Anatella. If you are using the paid version of Salesforce and you still get this error message, then you’ll need to enable the “REST API” for your organization: Here is the procedure:

1. Open the page to edit the “System Administrator” profile:



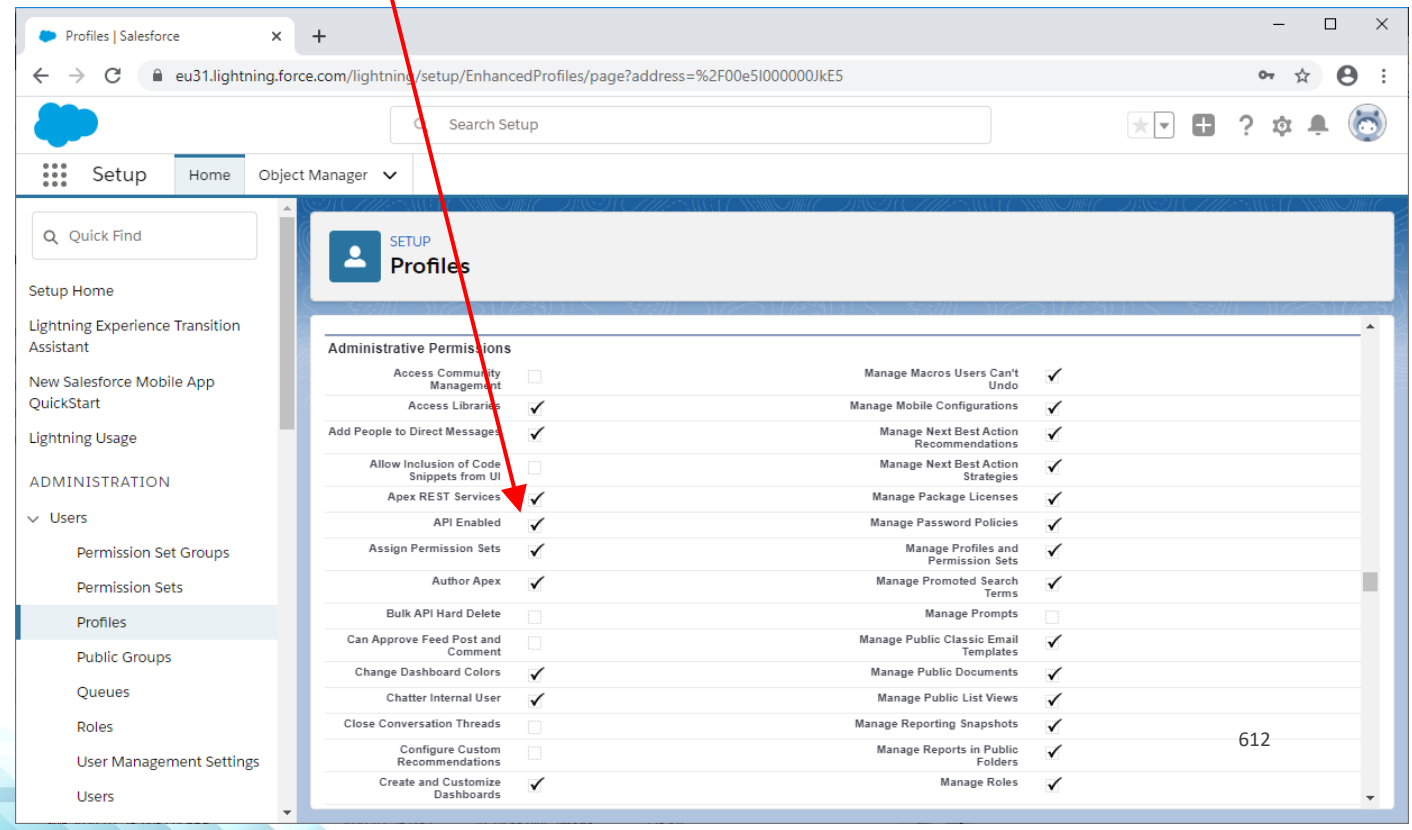
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



2. The page to edit the "System Administrator" profile contains many sections:

   Profile Detail
   Console Settings
   Page Layouts
   Field-Level Security
   Custom App Settings
   Connected App Access
   Tab Settings
   Record Type Settings
   Administrative Permissions
   General User Permissions
   Standard Object Permissions
   Desktop Integration Clients
   Session Settings
   Password Policies

   Scroll down to the "Administrative Permissions" section and verify that the "API Enabled" option is checked:

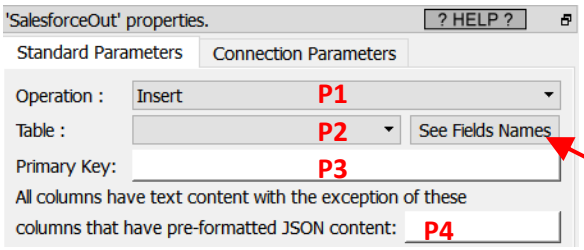


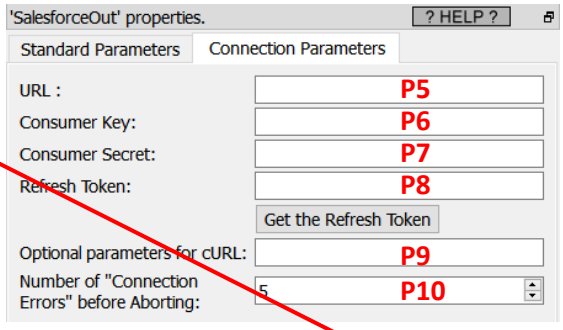
3. If the "API Enabled" option is unchecked, click the "Edit" button at the top of the page, change "API Enabled" option (i.e. enable it) and click the "Save" button.

5.23.26. SalesForce Upload

Icon:

Property window:





Short description:
Upload data to Salesforce

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from Salesforce:

- your "URL" (parameter **P5**)
- your "Consumer Key" (parameter **P6**)
- your "Consumer Secret" (parameter **P7**),
- your "Refresh Token" (parameter **P8**)

See the section 5.23.25.1. on how to get these 4 parameters.

Once you have completed the "setup process" described in the 5.23.25.1, you can use the parameters **P1** to **P4** to upload data to your Salesforce system.

The parameter **P1** selects the operating mode amongst 3 choices:

- Insert data
- Update data
- Delete data

When the operating mode is "Delete", the input table to the SaleforceOut Action must contains a column named "Id" that contains the list of "Id" to delete from the given table inside Salesforce.

When the operating mode is "Insert" or "Update", the columns of the input table to the SaleforceOut Action must match the column's names of the table to modify in Salesforce. For your convenience, you can get the required column's names when clicking here:
You don't need to provide *all* the columns from the Saleforce table in input: Many columns are usually optional.

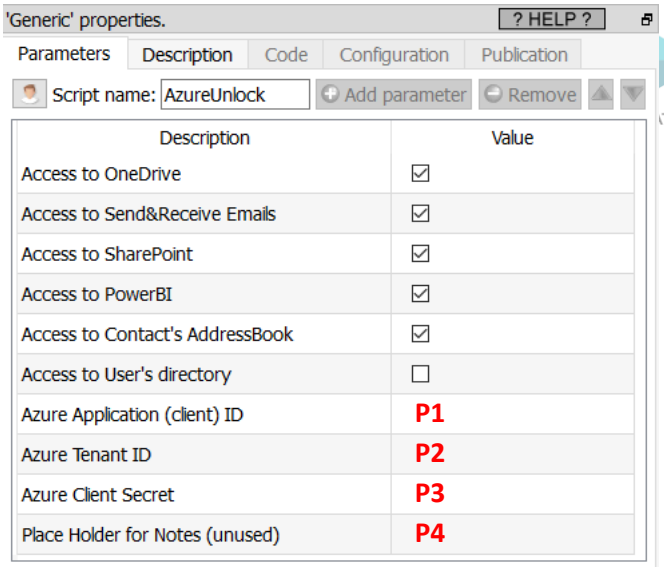
When the operating mode is "Update", you must provide in input a column named "Id" (in addition to the required columns to upload&update into Saleforce).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.27. Unlock Azure services

Icon:

Property window:



Short description:
Unlock access to Azure services

Long Description:
You need to get your Azure credentials (i.e. you need to get from Azure these 2 parameters: your "Application (client) ID", and your "Anatella Unlock Key") before using any of the following Actions:

- OneDrive:
  - List the files on a OneDrive (section 5.23.35)
  - Download files from a OneDrive (section 5.23.36)
  - Upload files to a OneDrive (section 5.23.37)
  - Delete files stored in a OneDrive (section 5.23.38)
- Email
  - Receive Emails from an Azure account (section 5.23.53)
  - Send Emails using an Azure account (section 5.23.54)
- SharePoint
  - Retrieve Lists and Objects from a SharePoint site (section 5.23.28)
  - Add items inside a SharePoint List (section 5.23.29)
- PowerBI: Add/Update/Delete datasets inside PowerBI (section 5.23.90)
- Manage Contact from Azure users and List All users (section 5.23.98)

All these Azure fonctionnalities require you to pay to Microsoft. You must pay even for the most basic functionalities such as listing some files on your OneDrive. Hopefully, if you buy the "Office 365 E1" (On the 2022-11, the cost is: 11.80€/month or 100.8€/year) you already get access to your OneDrive, to your Email and to SharePoint.

A limited version of PowerBI is also accessible for free.

First, a word of caution:

> You will need your "Application (Client) ID" and an "Anatella Unlock Key" to use (nearly) all the Azure Services actions inside Anatella.
>
> **There are no ways to copy/paste back these 2 informations from one Anatella action to another (this is, of course, "by design").**
>
> So, you should keep yourself these 2 informations in a secure place, if you intend to re-use them later.

To get your Azure Credentials (i.e. to get a "Application (client) ID" and an "Anatella Unlock Key") you need to follow the procedure given inside the next two sections 5.23.27.1. (First Time Setup) **and** 5.23.27.2. (Get or Renew your Anatella Unlock Key).

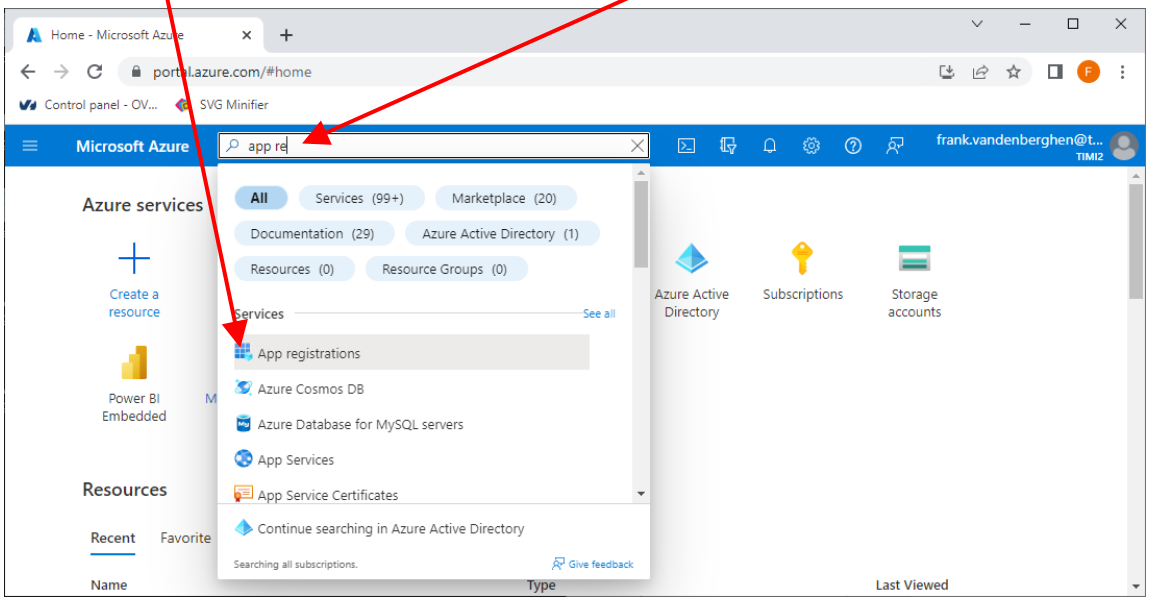
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Your “Application (Client) ID” will never expires. In opposition, your “Anatella Unlock Key” will expire after a few months (that depends on the settings that you selected at the step 20.3 from the next section 5.23.27.1.). The maximum duration of an “Anatella Unlock Key” is 2 years (this limitation is common to all the tools accessing Azure since it’s imposed by Azure). When your “Anatella Unlock Key” expires, you just need to redo the easy 3-steps procedure given inside section 5.23.27.2 (there is no need to redo the whole procedure from section 5.23.27.1!!).

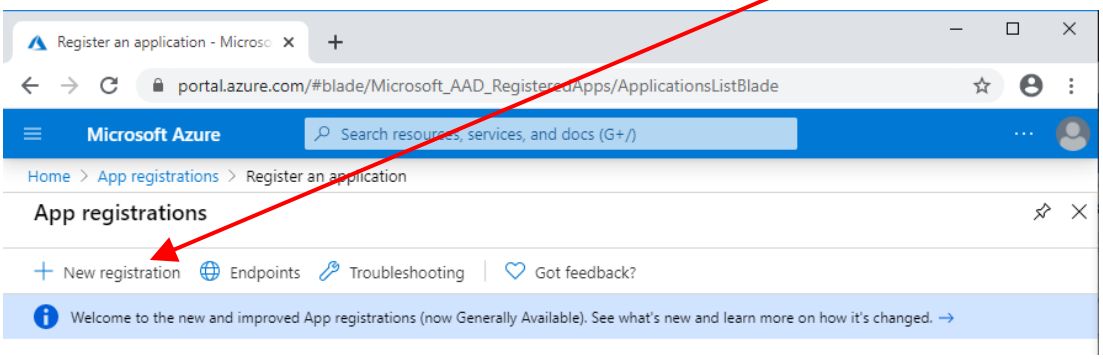
5.23.27.1. Azure First Time Setup

The procedure that is given inside this section must only by performed **once** by an Azure Administrator. The first-time-setup procedure to connect to Azure is the following:

1. Open the URL https://portal.azure.com and “log-in” using your normal “Login” and “Password” for Azure. Then, type “App Registration” in the searchbar and click on the corresponding icon:

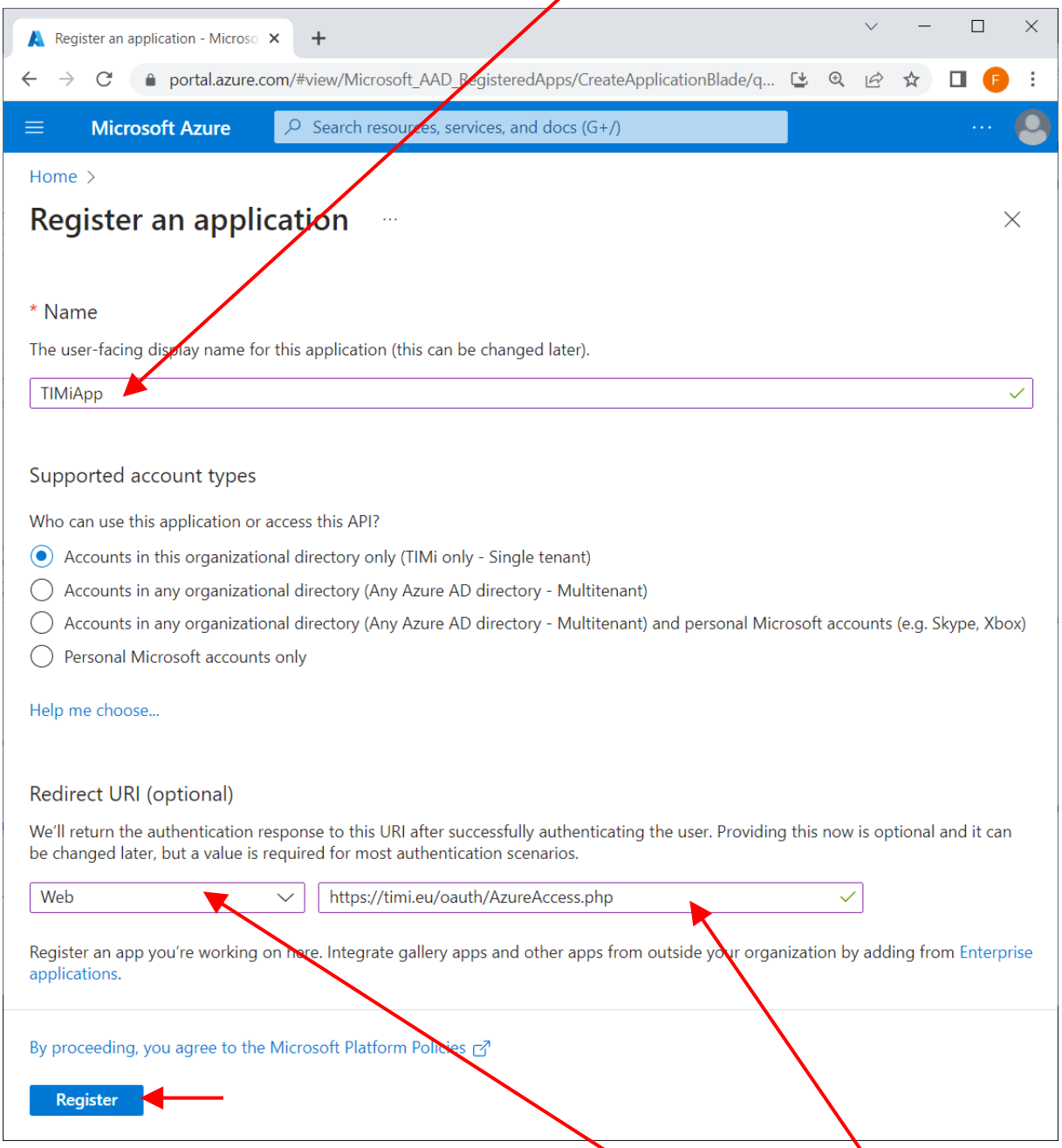


2. We will create a new Azure App: Click the “New registration” button:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Give a name to your Azure App (you can use any name)



4. For the redirect URI: select the "Web" type of application here and enter as URL: https://timi.eu/oauth/AzureAccess.php

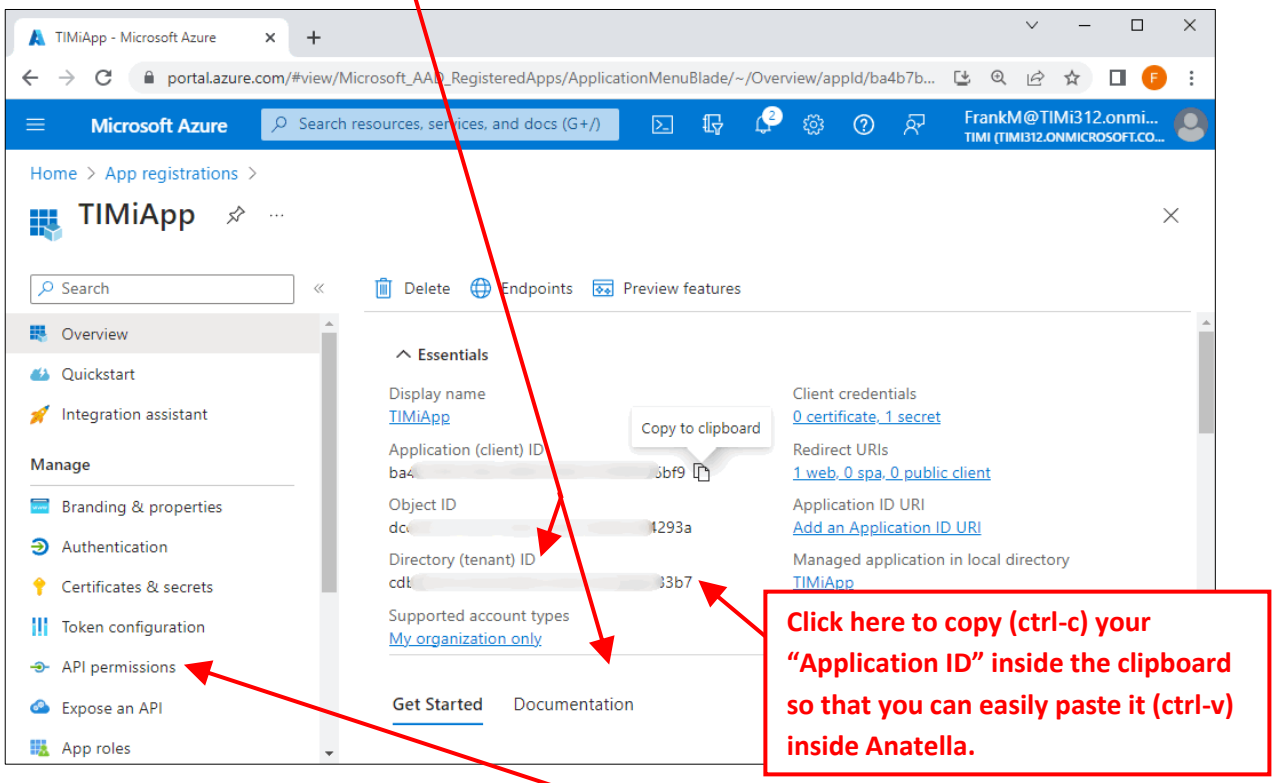
   …as the "redirect URI".

   WARNING: This is a very important step: The URI must be exactly the one given here above (this is case sensitive) and the type of the application must be "Web".

5. Click the "Register" button at the bottom of the webpage to go to the next page.

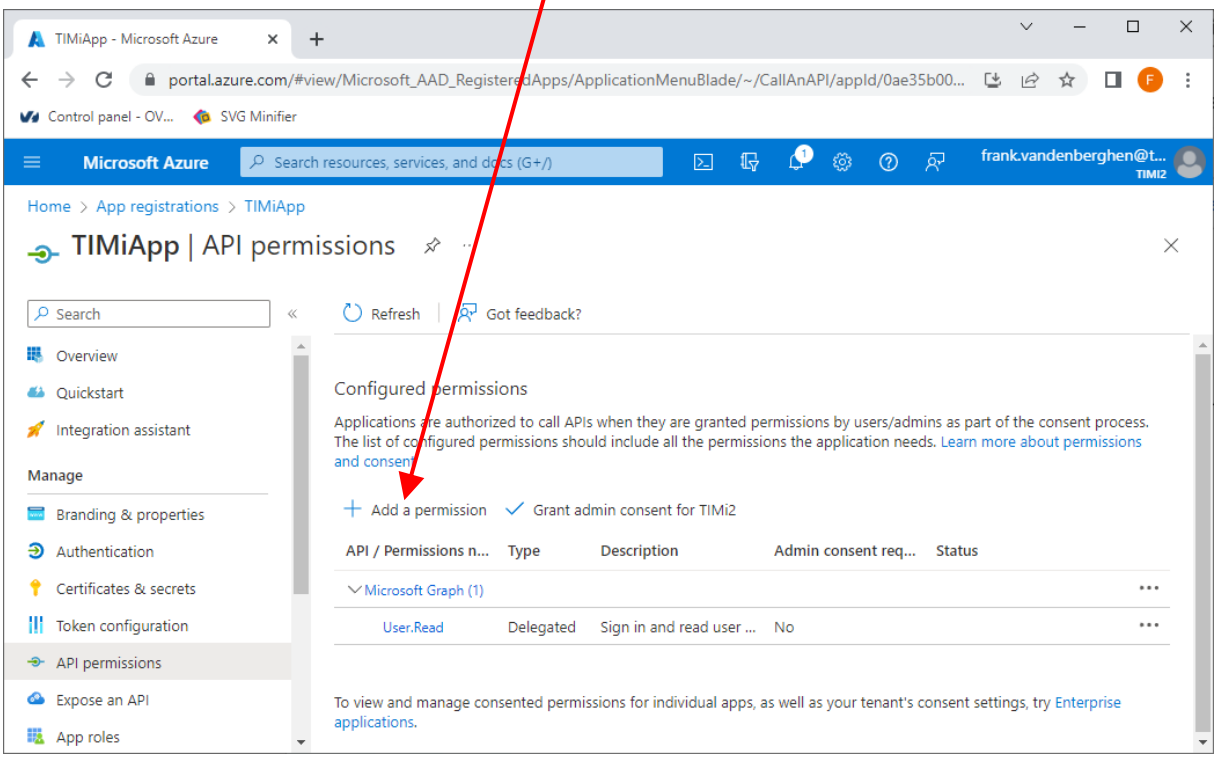
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. You are now receiving the Anatella parameter **P1** and **P2** (i.e. your "Application (client) ID" and your "Tenant ID"): They are here:



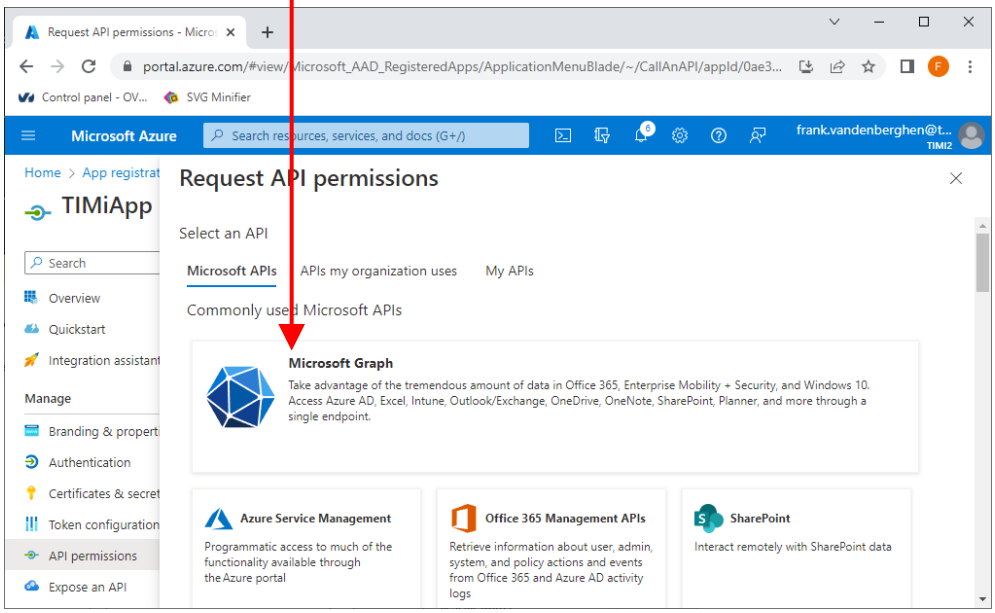
7. Click on "API Permissions" in the left panel here:
8. **Grant Permission to access SharePoint (you can skip the section 9 if you don't intend to access SharePoint)**

9.1. Click the "+ Add permission" button here:

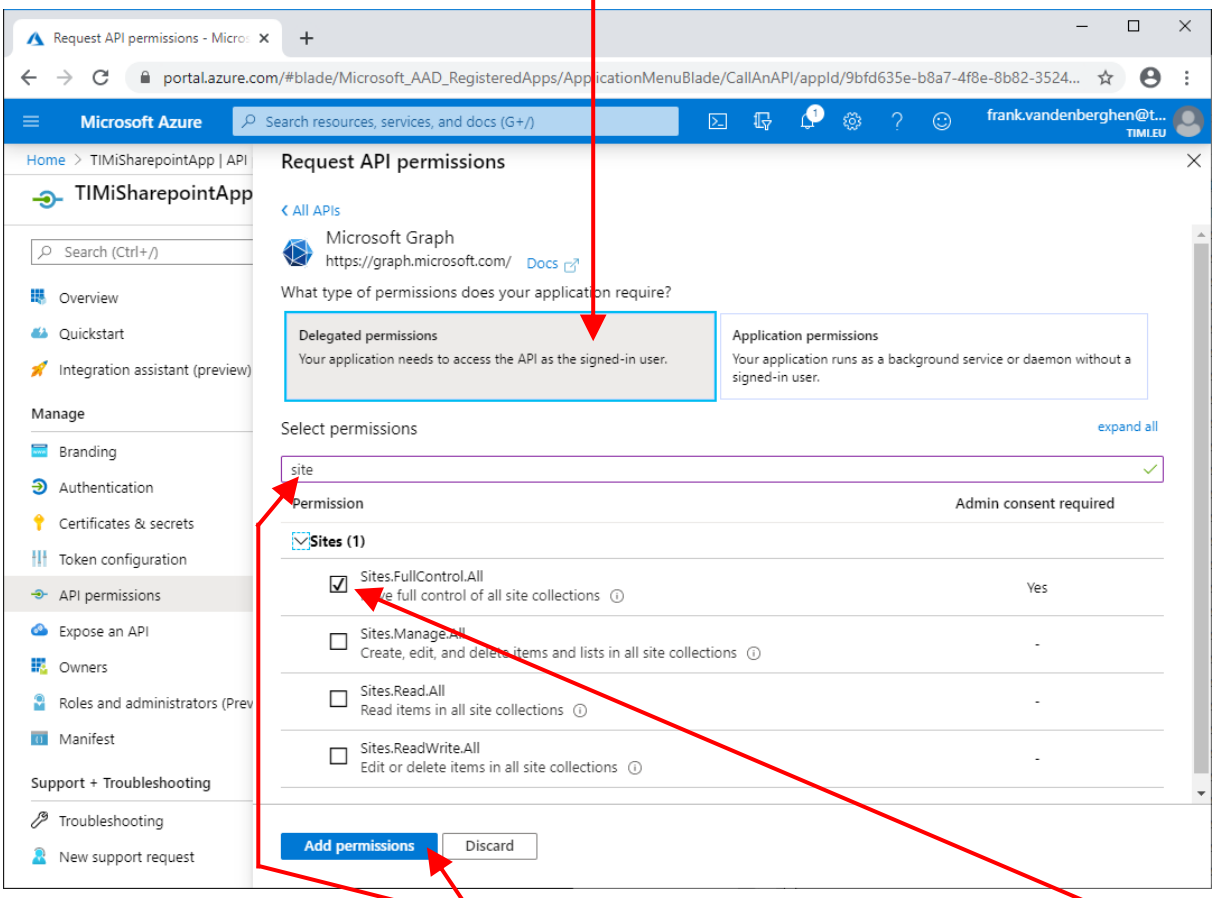


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.2. Click on “Microsoft Graph”:



9.3. Click on the “Delegated Permissions” button:

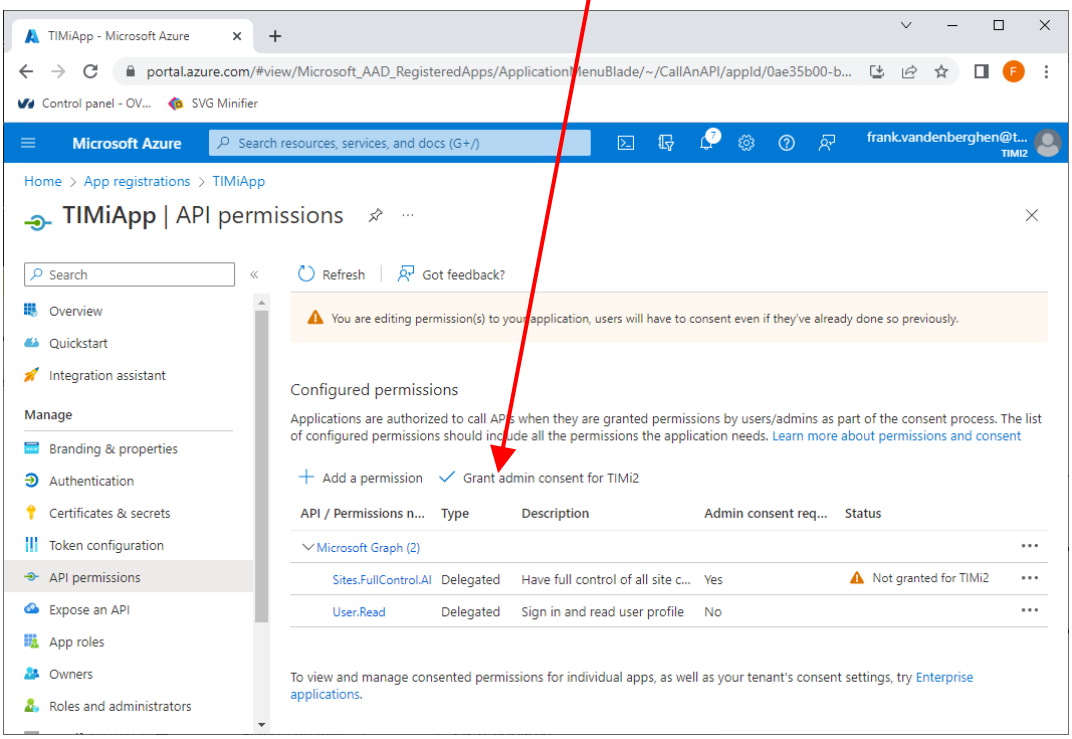


9.4. Write “site” in the search box here: ..and tick the checkbox named “Sites.FullControl.all” here:

9.5. Click on the “Add permissions” button:

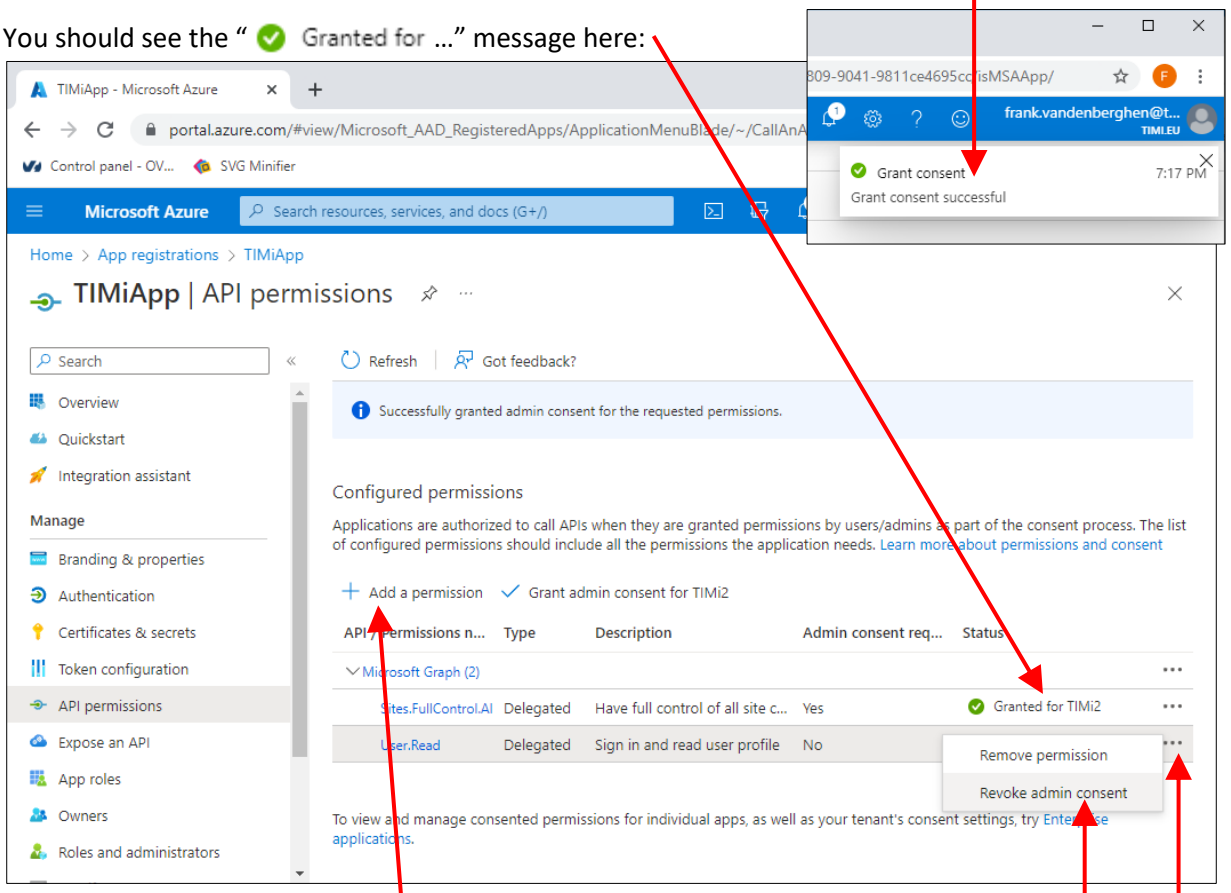
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.6. Click on the “Grant admin consent for…” button:



9.7. You should see “Grant consent succesful” in the top left corner of your browser:

9.8. You should see the “ Granted for … ” message here:

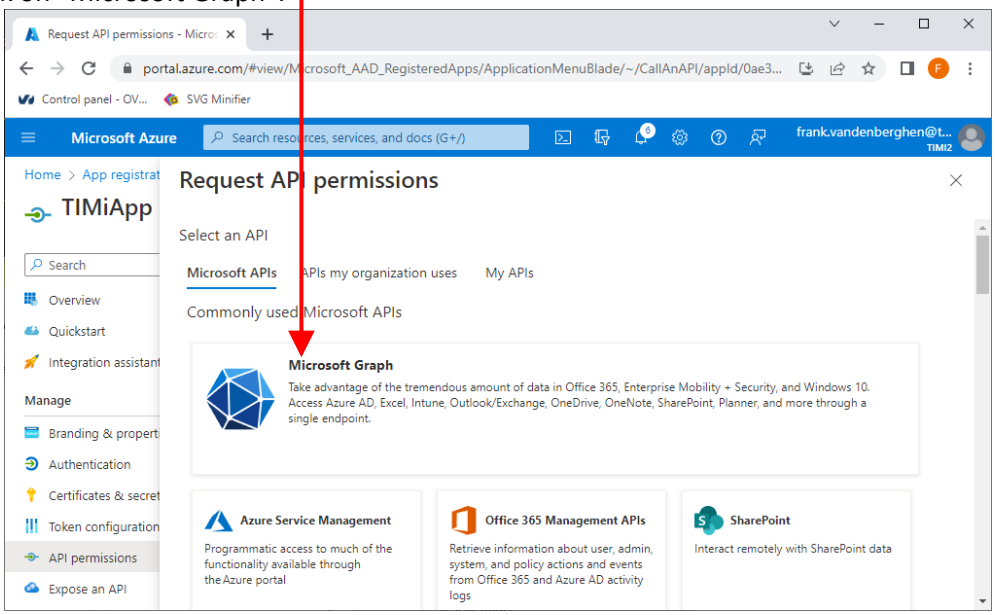


9.9. Click on the ··· icon for the “user Read” permission: …and click “Revoke admin consent”:

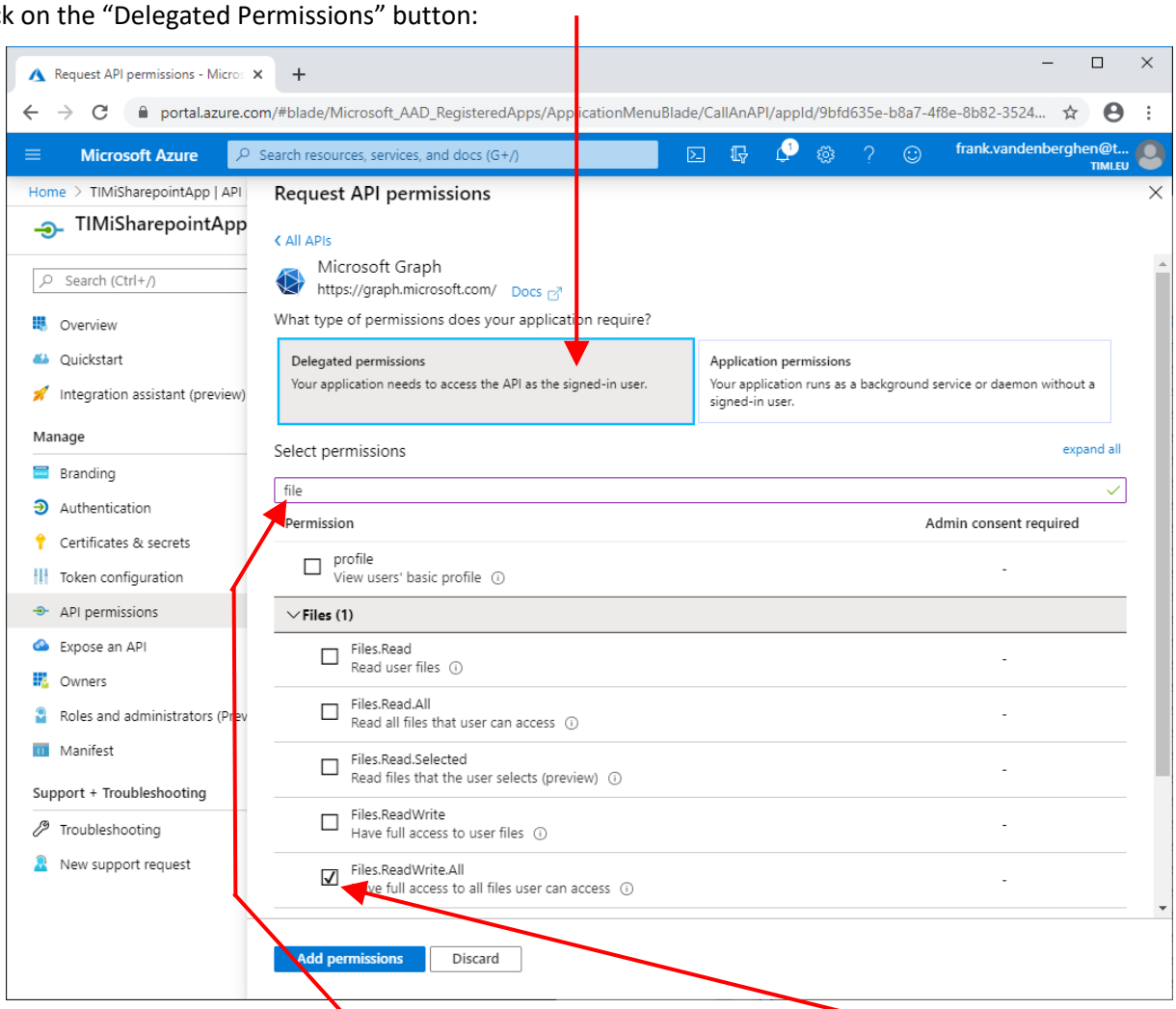
10. Click the “+ Add permission” button here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

11. Click on “Microsoft Graph”:



12. Click on the “Delegated Permissions” button:



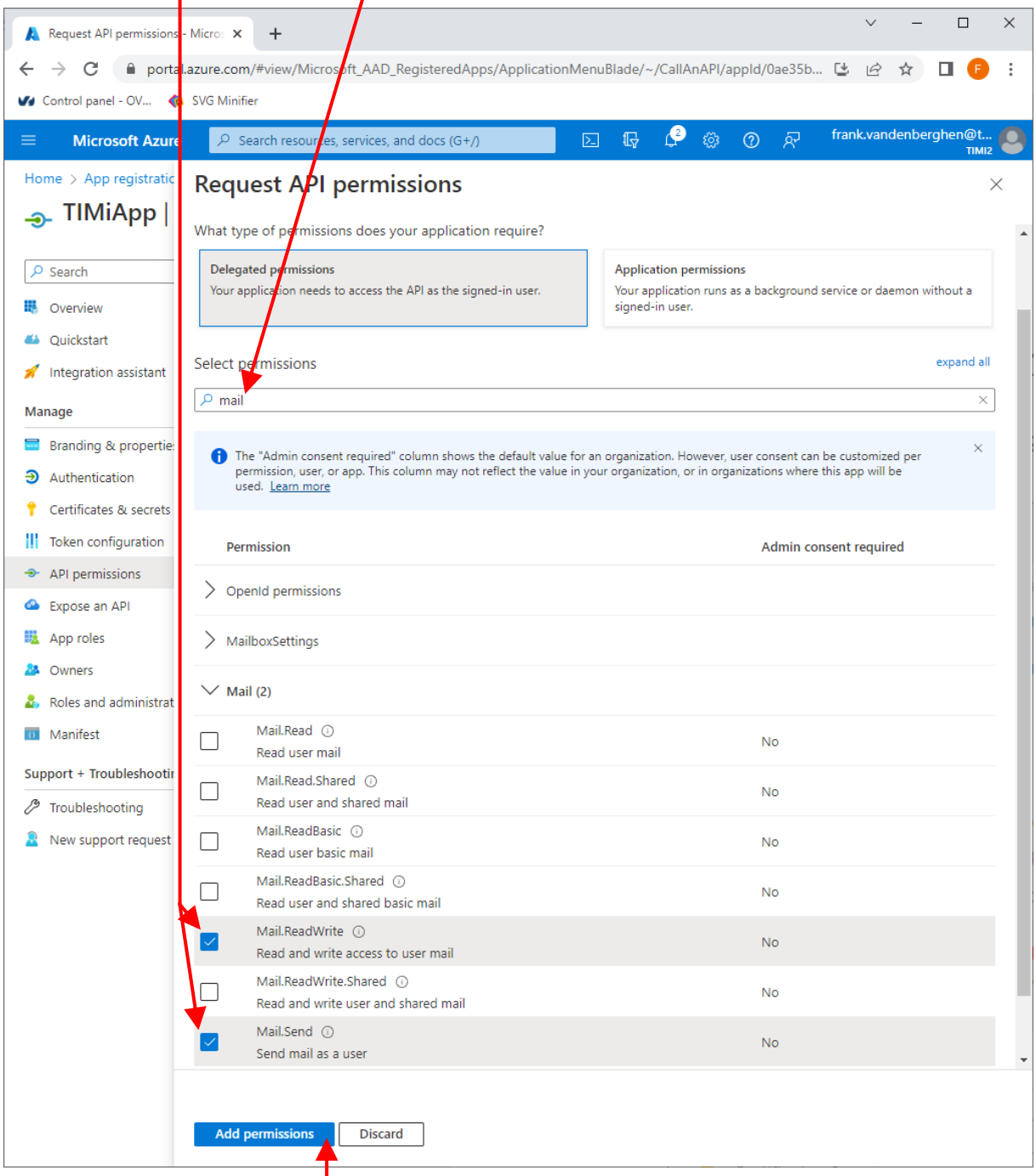
13. **<u>This permission is required to use OneDrive:</u>**

Write “file” in the search box here: ..and tick the checkbox named “Files.ReadWrite.all” here: (do not enable this permission if you don’t intend to access OneDrive)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

14. **<u>This permission is required to send/receive emails using Office 365:</u>**

Write “mail” in the search box here: ..and tick the checkboxes named “Mail.ReadWrite” and “Mail.send” here:(do not enable this permission if you don’t intend to send/receive emails using office 365)



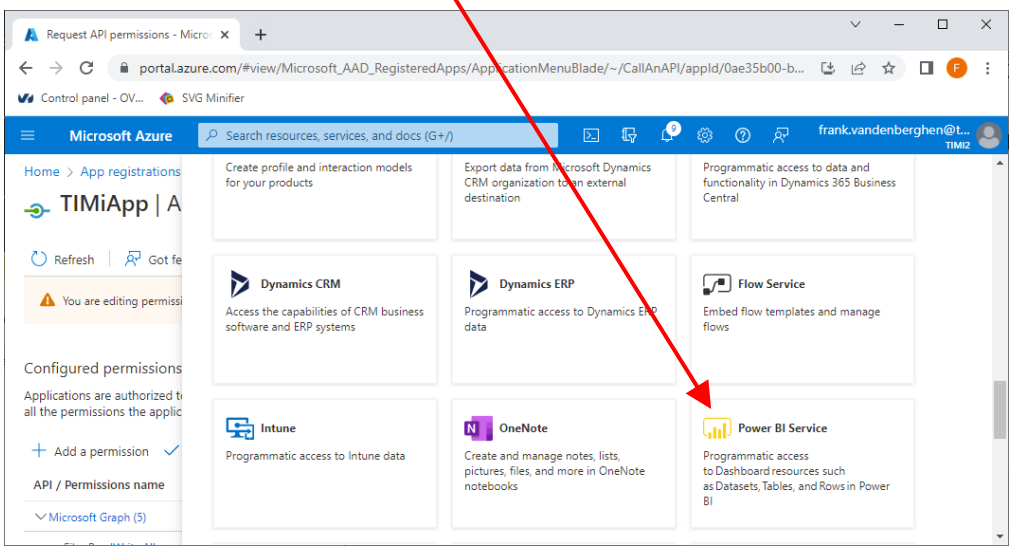
15. Click the “Add Permissions” button

16. **Grant Permission to access PowerBI (do not follow section 16 if you don’t intend to access PowerBI)**

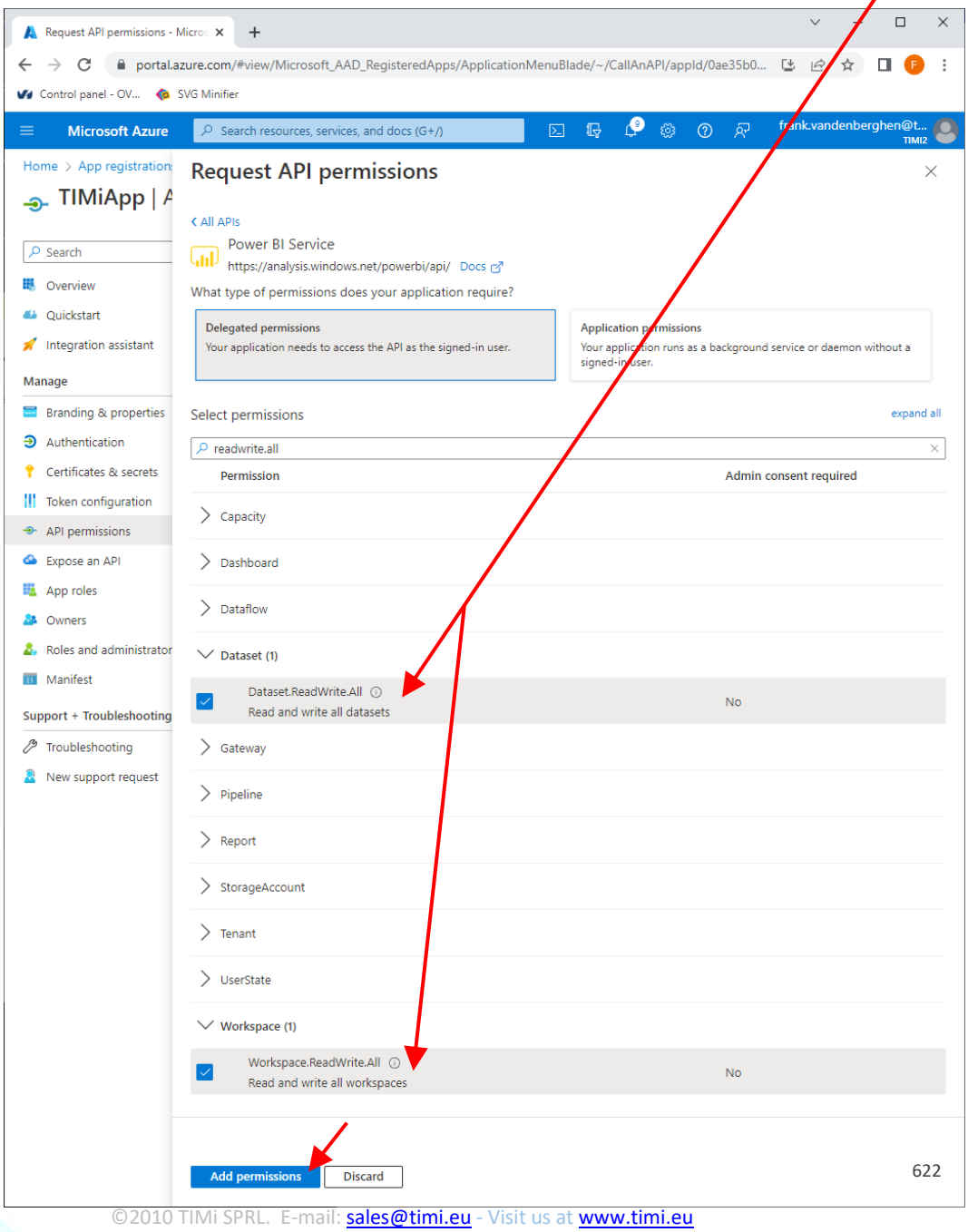
16.1. Click (again) on the + Add a permission button

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

16.2. Scroll down and click on “Power BI Service”:



16.3. Check the permissions named “Dataset.ReadWrite.All” and “Workspace.ReadWrite.All”:



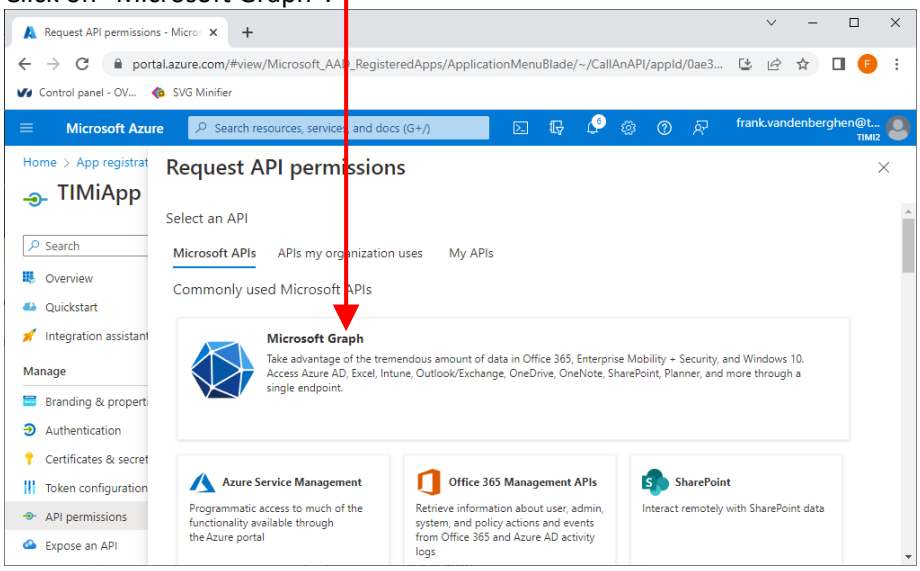
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

16.4. Click the **Add permissions** button.

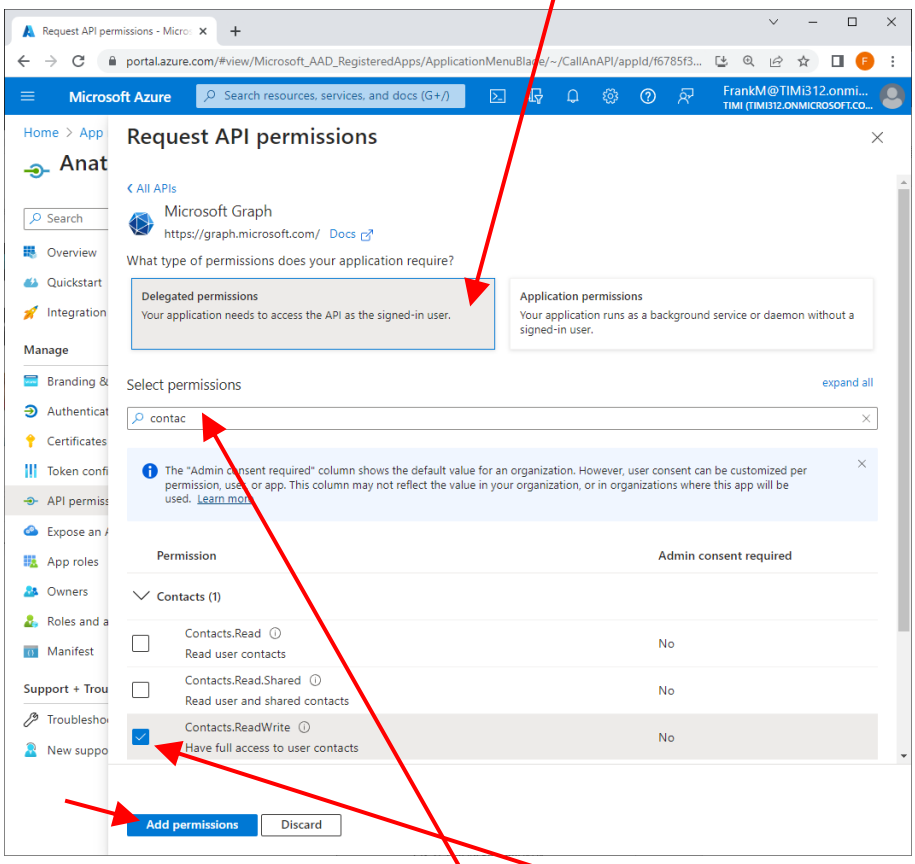
17. Grant permission to manage the Contacts of the users (Do not follow the section 17 if you don't intend to use the AzureAddContact Action)

17.1. Click (again) on the + Add a permission button

17.2. Click on "Microsoft Graph":



17.3. Click on the "Delegated Permissions" button:



17.4. Write "contact" in the search box here:
..and tick the checkboxes named "Contact.ReadWrite" here

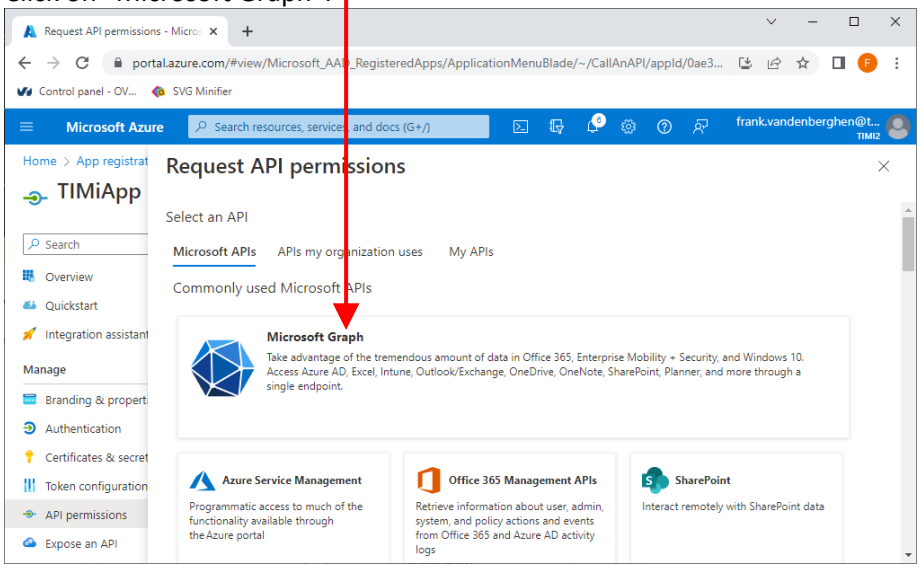
17.5. Click the **Add permissions** button

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

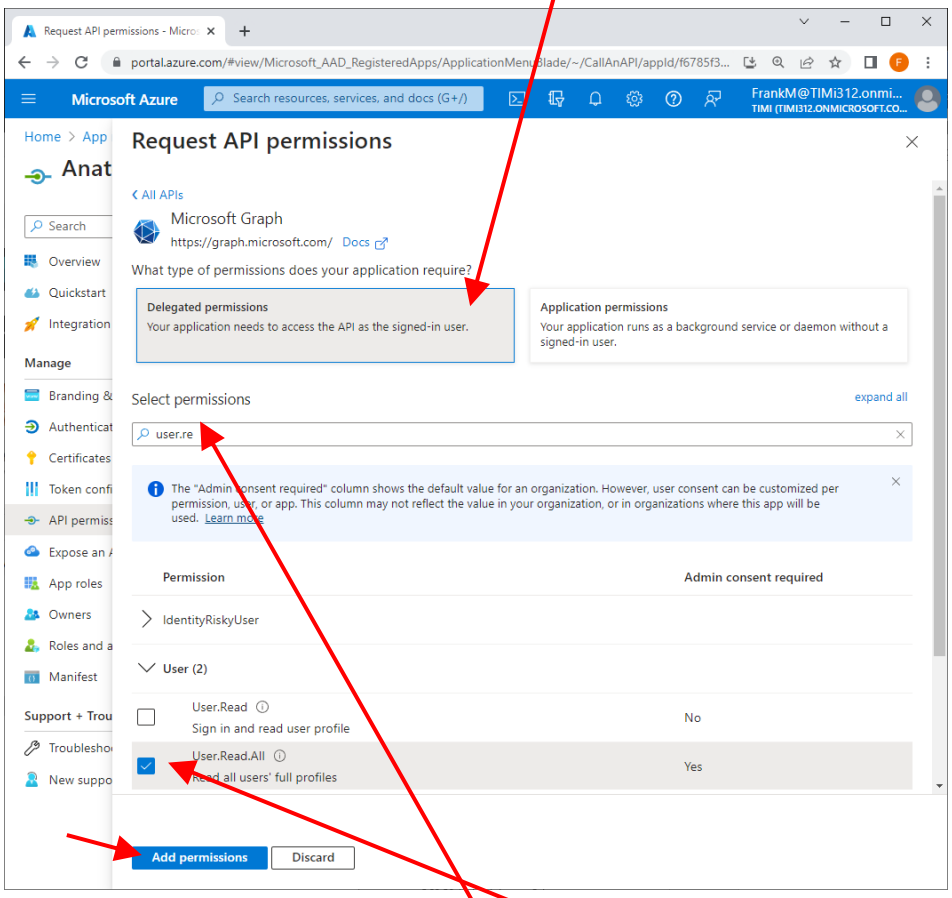
18. Grant permission to list the users inside your Organization (Do not follow the section 18 if you don't intend to use the special operation "List Users" (Parameter P6) from the AzureAddContact Action). This step requires admin. consent.

18.1. Click (again) on the + Add a permission button

18.2. Click on "Microsoft Graph":



18.3. Click on the "Delegated Permissions" button:

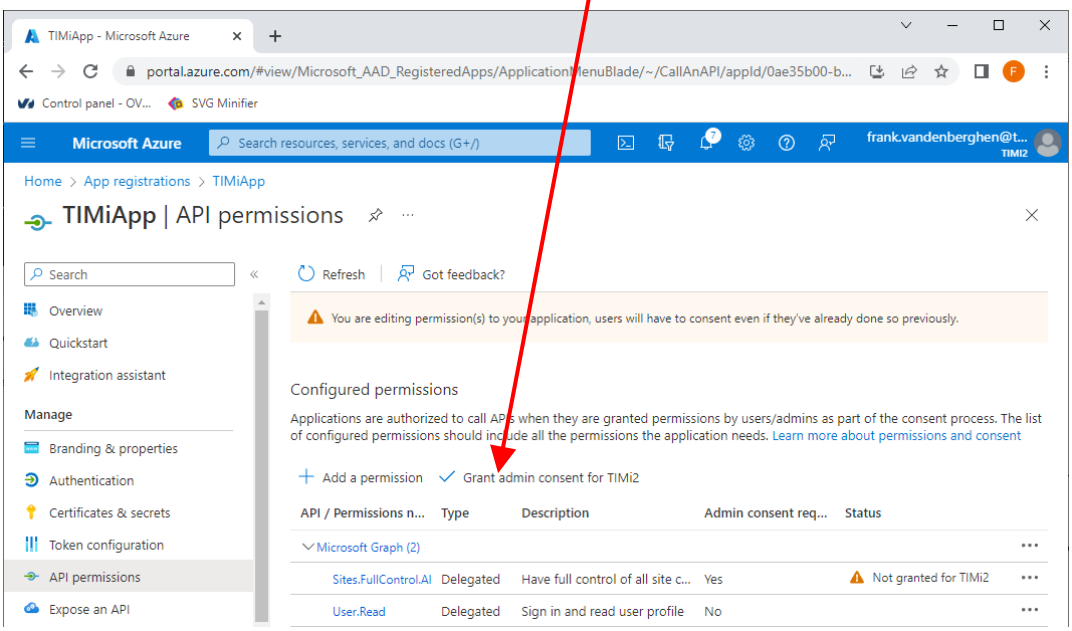


18.4. Write "user.read" in the search box here:
..and tick the checkboxes named "User.Read.All" here

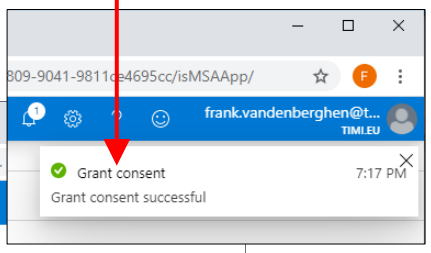
18.5. Click the Add permissions button

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

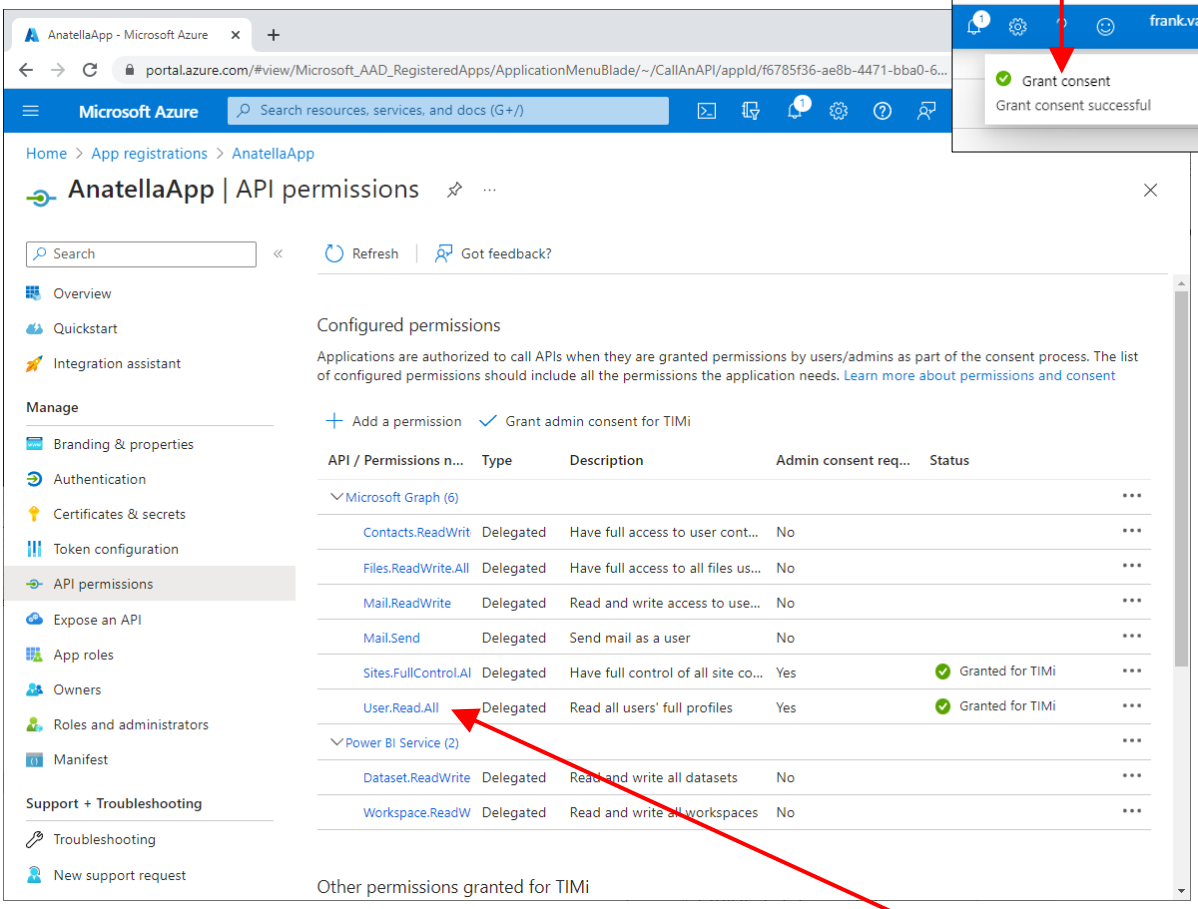
18.6. Click on the “Grant admin consent for…” button:



18.7. You should see “Grant consent succesful” in the top left corner of your browser:



19. If you enabled all permissions, you should now see:



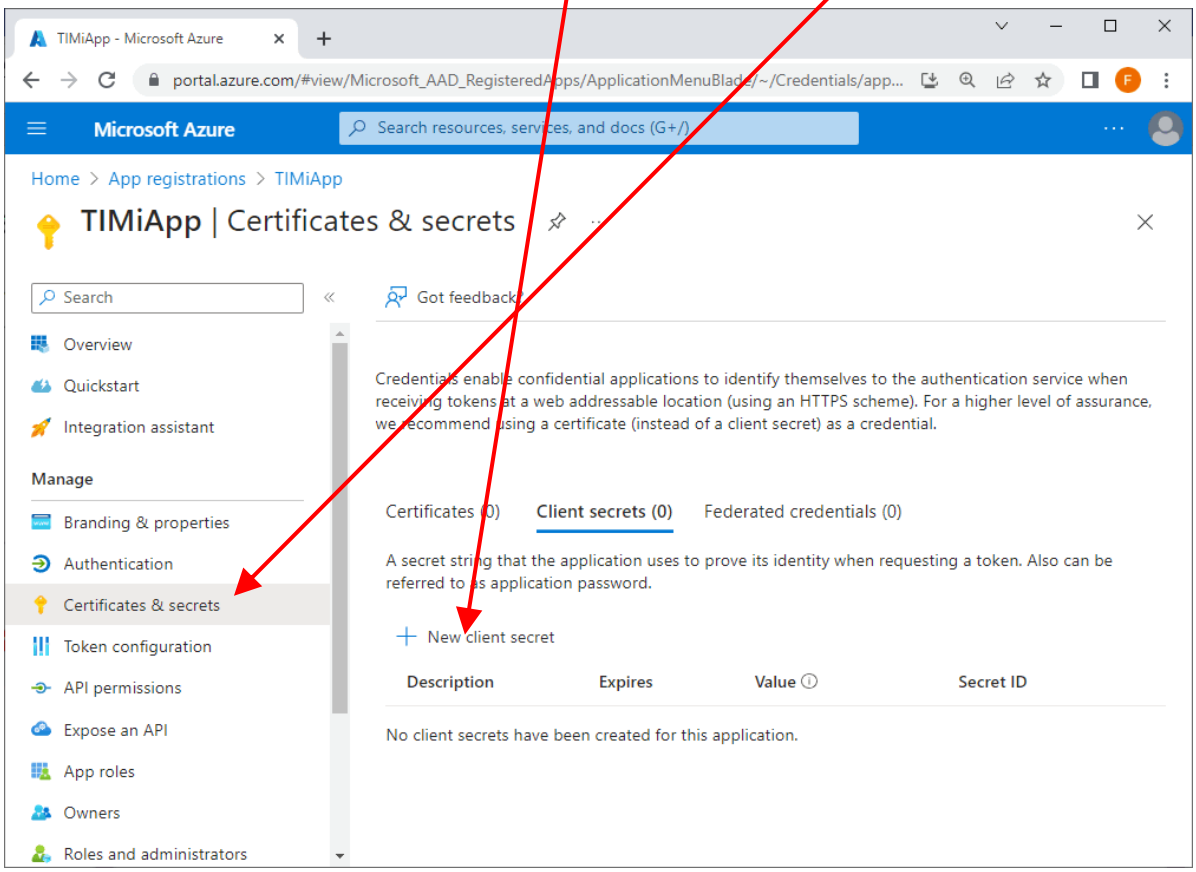
If you didn’t follow step 18, this is just “User.Read” (instead of “User.Read.All”)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

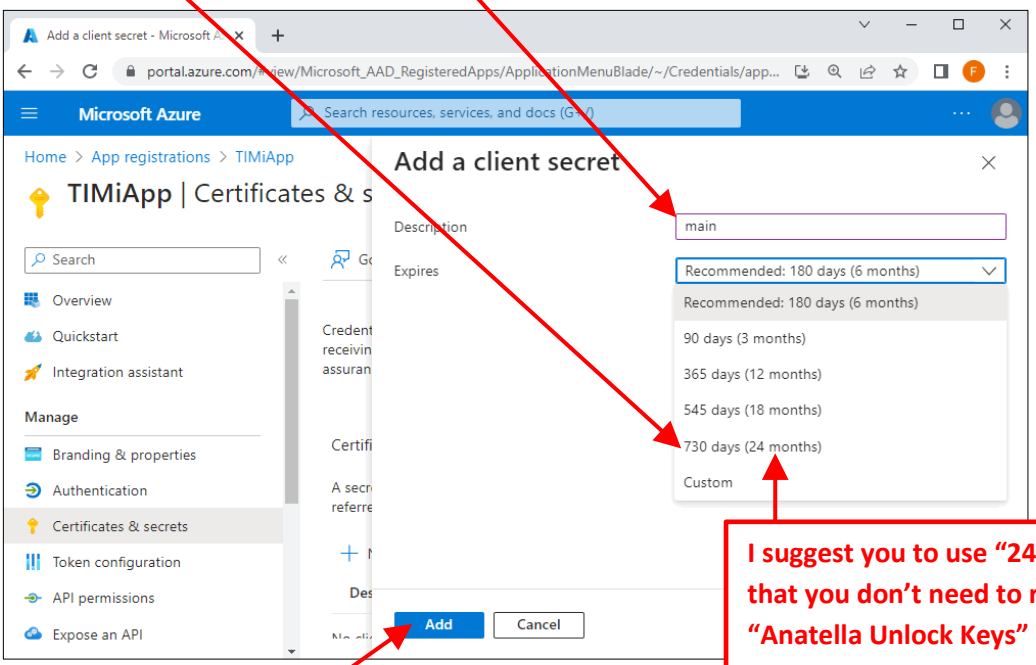
20. Get the parameter **P3** "Azure Client Secret".

20.1. Click on the section "Certificates and secrets" inside the Left-Menu

20.2. Click on the "+ New client secret" button:



20.3. Given any name for the description and select the maximum validity period of your "Anatella Unlock Key":

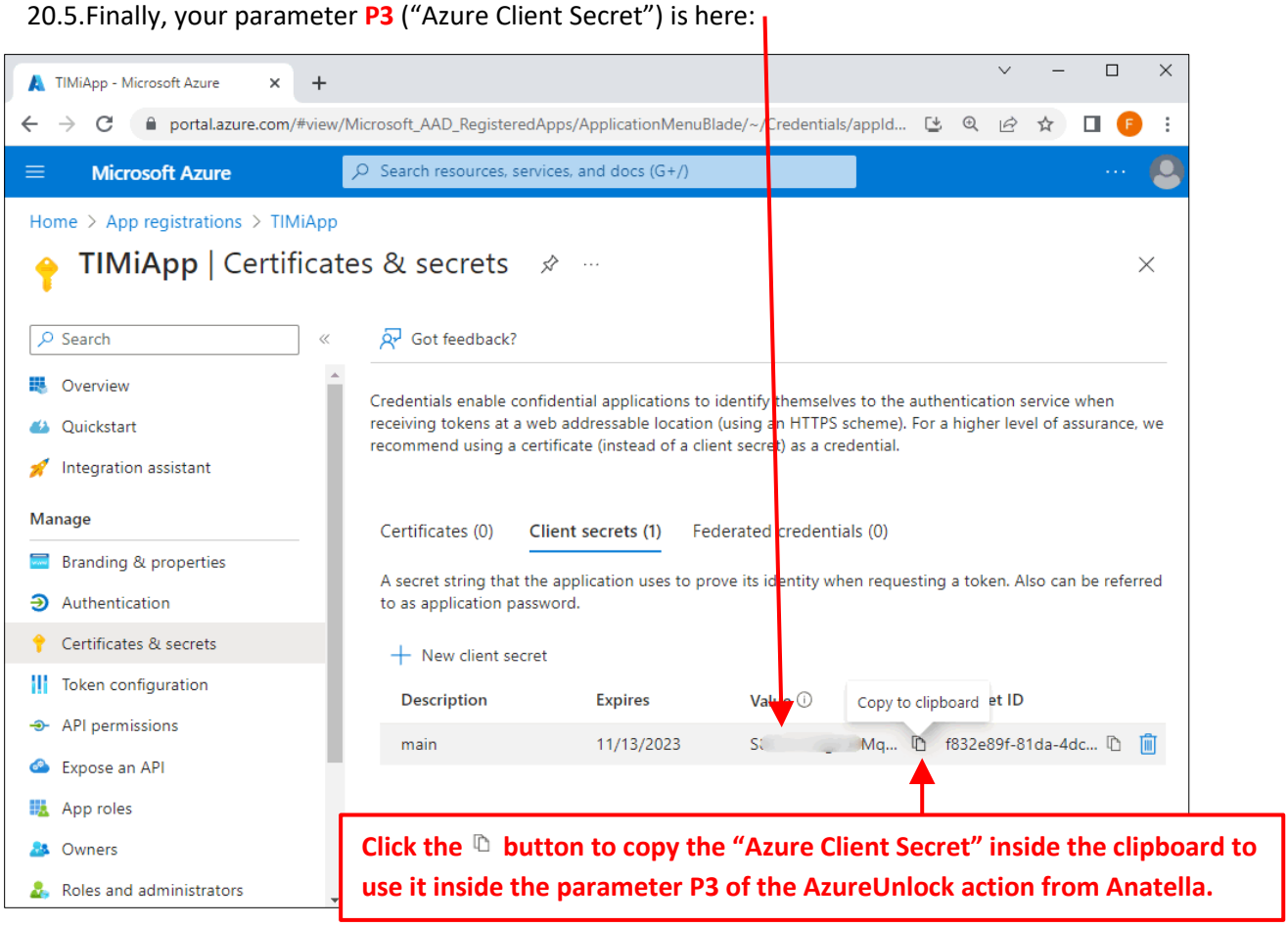


I suggest you to use "24 months" so that you don't need to renew your "Anatella Unlock Keys" all the time.

20.4. Click the "Add" button:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

20.5. Finally, your parameter **P3** ("Azure Client Secret") is here:



20.6. There is no way to retrieve back the parameter **P3** ("Azure Client Secret") if you lose it:
i.e. Store this parameter preciously!

21. Proceed to the next section to get your first "Anatella Unlock key" to unlock the integration between Anatella and Azure!

5.23.27.2. Get or Renew your Anatella Unlock Key.

Follow the procedure given in this section to get your first "Anatella Unlock key"!

Also: Each different Azure user that wants to access his own Azure services (or his own Azure data) must get his own personal "Anatella Unlock key" by following the procedure given here below.

Also, after a few months, your "Anatella Unlock key" will expire (this depends on the settings that you selected at step 20.3 from the previous section 5.23.27.1.). The maximum duration of an "Anatella Unlock Key" is 2 years (this is common to all tools accessing Azure since it's imposed by Azure). When your "Anatella Unlock Key" expires, you just need to redo the easy procedure given below inside the current section.

The procedure to get (or renew) your "Anatella Unlock key" is:

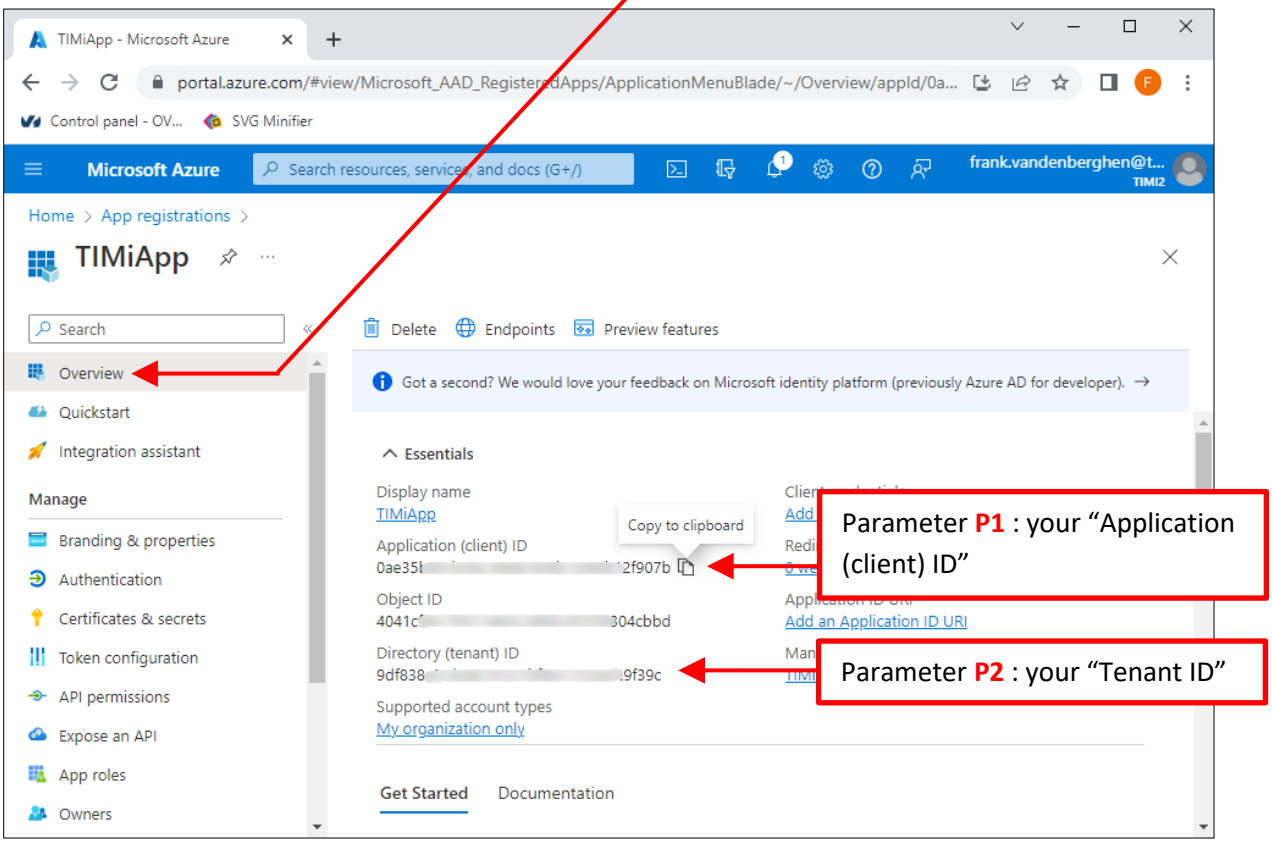
1. Go back inside Anatella. Place an "AzureUnlock" action inside your graph and open its properties. Inside the "AzureUnlock" action, you need to fill-in the parameters **P1**, **P2** and **P3** using the values obtained while following the procedure from the previous section5.23.27.1.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

More precisely:

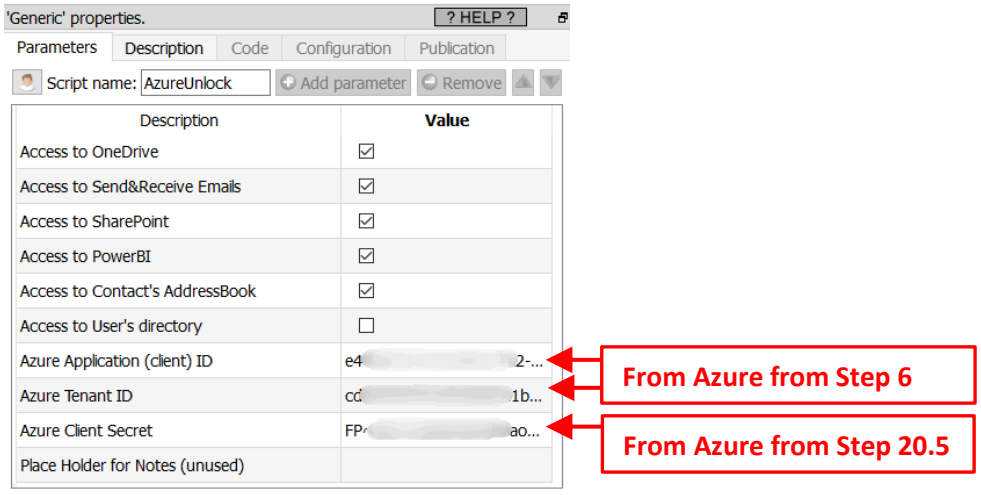
- The parameters **P1** and **P2** (i.e. your “Application (client) ID” and your “Tenant ID” were obtained during the step 6 of the procedure given in the previous section.

  Just in case: you can click on the Overview button on the top of the left panel inside the Azure portal to see again these two parameters **P1** and **P2**:



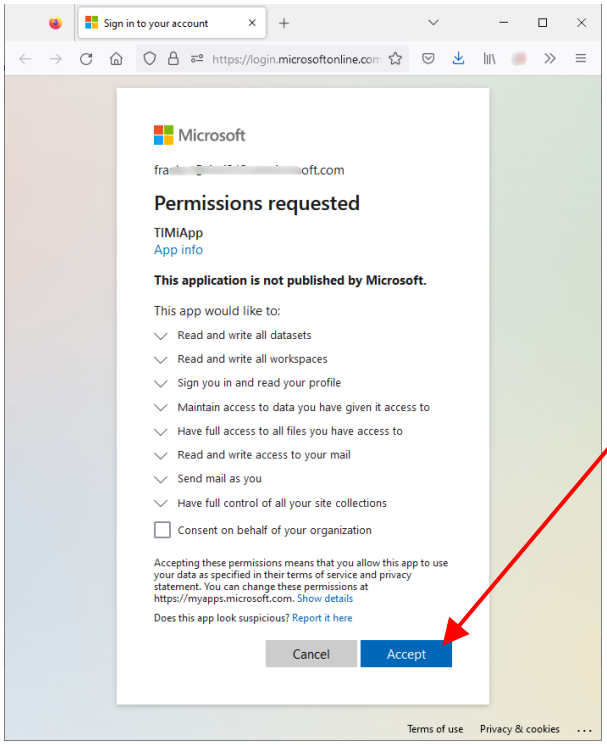
- The parameter **P3** (i.e. your “Azure Client Secret”) was obtained during the step 20.5 of the procedure given in the previous section.

You should now see something like:

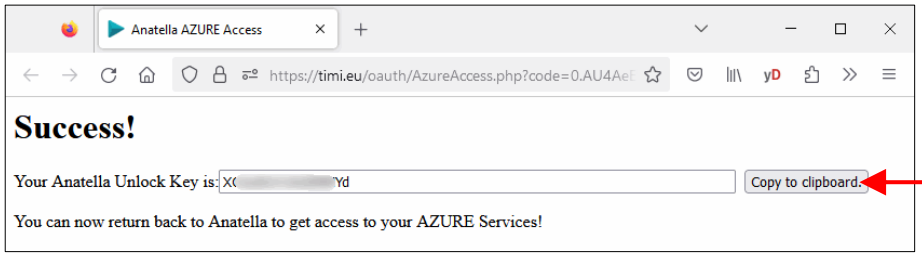


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Run the AzureUnlock action. An internet browser opens. Sign-in into the account that possesses the OneDrive/Email/SharePoint/PowerBI that you want to access. Click on “Accept”:

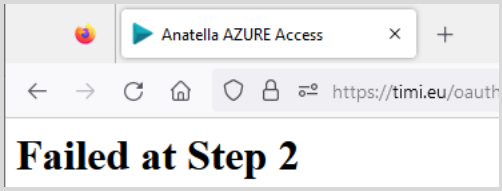


3. You finally get your “Anatella Unlock Key”:



Click here to easily copy/paste your “Anatella Unlock Key” inside Anatella

Sometime, instead of the “Success !” page (that you see here just above), you get:



…followed by a bunch of garbage codes.

Don’t worry: That’s just Azure that is a little bit slow. Your new application settings need a little bit of time to propagate. Just wait a few seconds and run again the AzureUnlock action and, at some point, it will give you the required “Anatella Unlock Key”.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Don't stop directly once you have your Anatella Unlock Key!

I strongly advise you to also secure your Azure account by following the procedure given inside the next section 5.23.27.3.!

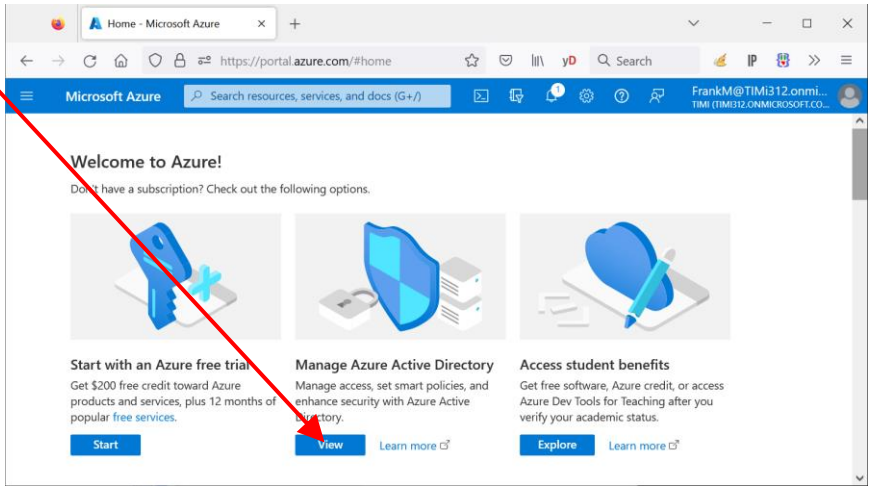
Your can copy/paste your "Anatella Unlock Key" into the parameter **P4** of the AzureUnlock action. This acutally doesn't do anything: That's just a common place to store it (just in case you need it later).

Also: You should keep somewhere safe the graph with your current AzureUnlock action because you'll need to run it again later to renew your "Anatella Unlock Keys" (once they expire after a few months).

5.23.27.3. Securing access to your Azure account

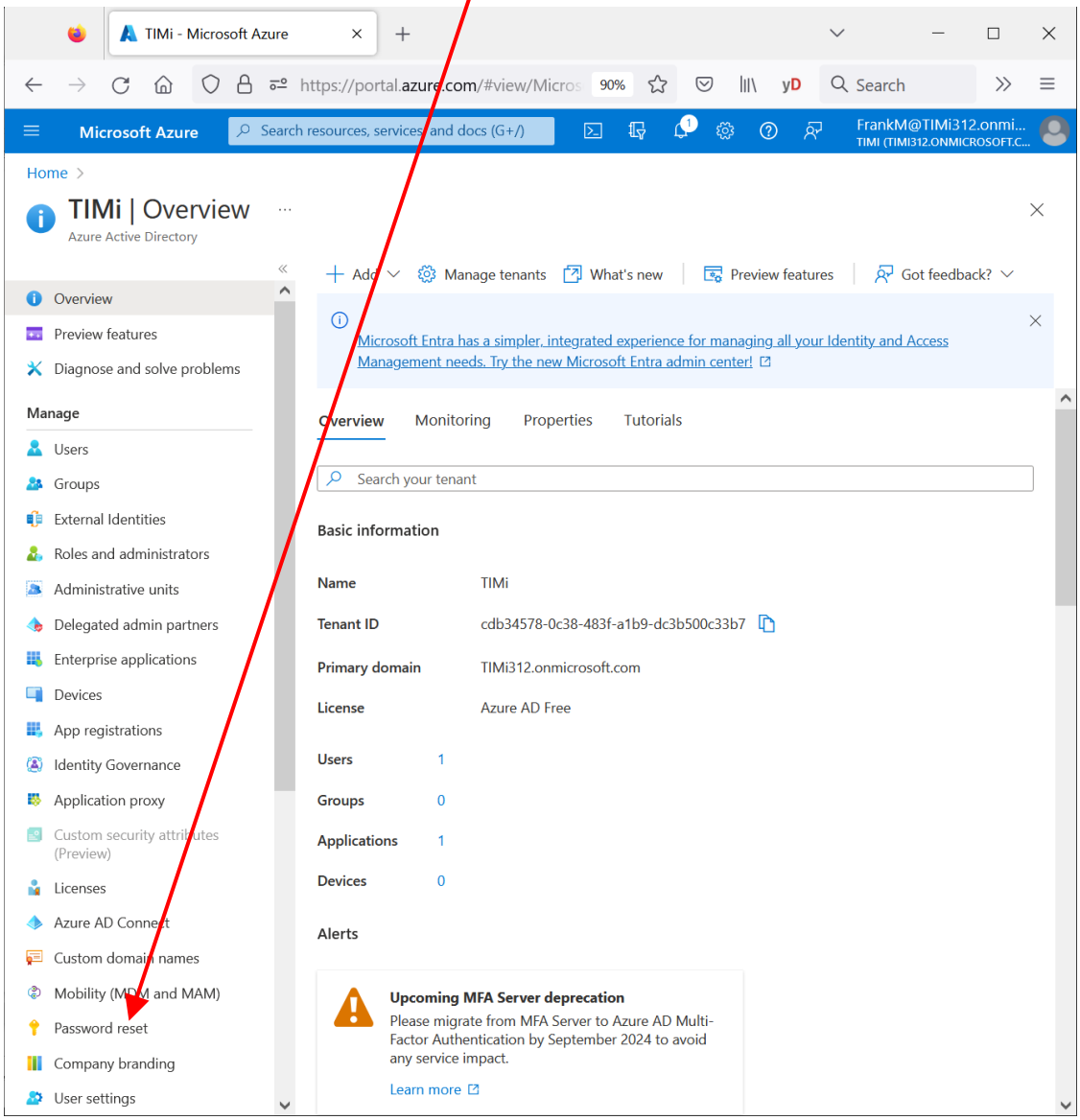
In the unfortunate situation where you lost your Azure password, you might want to reset it. By default, Microsoft does not allow you to reset your password (this is typically done using an SMS or an email message). By default, if you lose your Azure password, you basically lose everything because there is no way to reset it, even when contacting the hotline from Microsoft (Trust me on that!). Let's now see how to activate the option that allows you to reset your password:

1. Open the URL https://portal.azure.com and "log-in" using your normal "Login" and "Password" for Azure. Click on the "view" button below the "Manage Azure Active Directory" icon:



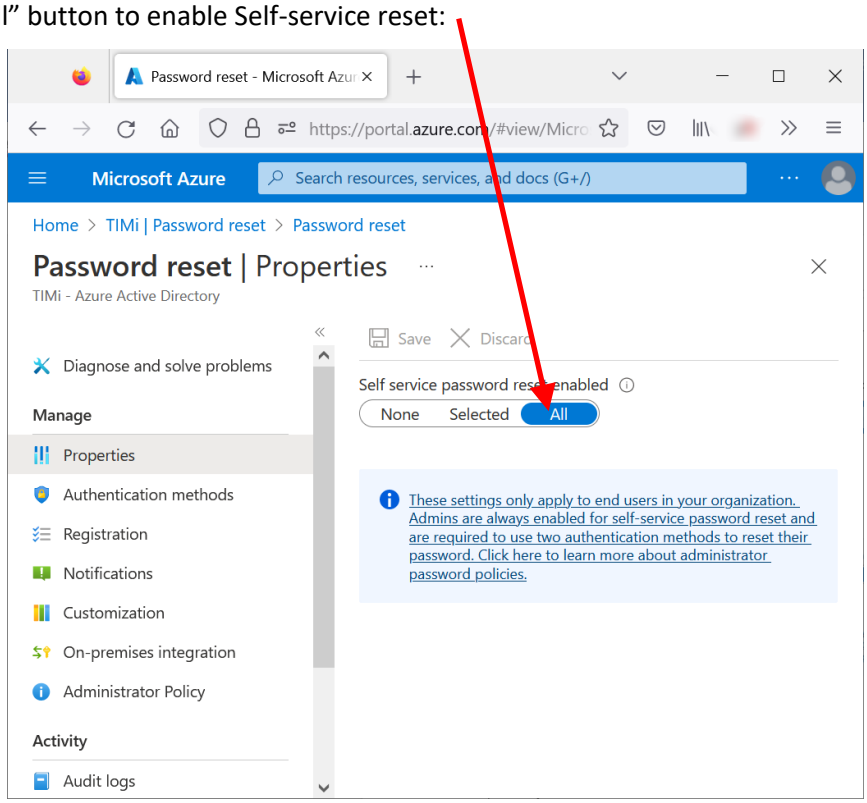
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Click on the “Password reset” entry in the left menu:



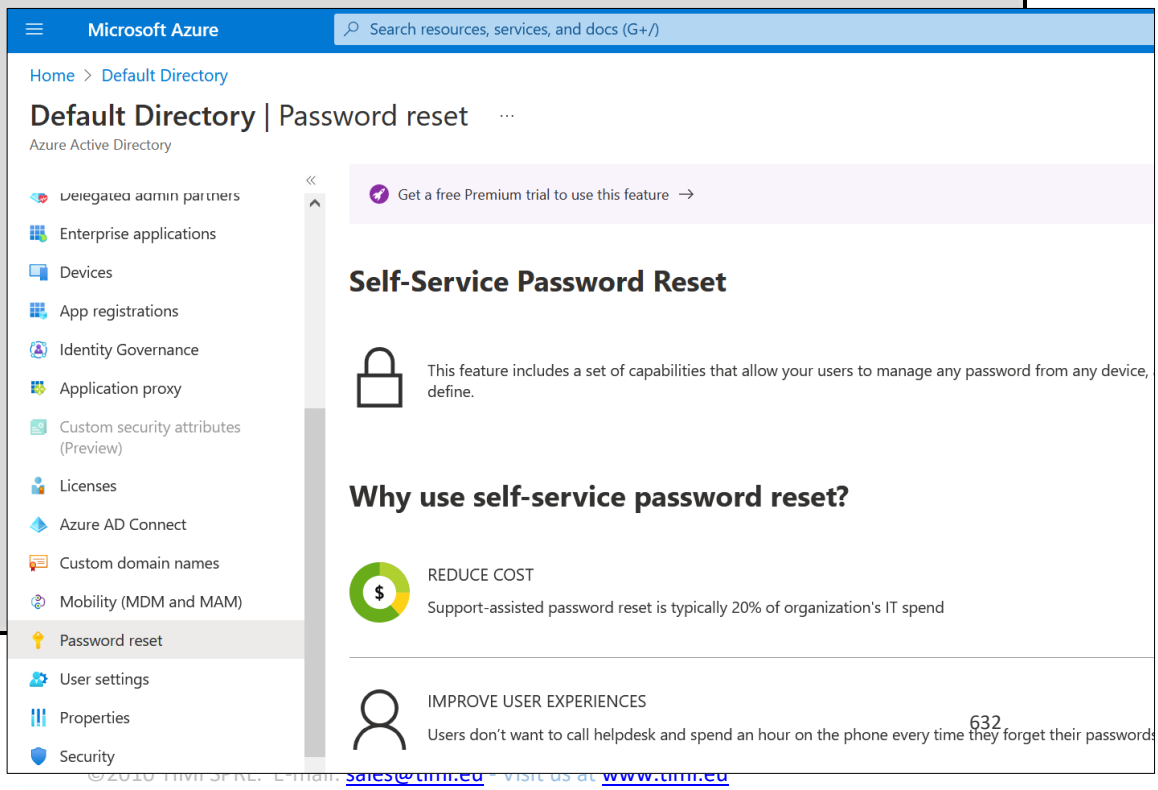
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Click on the “All” button to enable Self-service reset:



4. From now on, you should be able to reset your password (and get back access to your Azure account) in case you lost it.

If you didn’t pay Microsoft (e.g. you are using the free access to setup Azure or you temporarily paused you Microsoft Office 365 subscription), you loose access to the “self-service-password-reset” service. In such situation, if you lose your password, all is lost (because you’ll never be able to reset your password). Also, if you are using the free access to Azure, you don’t have access to the “self-service-password-reset” service neither: Instead, you see a message inviting you to pay:

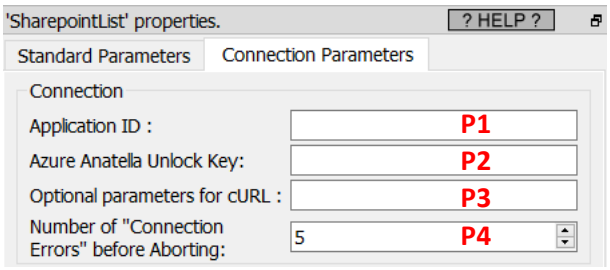


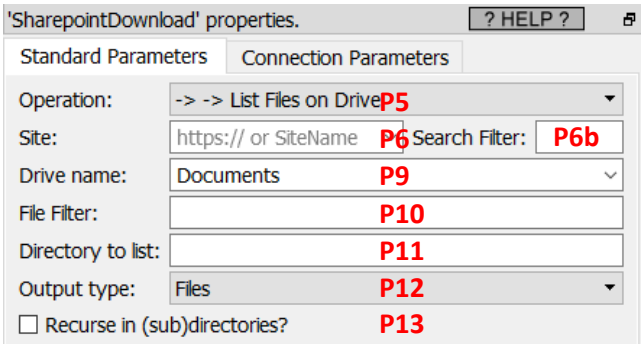
©2010 TIMI SPRL. E-mail: sales@timi.eu - visit us at www.timi.eu

5.23.28. SharePoint Download

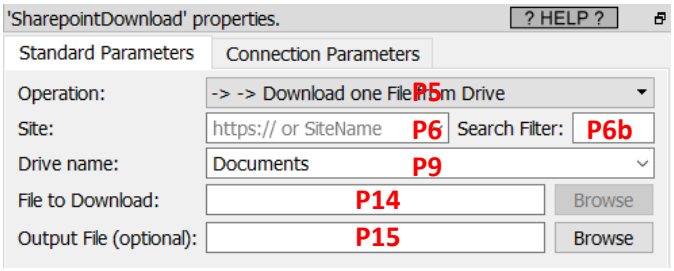
Icon:

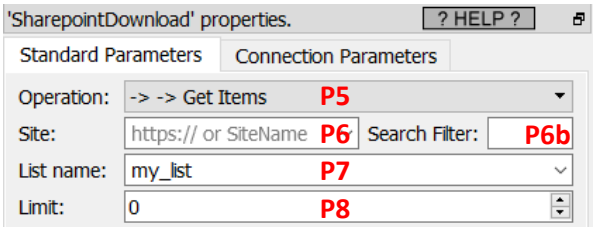
Property window:





Short description:
Download data from Microsoft SharePoint





Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P3** for web-access through a PROXY server.

This action currently only supports Microsoft Sharepoint **on Azure**.

To use this Action, you'll first need to get 2 parameters from the Azure Website: i.e. You need to obtain your "Application ID" (parameter **P1**) and your "Anatella Unlock Key" (parameter **P2**) from the Azure Website. Please see the section 5.23.27. for more details on how get these 2 parameters.

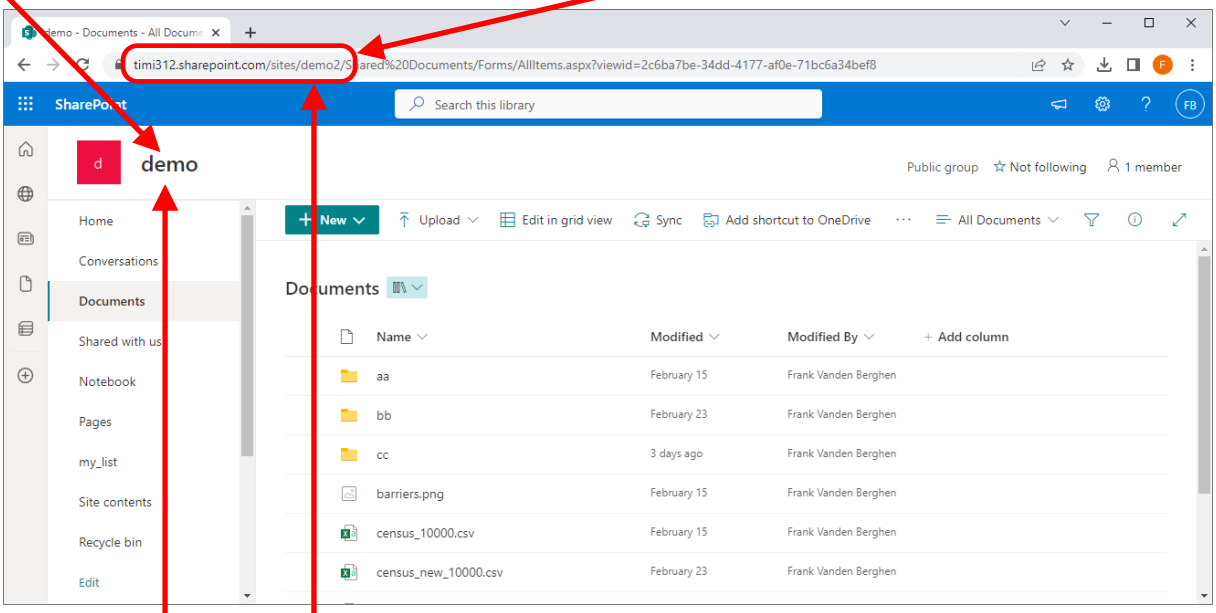
Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P5** to **P15** to download data from your Microsoft SharePoint system.

If this Action does not show you a "Site" or a "List" that you just defined inside SharePoint, then you just need wait a little. That's just Azure that is a little bit slow. It will appear, eventually.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameter **P6** is the Sharepoint Site that you want to access. This Site can be defined…

1. … using the "Site Display Name": It's visible here:
2. … using the Sharepoint Site URL: it's visible here:



The "Site Display Name" is: demo

The Sharepoint Site URL is: https://timi312.sharepoint.com/sites/demo2

The parameter **P6** is a combobox: It means that you can click the Drop-Down arrow at the right-side of the control to see a list of all the Sharepoint Sites that are accessible to the current user (more precisely: you see the list of "Display Names" of all the accessible sites). Sometime, when you have a very long list of "Sites", Azure has some difficulties to compute to whole list of Sites and you only see a partial result: i.e. you get a truncated list. To get around this bug in Azure, there are two different options inside Anatella:

1. You can use the parameter **P6b** to limit the length of the list of Sites returned by Azure to a smaller size. The list becomes smaller and should now contains the Site that you want to access.
2. You can enter the "Sharepoint Site URL" (instead of the "Site Display Name"). This URL typically looks like this: https://<your_site_host>.sharepoint.com/sites/<your site id name>
   This URL is visible here:

Inside a sharepoint server, you can manipulate two different types of assets:

1. "Items" (inside a web "List" inside a "Site")
2. "Files" (inside a web "Drive" inside a "Site")

The parameter **P5** allows you to select what type of assets you want to extract out of sharepoint.

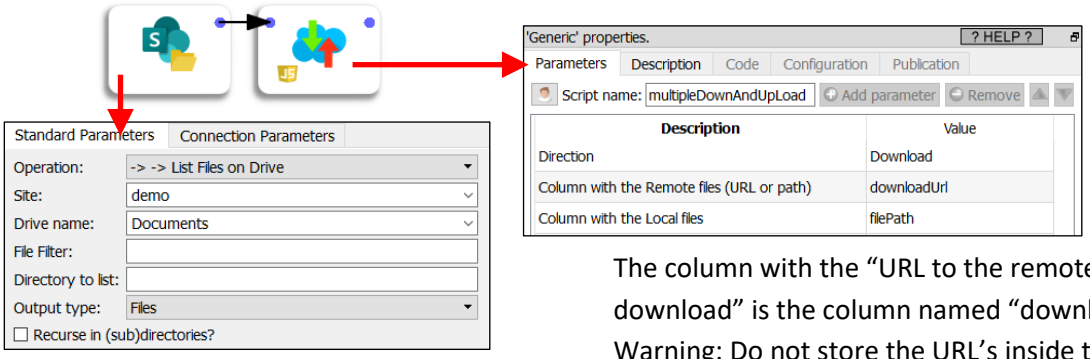
The available options for the parameter **P5** are:

1. Get Sites: returns a table with all the Sites inside your sharepoint server
2. *ITEM management:*
   1.1 Get List: return the "Lists" available inside the selected "Site" (parameter **P6**) on your sharepoint server
   1.2 Get Items: return the "Items" inside the selected "List" (parameter **P7**) inside the selected "Site" (parameter **P6**) on your sharepoint server

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. *FILE management:*
   12.1. Get Drives: return the "Drives" available inside a specific "Site" (parameter **P6**) on your sharepoint server
   12.2. List Files: return the list of files inside the selected "Drive" (parameter **P9**) inside a specific "Site" (parameter **P6**) on your sharepoint server
   12.3. Download one file: self-explanatory

The last option for the parameter **P5** allows to easily download **one** file out of sharepoint. If you need to download **several** files out of sharepoint, you'll use the following two actions together:

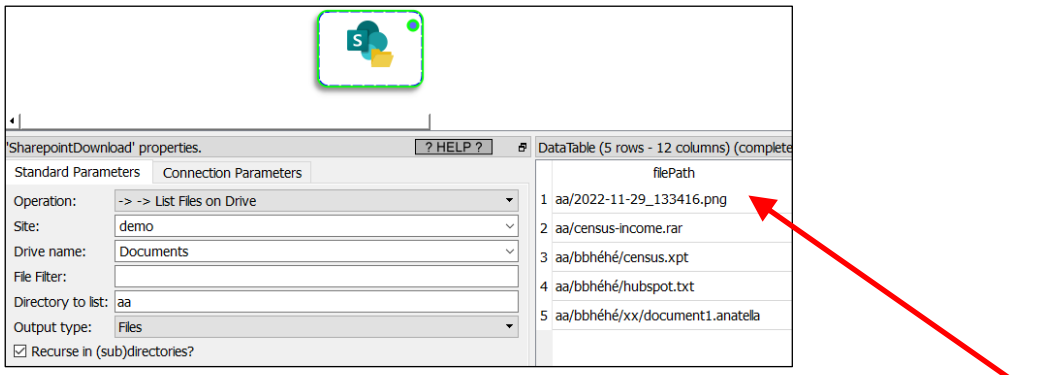


Use the parameters **P10** to **P13** to select the files to download

The column with the "URL to the remote file to download" is the column named "downloadUrl".
Warning: Do not store the URL's inside the "downloadUrl" column: they are only valid for 1h.

The directories do not have a "downloadURL": This is normal & expected.

If you want to download a whole directory, just list all the files inside your directory and then download each file using the multipleDownloadAndUpload Action.

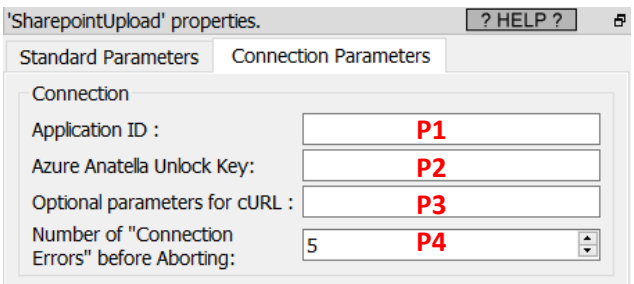


The value of the parameter **P14** ("File to Download") can be found inside the column "filePath" here:

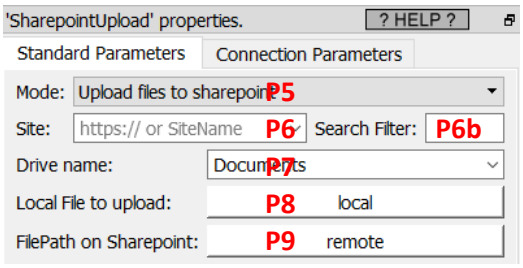
5.23.29. SharePoint Upload

Icon:

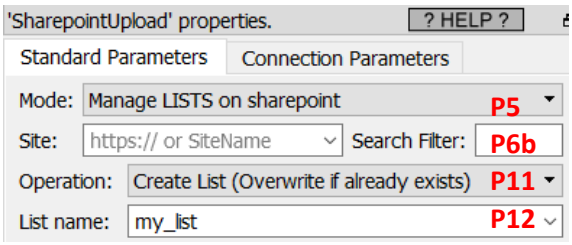
Property window:

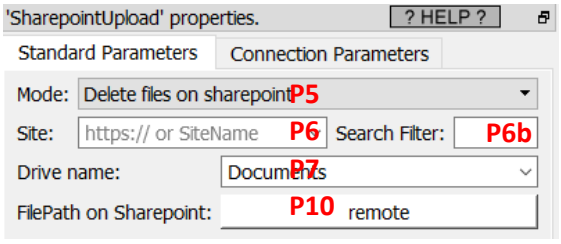


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Short description:
Upload data to Microsoft SharePoint





Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P3** for web-access through a PROXY server.

To use this Action, you'll first need to get 2 parameters from the Azure Website: i.e. You need to obtain your "Application ID" (parameter **P1**) and your "Anatella Unlock Key" (parameter **P2**) from the Azure Website. Please see the section 5.23.27. for more details on how get these 2 parameters.

Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P5** to **P12** to upload data to your Microsoft SharePoint system.

Please refer to the explanations from the previous section 5.23.28 to initialize properly the parameter **P6**. (i.e. the parameter **P6** can be the "Site Display name" or the "Sharepoint Site URL").

The parameter **P5** is the mode. It can be:

- Upload files on the sharepoint server
- Delete files on the sharepoint server
- Manage Lists on the sharepoint server

The first two modes (that are selected using the parameter **P5**) are self-explanatory (to upload/delete files on the Sharepoint server). The last mode (to manage "Lists") is slghtly more complex to operate and is described in the rest of this section.

The parameter **P10** defines the operation to perform on the selected sharepoint "LIST" (parameter **P11**): The available options are:

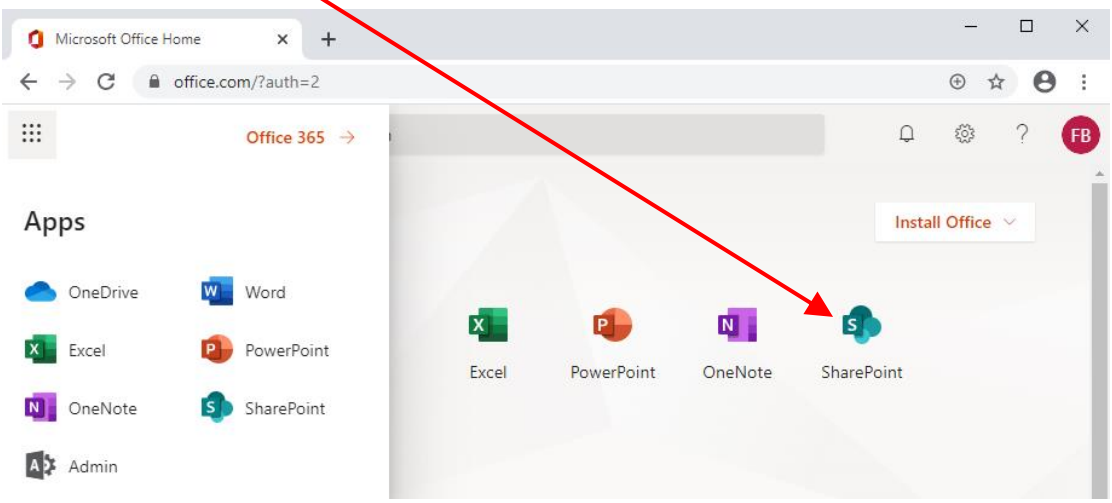
- Create List (Overwrite if already exists)
- Append to list
- Delete list
- Delete list and Append
- Update list

All these operating modes (with the exception of the "Delete list" operating mode) are expecting a table in input. This input table will be injected inside the given "List" inside the given Sharepoint site in Azure. Ideally, this input table should have a column named "Title" because Sharepoint is imposing that all Lists have a "Title" column. We are suggesting you to have a "Title" column in the table in input but you can also decide to ignore our recommendation: In this case, SharePoint will automatically add an empty "Title" column inside your list. Hopefully, you can easily hide this empty "Title" column in this way:

1. Open this URL: https://www.office.com and log-in into your Office account

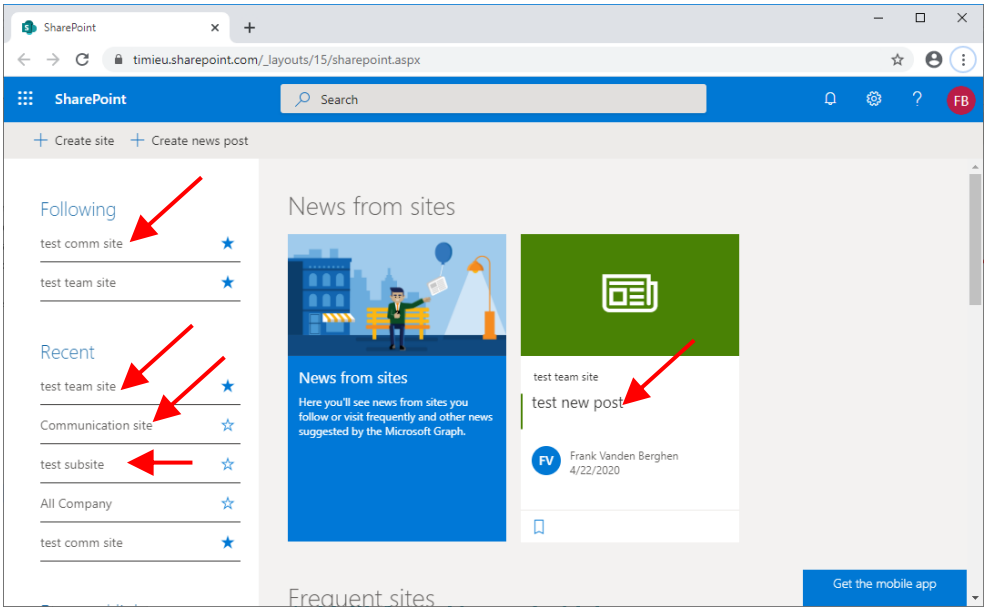
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Select “Sharepoint”:



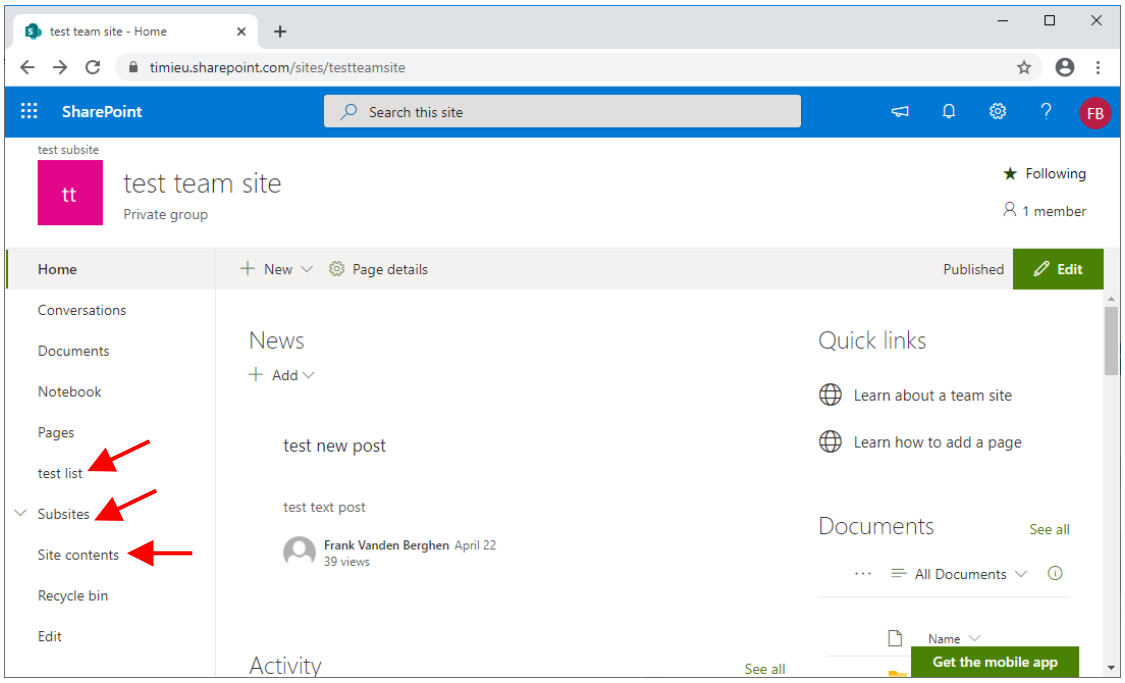
3. Navigate to the site and to the list to edit:

1.1. Select a site: Click here:

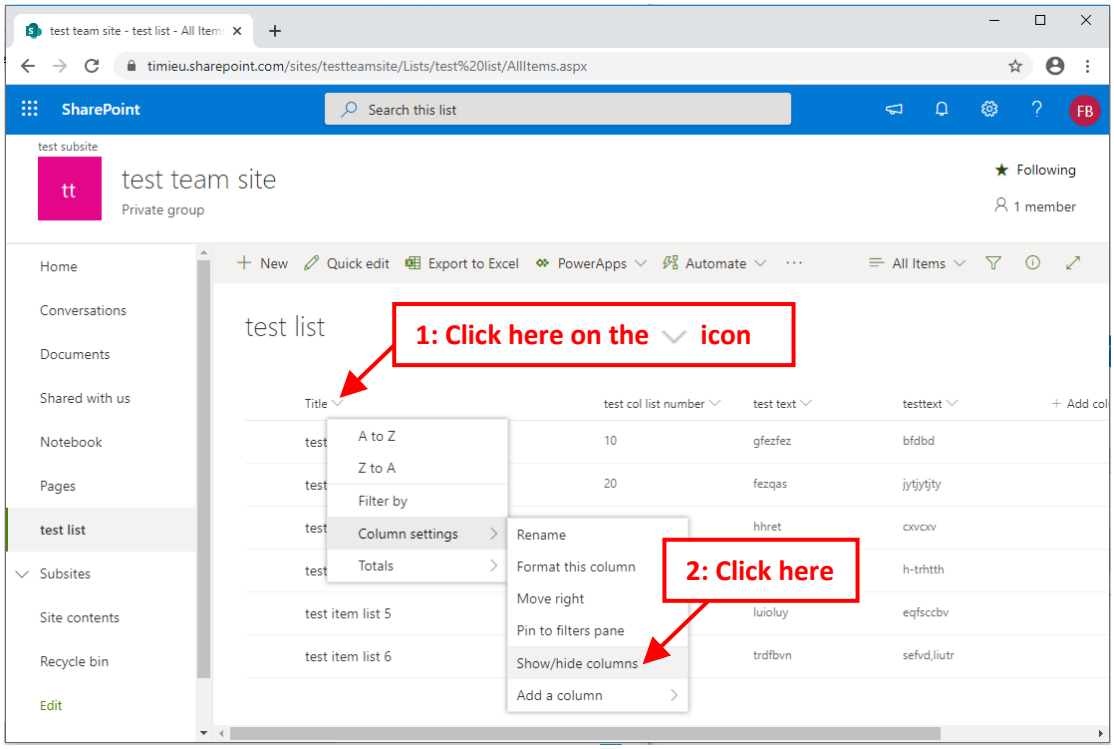


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1.2. Select a list to edit inside your site: Click here:

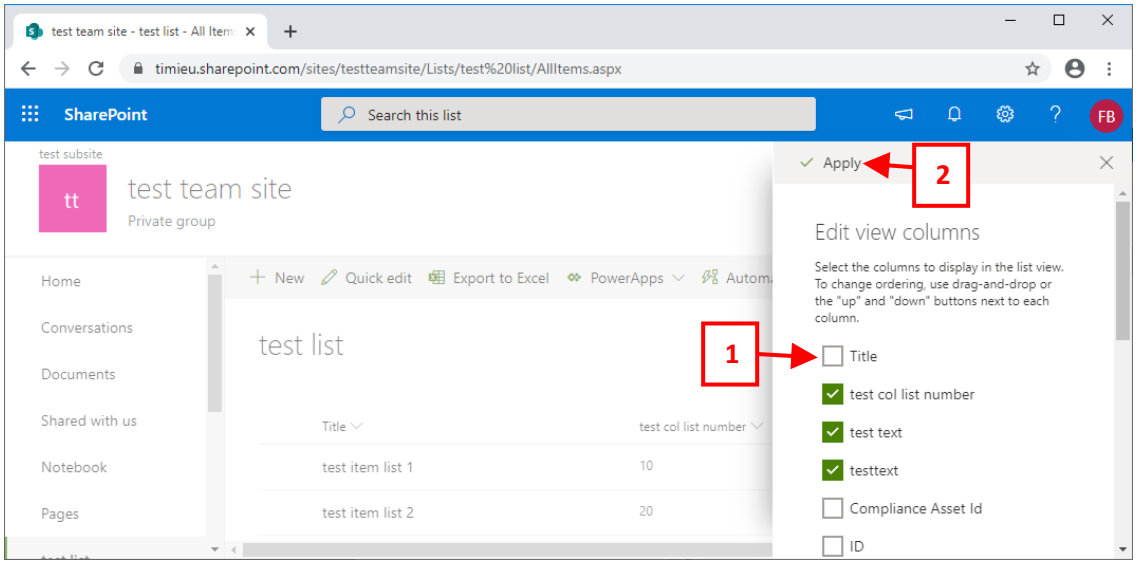


2. Click the icon next to the header of the column "Title": This opens a contextual menu: Navigate to the "Show/hide columns" Option:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Unchek the checkbox near “Title” and click the “Apply” button:



5.23.30. Azure Storage List Files

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: StorageListFile | Add parameter | Remove

Description	Value	
(Optional:) Folder path from which to get t...		P1
Container name	my-container-name	P2
Storage name	my-azure-storage-name	P3
Shared Key		P4
Debug Display?	No debug	P5
Optional parameters for cURL		P6
Number of "Connection Errors" before Ab...	3	P7

Short description:
List files in a container in an Azure Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

To use this Action, you’ll need to get several parameters from the Microsoft Website (i.e. you need the parameters **P2**, **P3** and **P4**). Please see the next section 5.23.30.1 for more details on how to get these parameters.

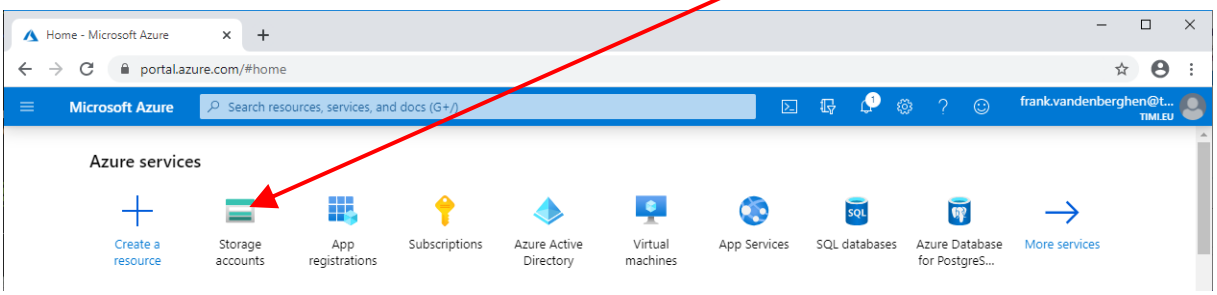
Once you have completed the “setup process” described in the section 5.23.30.1, you can use the parameter **P1** to list the files in your Azure storage system.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

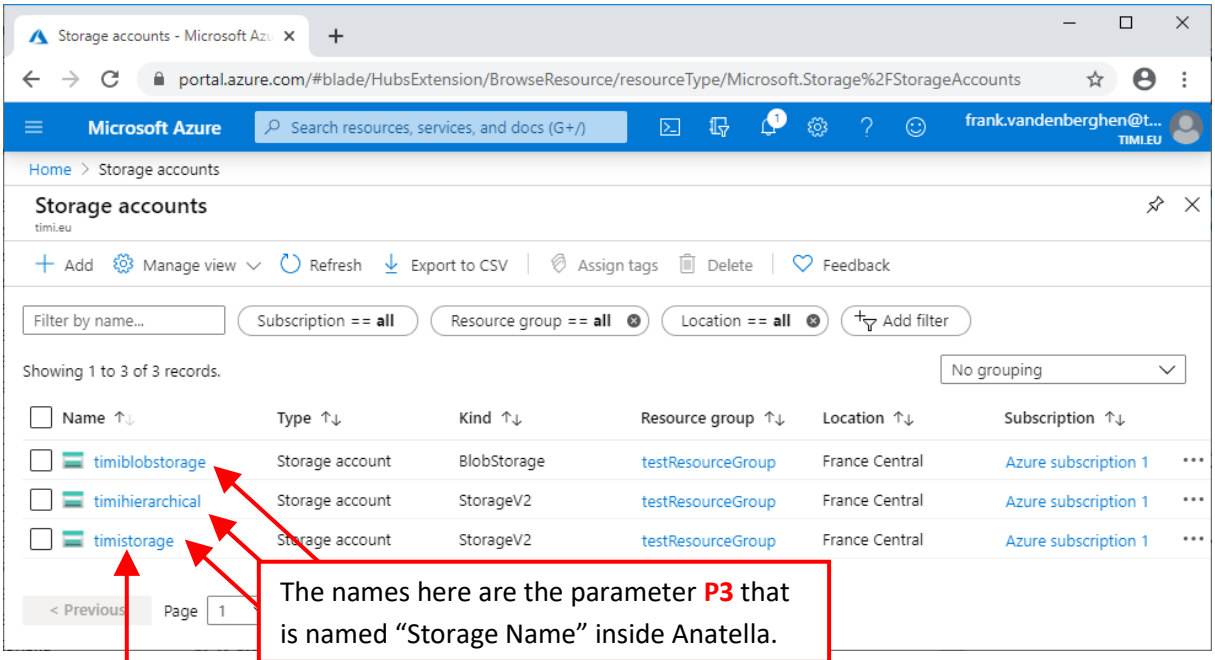
5.23.30.1. Azure Storage first time setup

Before using the Azure storage Actions inside Anatella, you need to get the parameters **P2**, **P3** and **P4** from the Azure website. Here are the steps to get these 3 parameters:

1. Open the URL https://portal.azure.com and "log-in" using your normal "Login" and "Password" for Azure. Then, click on the "Storage Accounts" icon:



2. You arrive on a page where you see all the available Azure Storage accounts. Here is an example:

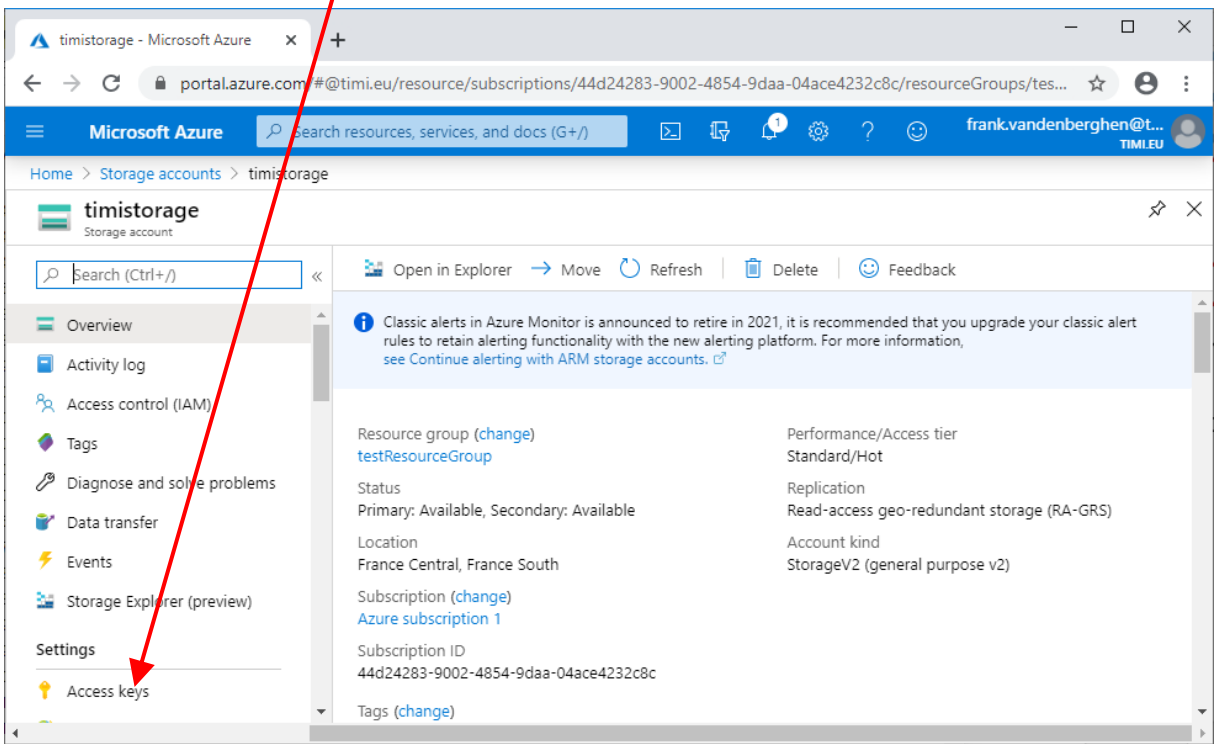


On may 2020, Anatella supports all types of Azure blob storage (including "StorageV2"!).

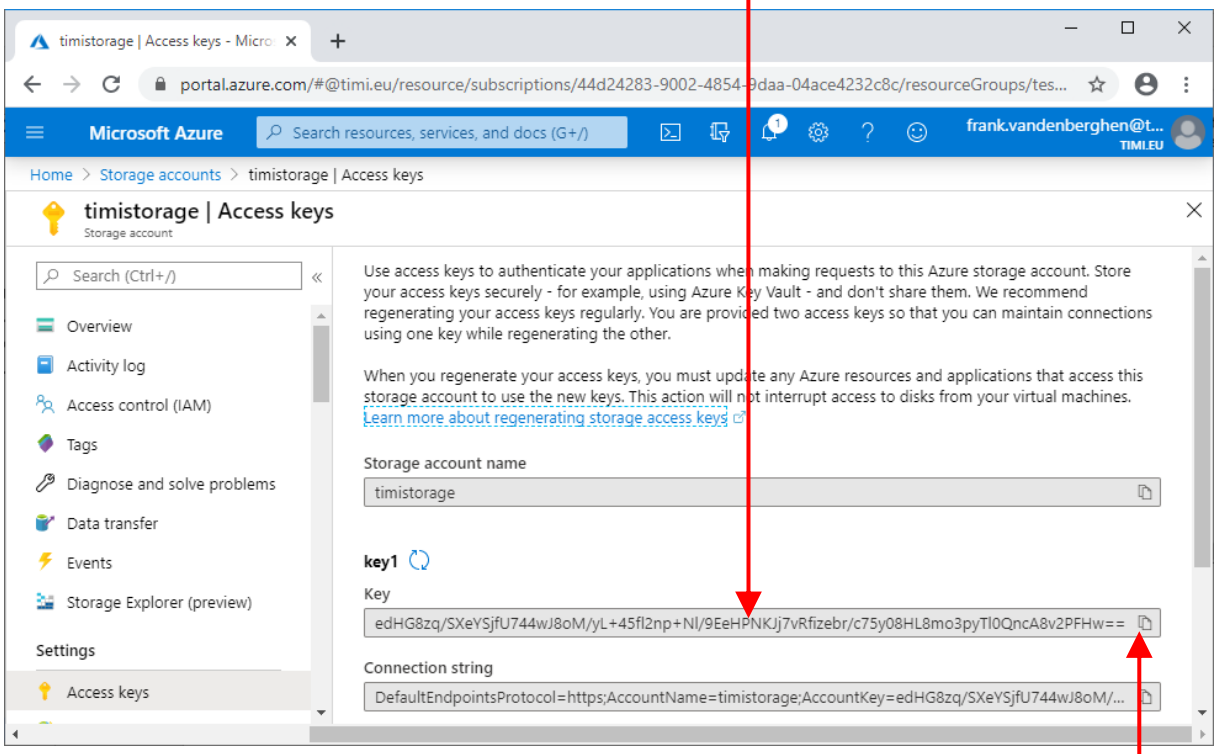
Let's now assume that we want to work on the Storage named "timistorage" (i.e. the parameter **P3** named "Storage Name" is "timistorage"): Let's click on "timistorage" here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Click on “Access Key” here:



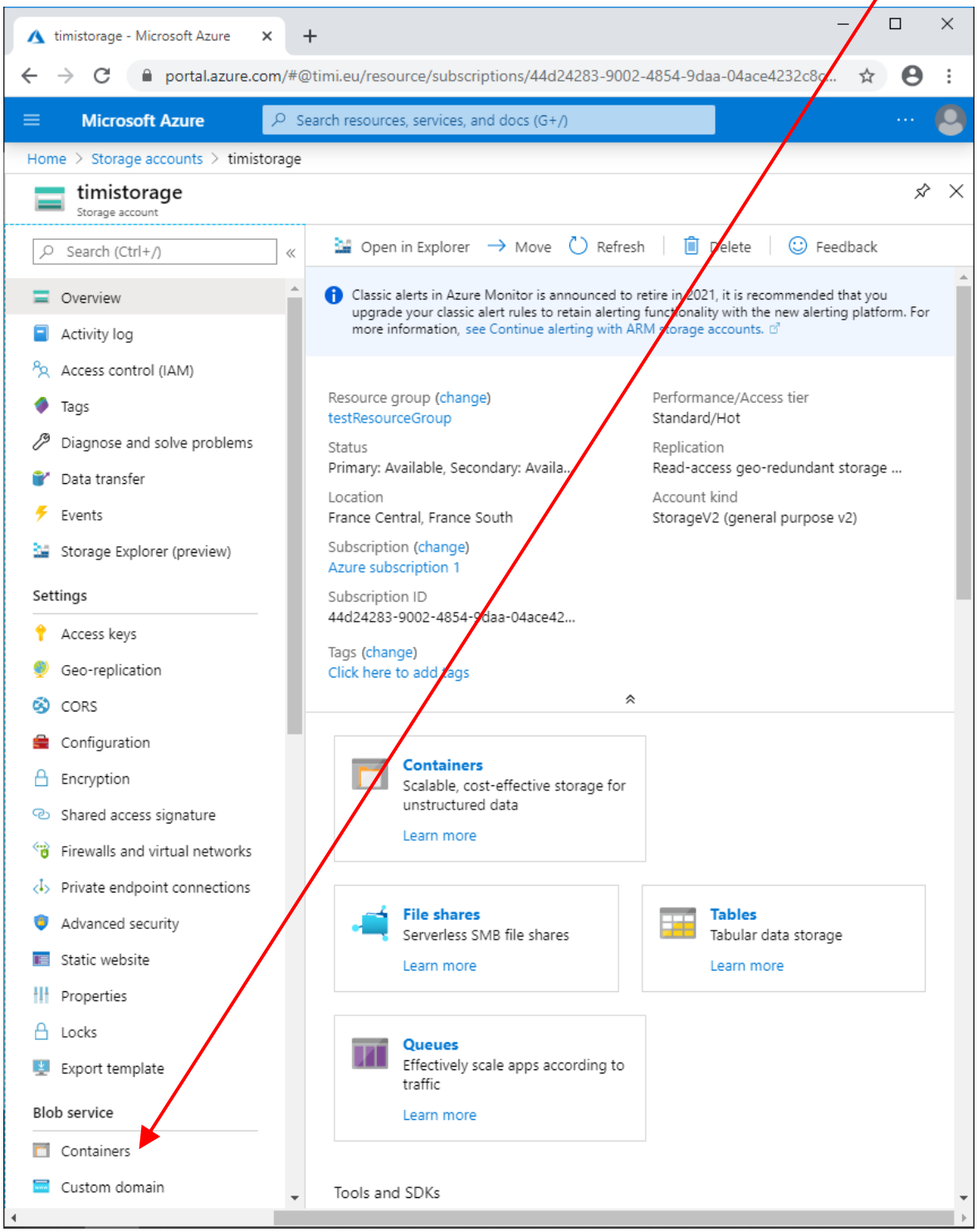
4. You get the Anatella parameter **P4** named “Shared Key” here:



Click here to copy (ctrl-c) your “Shared Key” inside the clipboard so that you can easily paste it (ctrl-v) inside Anatella.

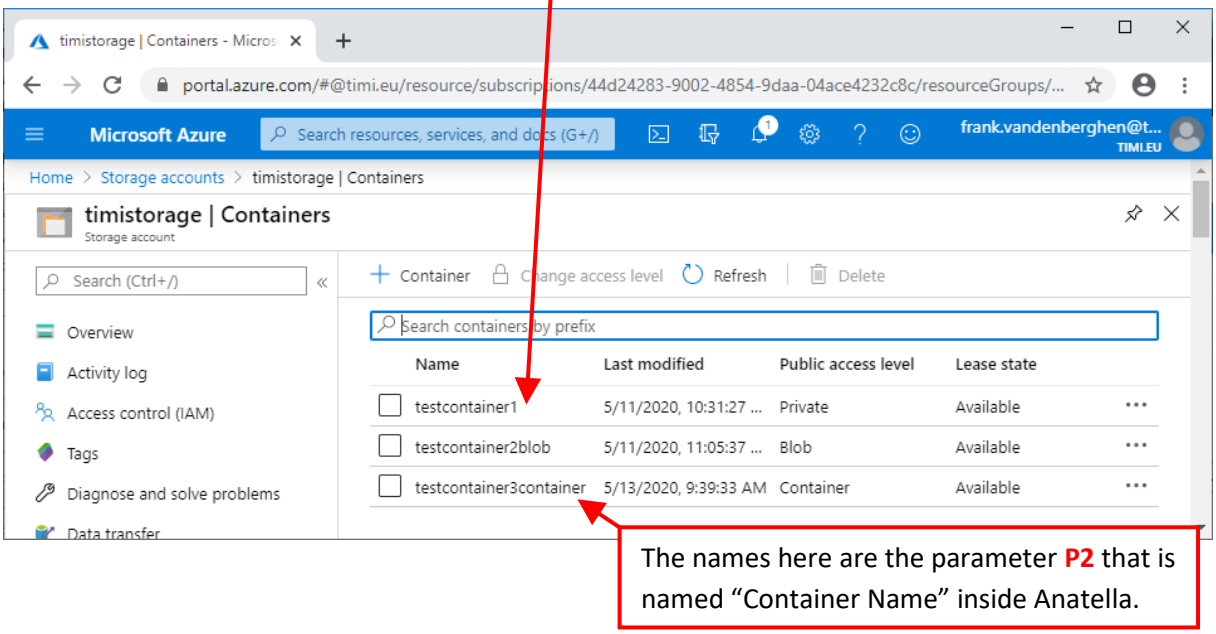
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5. To get the Anatella parameter **P2** (that is named "Container Name"), you now have two different options:
   1.1. You can use the AzureStorageListContainers action from section 5.23.32.
   1.2. You can use the Azure portal: Click on the "Containers" link inside the "Blob Service" section:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

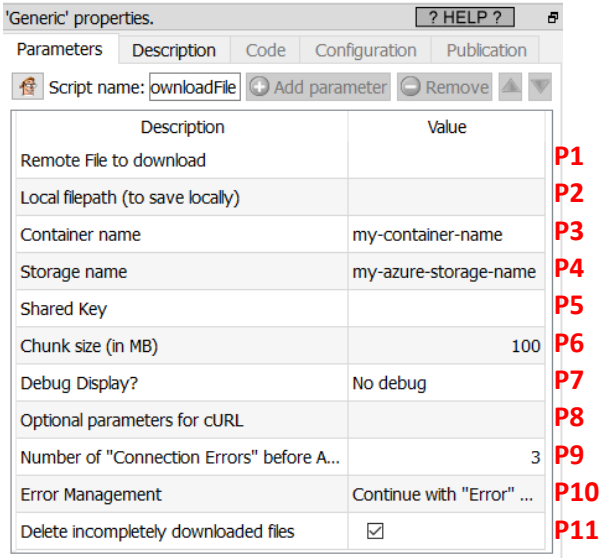
1.3. The available containers are listed here:



5.23.31. Azure Storage Download Files

Icon:

Property window:



Short description:
Download files from a container in an Azure Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Microsoft Website (parameters **P3**, **P4** and **P5**). Please see the section 5.23.30.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.30.1, you can use the parameters **P1** and **P2** to download the required files from your Azure storage system.

The parameter **P6** (i.e. the Chunk Size) is usefull when downloading very large files. The file is downloaded chunk-by-chunk. If one connection error happens during the download only the last chunk needs to be re-downloaded.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.32. Azure Storage Upload Files

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: eUploadFile | Add parameter | Remove

Description	Value	
File to upload		P1
(optional) Filename on remote server		P2
Container name	my-container-name	P3
Storage name	my-azure-storage-name	P4
Shared Key		P5
Chunk size (in MB - max 100MB)	100	P6
Debug Display?	No debug	P7
Optional parameters for cURL		P8
Number of "Connection Errors" before A...	3	P9
Error Management	Continue with "Error" ...	P10

Short description:
Upload files to a container in an Azure Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Microsoft Website (parameters **P3**, **P4** and **P5**). Please see the section 5.23.30.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.30.1, you can use the parameters **P1** and **P2** to upload the required files to your Azure storage system.

The parameter **P6** (i.e. the Chunk Size) is usefull when uploading very large files. The file is uploaded chunk-by-chunk. If one connection error happens during the upload only the last chunk needs to be re-uploaded.

5.23.33. Azure Storage Delete Files

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: geDeleteFile | Add parameter | Remove

Description	Value	
File to delete		P1
Container name	my-container-name	P2
Storage name	my-azure-storage-name	P3
Shared Key		P4
Debug Display?	No debug	P5
Optional parameters for cURL		P6
Number of "Connection Errors" before A...	3	P7
Error Management	Continue with "Error" ...	P8

Short description:
Delete files from a container in an Azure Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

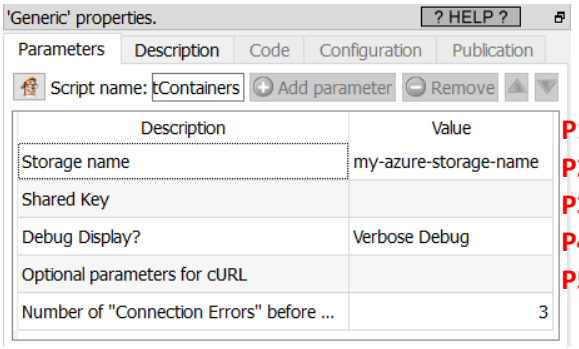
To use this Action, you'll need to get several parameters from the Microsoft Website (parameters **P2**, **P3** and **P4**). Please see the section 5.23.30.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.30.1, you can use the parameter **P1** to delete the required files in your Azure storage system.

5.23.34. Azure Storage List Containers

Icon:

Property window:



Short description:
List containers in an Azure Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

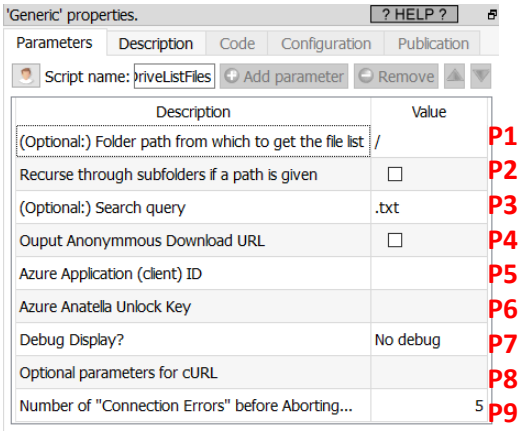
To use this Action, you'll need to get several parameters from the Microsoft Website (parameters **P1** and **P2**). Please see the section 5.23.30.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.30.1, you can use this action to list the available containers in your Azure storage system.

5.23.35. OneDrive List Files

Icon:

Property window:



Short description:
List the files in your OneDrive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get 2 parameters from the Azure Website (i.e. you need need your "Application ID" -parameter **P5**- and your "Anatella Unlock Key" -parameter **P6**). Please see the section 5.23.27. for more details on how to get these parameters.

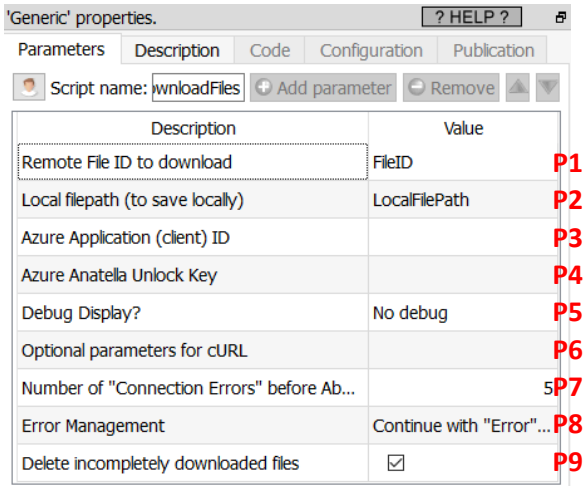
Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P1**, **P2**, **P3**, **P4** to list the files in your OneDrive.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.36. OneDrive Download Files

Icon:

Property window:



Short description:
Download files from a container in your Onedrive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

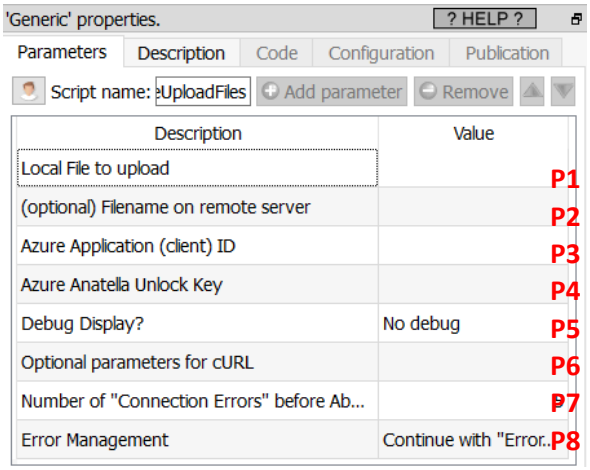
To use this Action, you'll need to get 2 parameters from the Azure Website (i.e. you need the your "Application ID" -parameter **P3**- and your "Anatella Unlock Key" -parameter **P4**). Please see the section 5.23.27. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P1** and **P2** to download the required files from your OneDrive.

5.23.37. OneDrive Upload Files

Icon:

Property window:



Short description:
Upload files to a container in your Onedrive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

To use this Action, you'll need to get 2 parameters from the Azure Website (i.e. you need the your "Application ID" -parameter **P3**- and your "Anatella Unlock Key" -parameter **P4**). Please see the section 5.23.27. for more details on how to get these parameters.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

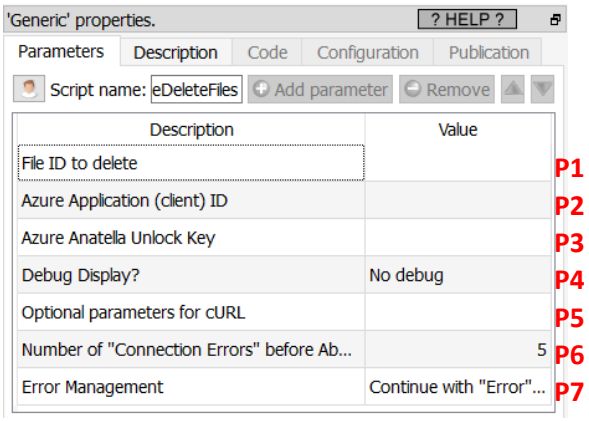
Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P1** and **P2** to upload the required files to your OneDrive.

The parameter **P6** (i.e. the Chunk Size) is usefull when uploading very large files. The file is uploaded chunk-by-chunk. If one connection error happens during the upload only the last chunk needs to be re-uploaded again.

5.23.38. OneDrive Delete Files

Icon:

Property window:



Short description:
Delete files from a container in your Onedrive

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

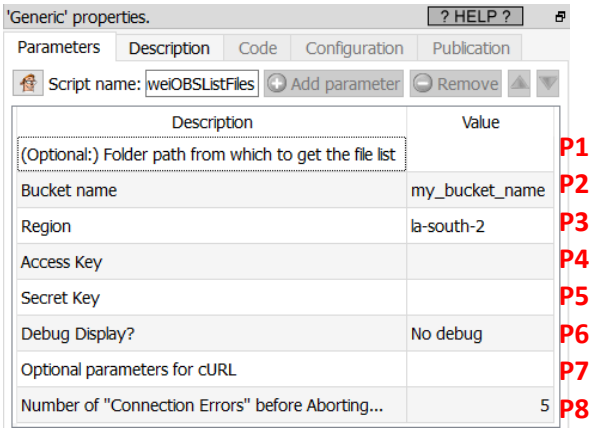
To use this Action, you'll need to get 2 parameters from the Azure Website (i.e. you need the your "Application ID" -parameter **P2**- and your "Anatella Unlock Key" -parameter **P3**). Please see the section 5.23.27. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.27., you can use the parameter **P1** to delete the required files inside your OneDrive.

5.23.39. Huawei Blob Storage List Files

Icon:

Property window:



Short description:
List files in a bucket in an Huwaei Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To use this Action, you'll need to get several parameters from the Huawei Website (i.e. you need the parameters **P2**, **P3**, **P4** and **P5**). Please see the next section 5.23.39.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.39.1, you can use the parameter **P1** to list the files in your Huwaei Blob Storage system.

5.23.39.1. Huawei Blob Storage First Time Setup

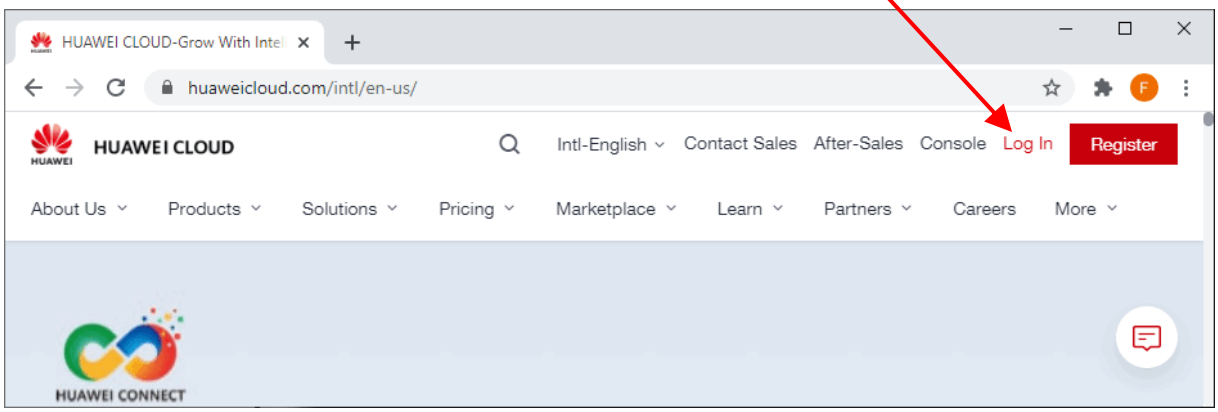
Before using the Huawei Blob storage Actions inside Anatella, you need to get these 4 parameters:

- Anatella Parameter **P1**: Region
- Anatella Parameter **P2**: Bucket Name
- Anatella Parameter **P3**: Access-Key
- Anatella Parameter **P4**: Secret-Key

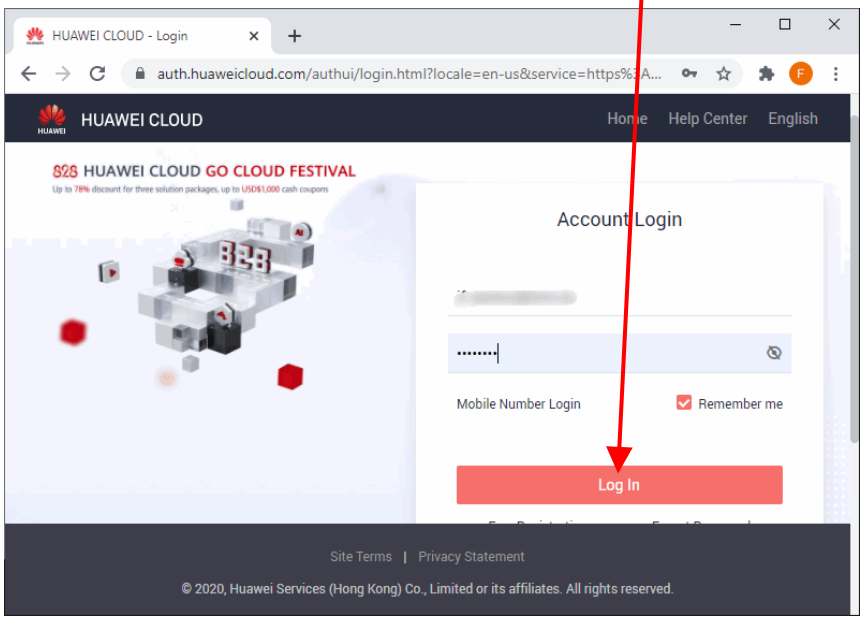
…from the Huwaei website.

Here are the steps to get these 4 parameters:

1. Open https://www.huaweicloud.com and click on the "Log in" button:

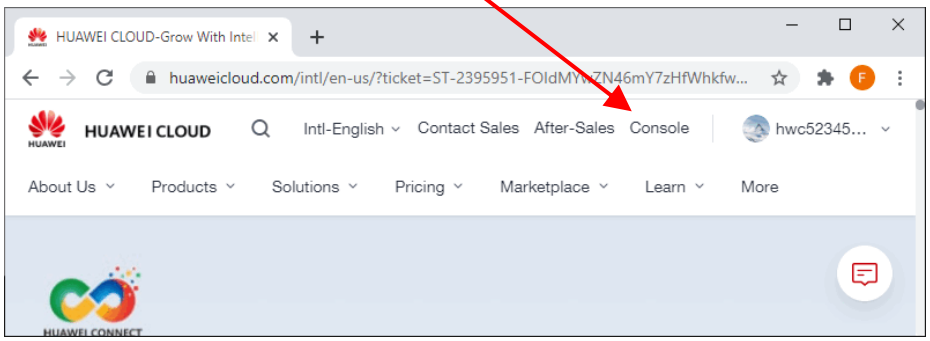


2. Enter your credentials and click the "Log In" button:

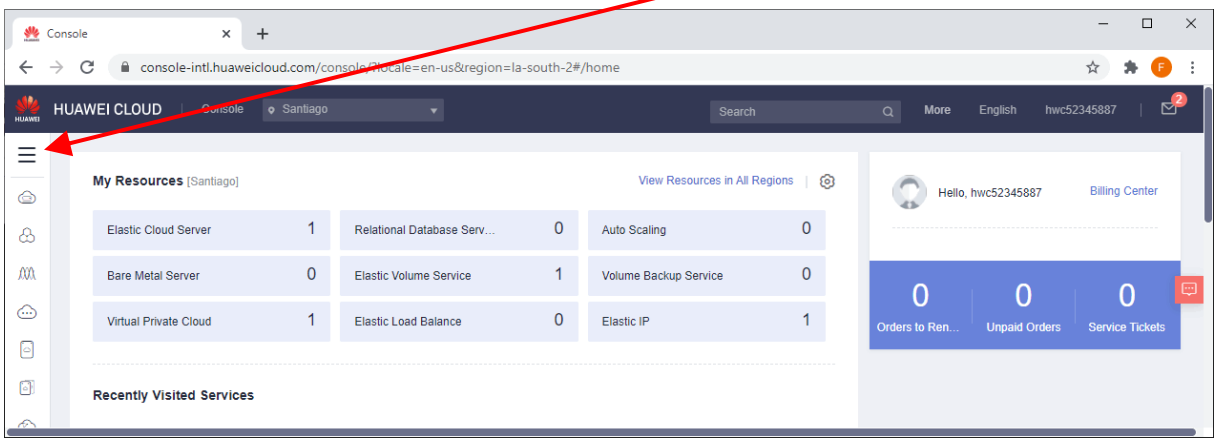


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

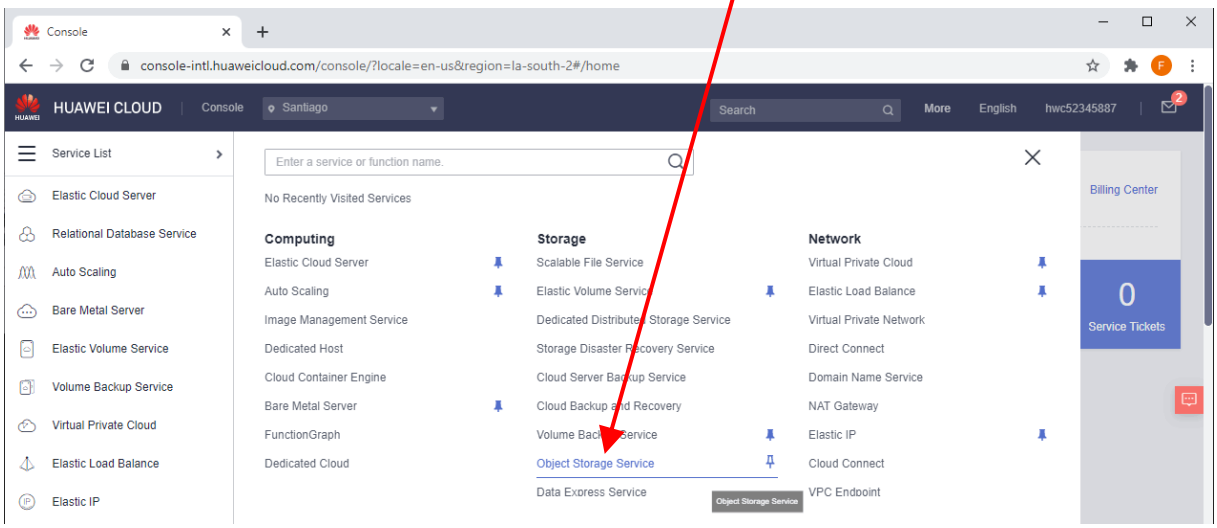
3. Click on the “console” link inside the header:



4. Click on the ☰ symbol at the top of the column on the left:

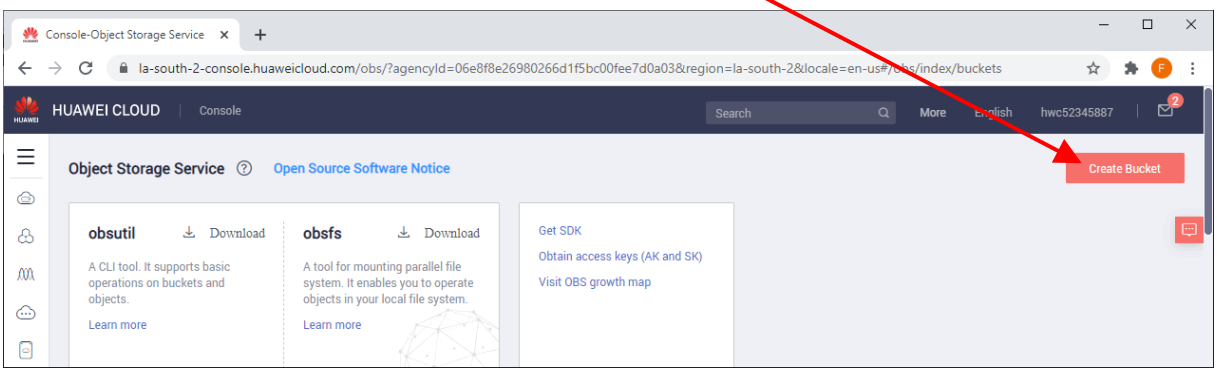


5. …and select (click) “Object Storage Service” in the big menu:

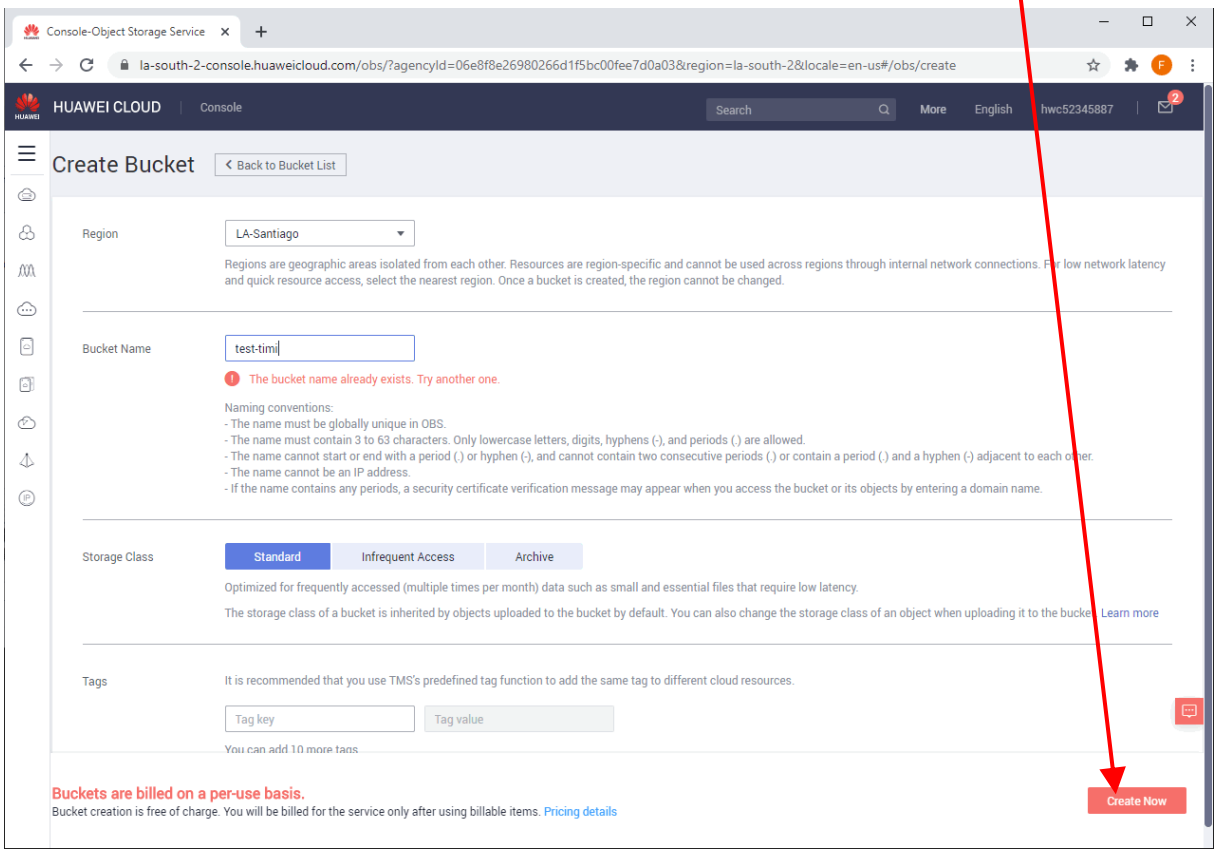


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. Click on the “Create Bucket” button on the top right:

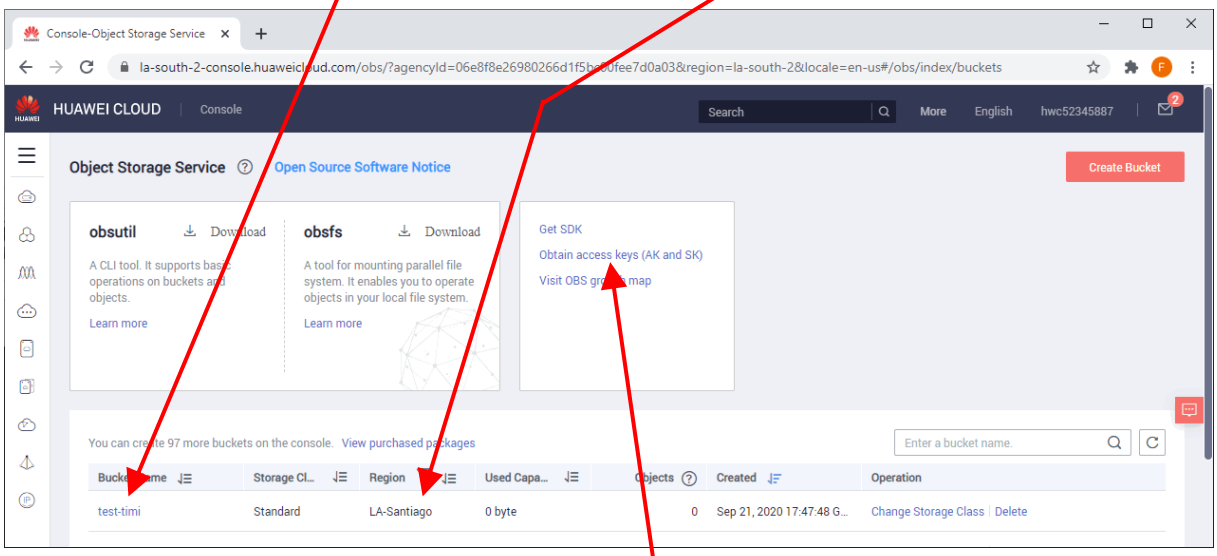


7. Fill-in all the required fields on click on the “Create Now” button on the bottom:



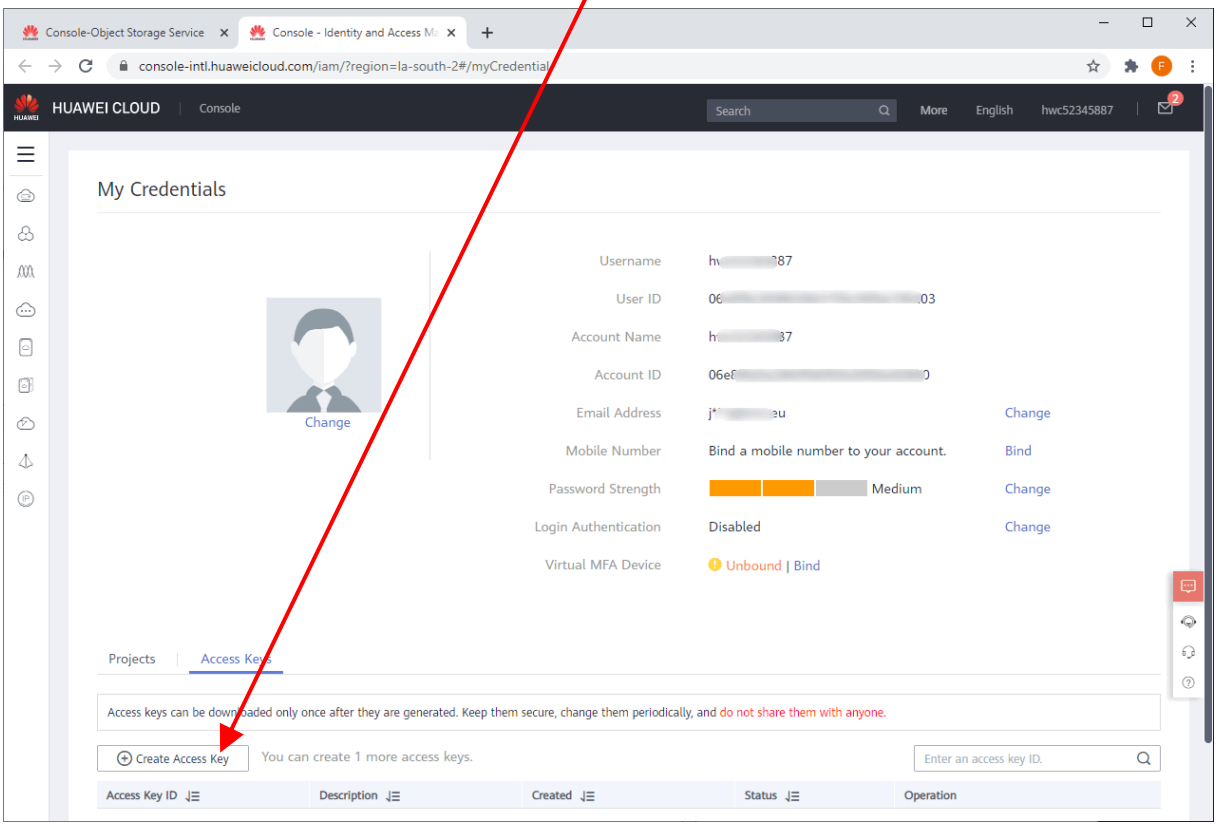
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

8. You receive the Anatella Parameter **P2** (Bucket Name) and **P1** (Region):



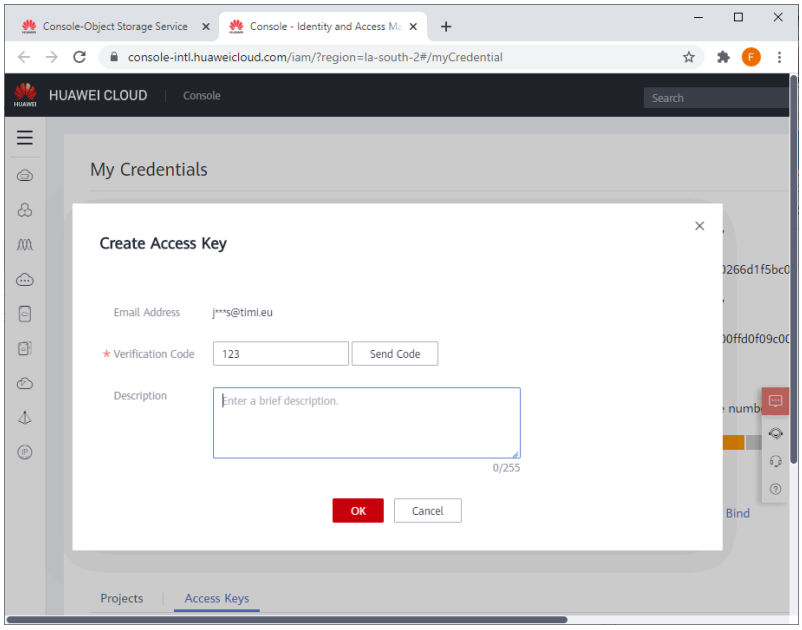
9. Click on the “Obtain access keys (AK and SK)” url here:

10. Click on the button “Create Access Key”:

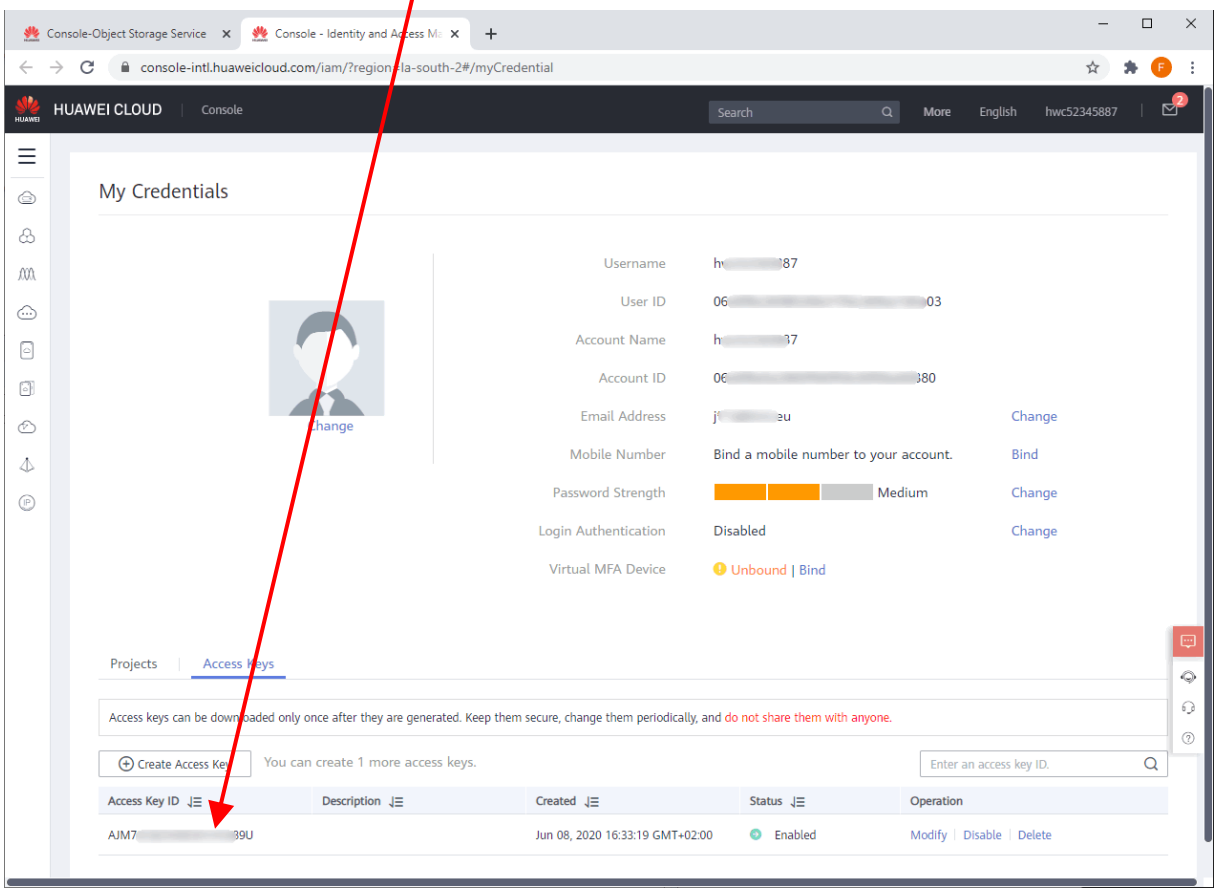


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

11. Fill-in the next screen:



12. You receive the Anatella Parameter **P3** (Access-Key) and **P4** (Secret-Key):



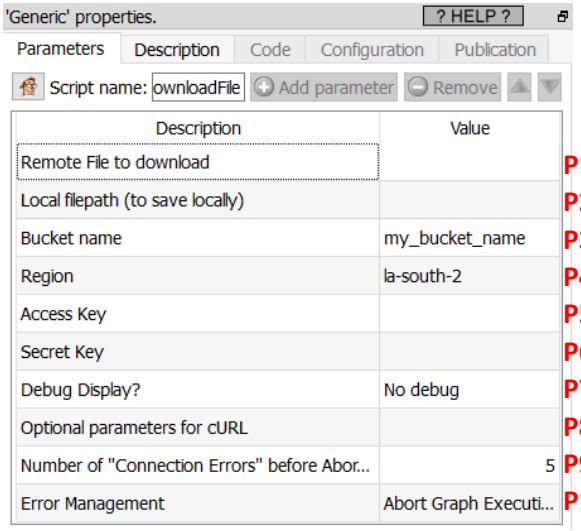
Please make sure to keep the Anatella Parameter **P4** (Secret-Key) is a safe place since it cannot be retreived from the Huawei website later on.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.40. Huawei Blob Storage Download Files

Icon:

Property window:



Short description:
Download files from a Huwaei Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

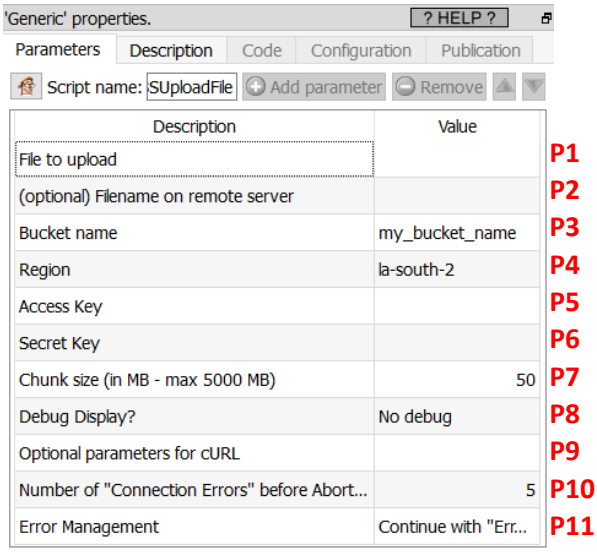
To use this Action, you'll need to get several parameters from the Huawei Website (i.e. you need the parameters **P3**, **P4**, **P5**, and **P6**). Please see the section 5.23.39.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.39.1, you can use the parameter **P1** and **P2** to download files from your Huwaei Blob Storage system.

5.23.41. Huawei Blob Storage Upload Files

Icon:

Property window:



Short description:
Upload files to a Huwaei Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Huawei Website (i.e. you need the parameters **P3**, **P4**, **P5**, and **P6**). Please see the section 5.23.39.1 for more details on how to get these parameters.

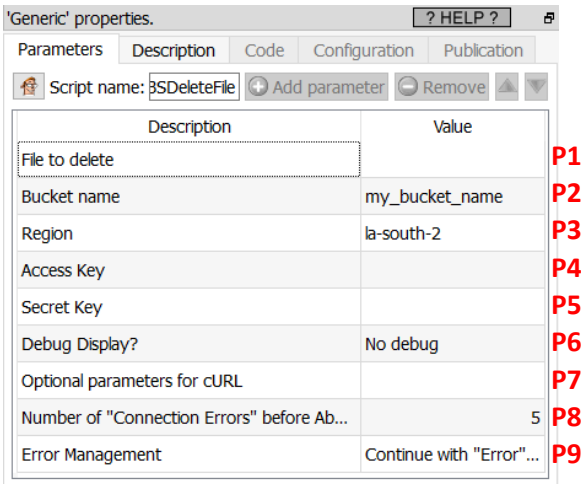
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Once you have completed the "setup process" described in the section 5.23.39.1, you can use the parameter **P1** and **P2** to upload files to your Huwaei Blob Storage system.

5.23.42. Huawei Blob Storage Delete Files

Icon:

Property window:



Short description:
Delete files stored in a Huwaei Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

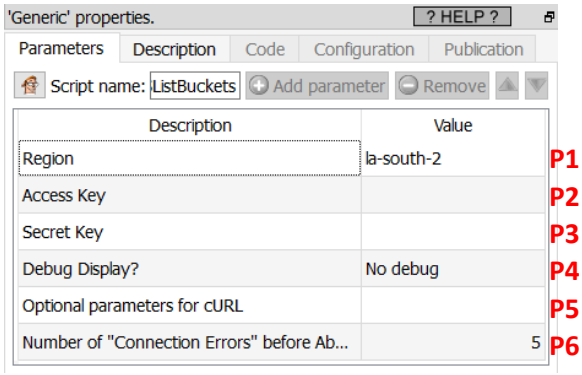
To use this Action, you'll need to get several parameters from the Huawei Website (i.e. you need the parameters **P2**, **P3**, **P4** and **P5**). Please see the section 5.23.39.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.39.1, you can use the parameter **P1** to delete files stored in your Huwaei Blob Storage system.

5.23.43. Huawei Blob Storage List Buckets

Icon:

Property window:



Short description:
List the buckets in a Huwaei Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Huawei Website (i.e. you need the parameters **P1**, **P2** and **P3**). Please see the section 5.23.39.1 for more details on how to get these parameters.

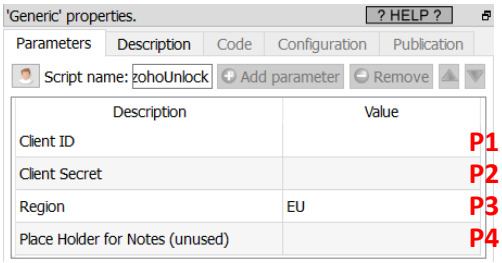
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Once you have completed the "setup process" described in the section 5.23.39.1, you can use this action to list the buckets from your Huwaei Blob Storage system.

5.23.44. Unlocks ZohoCRM

Icon:

Property window:



Short description:
Unlocks access to Zoho CRM

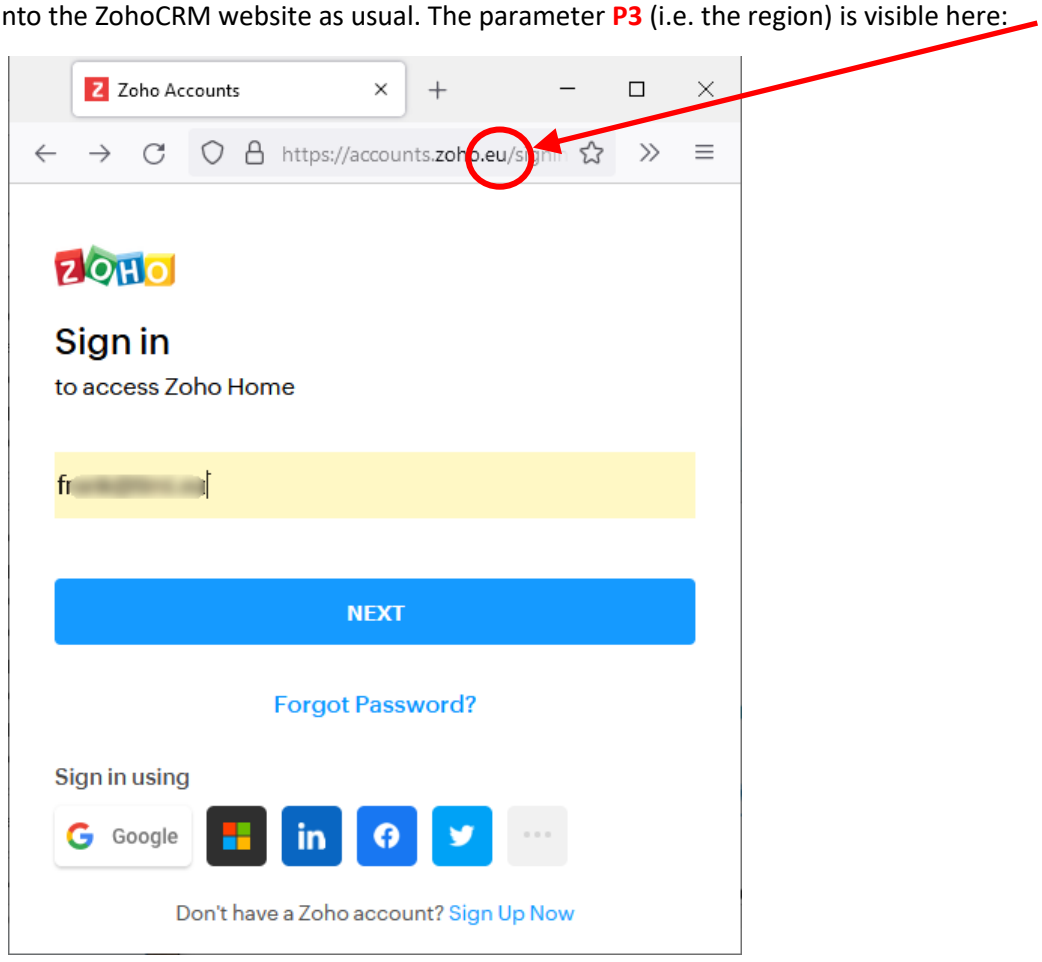
Long Description:
Before downloading data from Zoho CRM with Anatella, you need to get the 4 parameters:

- Client ID (Parameter **P1**)
- Cient Secret (Parameter **P2**)
- Region (Parameter **P3**)
- Refresh Token: You get this parameter using the current zohoUnlock action.

…from the ZohoCRM website.

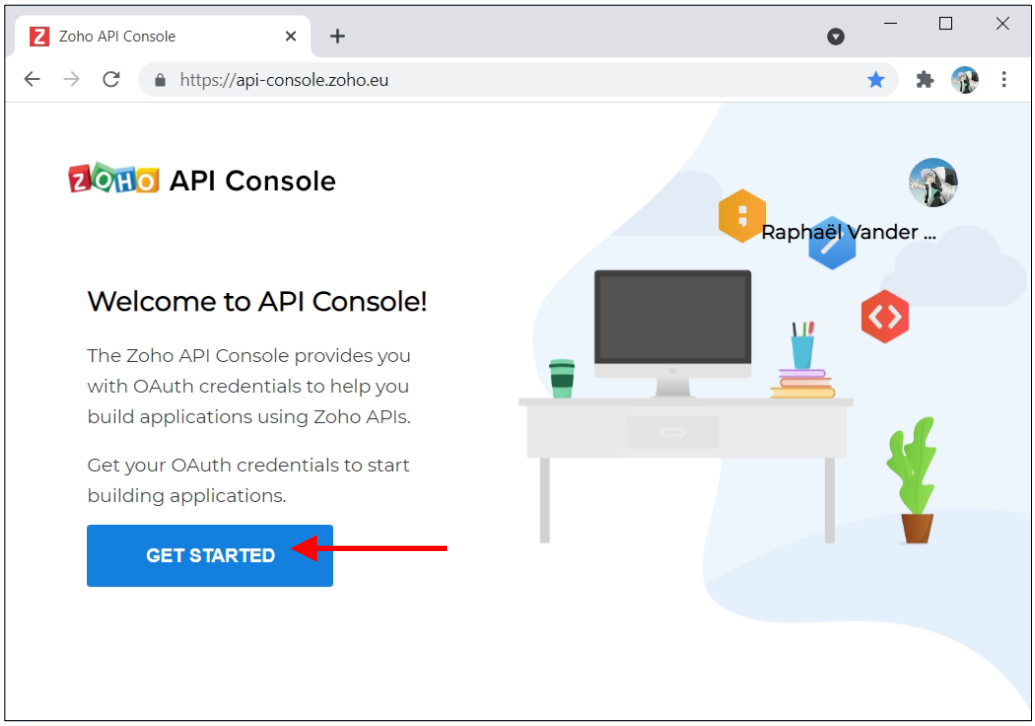
Here are the steps to get these 4 parameters:

1. Log-in into the ZohoCRM website as usual. The parameter **P3** (i.e. the region) is visible here:

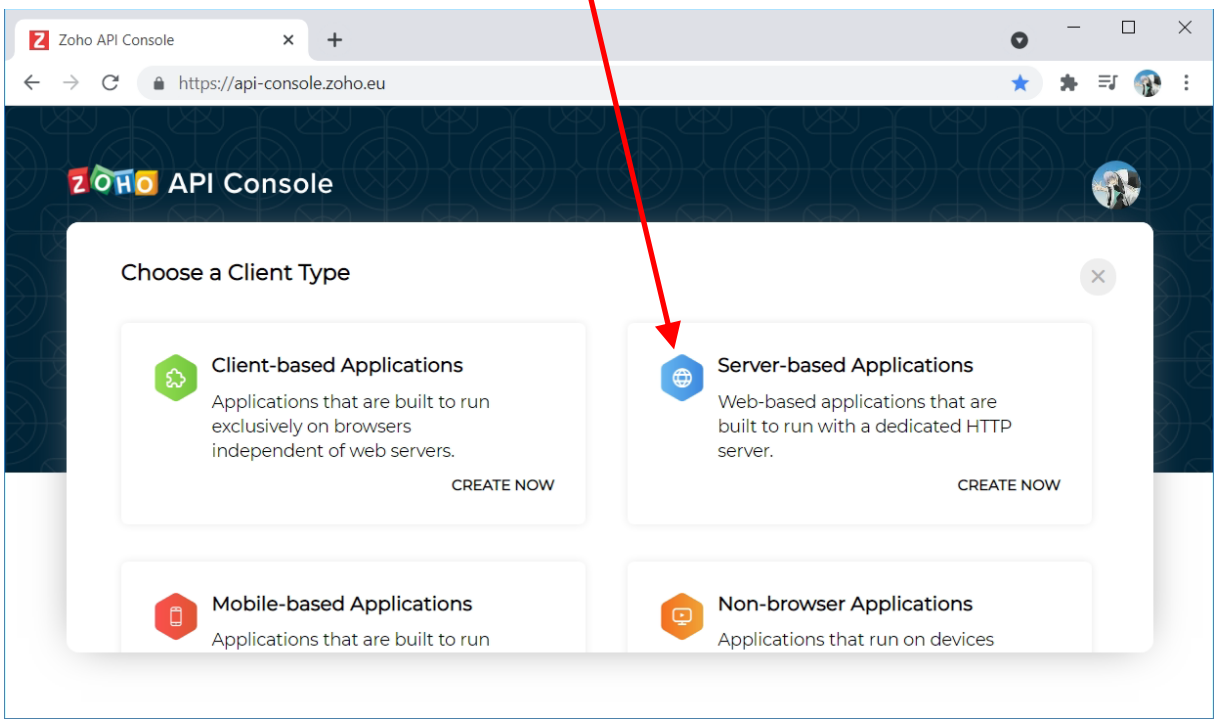


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Open the URL: https://api-console.zoho.eu/ (replace the ".eu" url-extention with your own region) and click on the "GET STARTED" button:



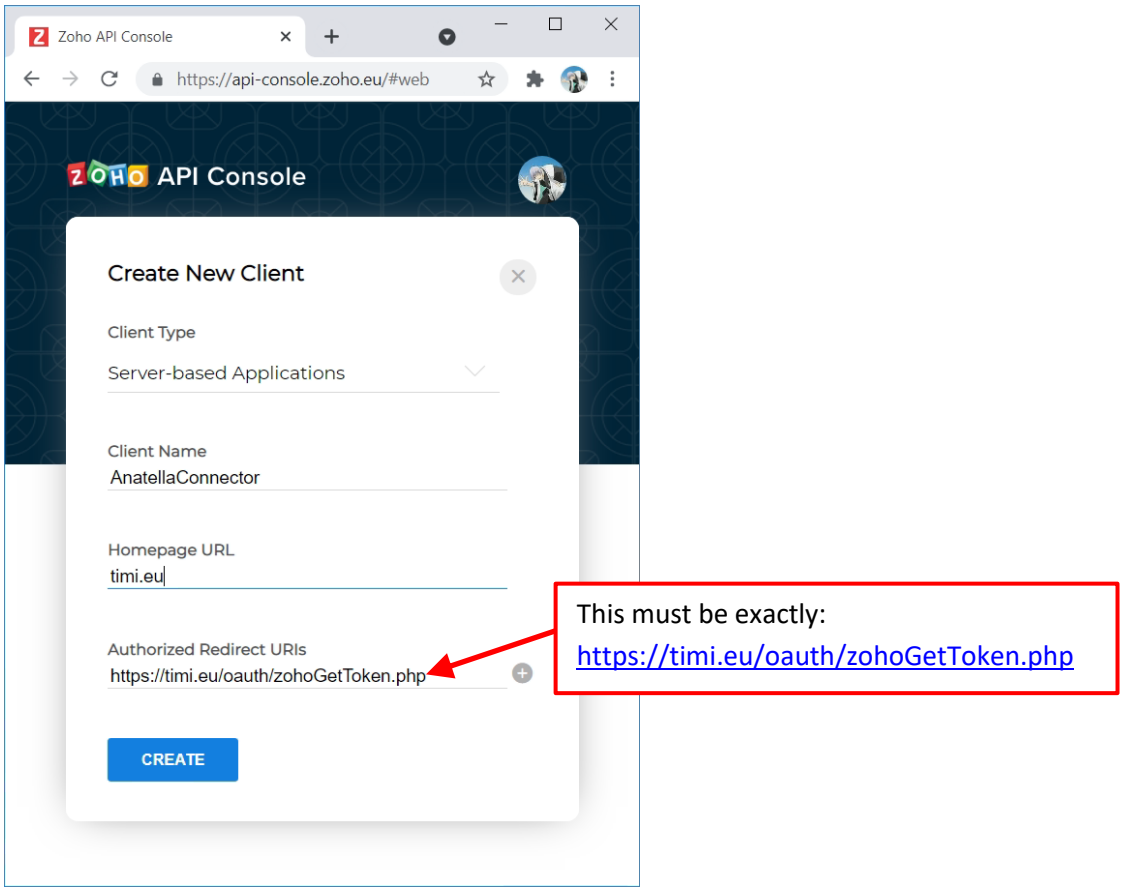
3. Click on the "Server based Applications" option:



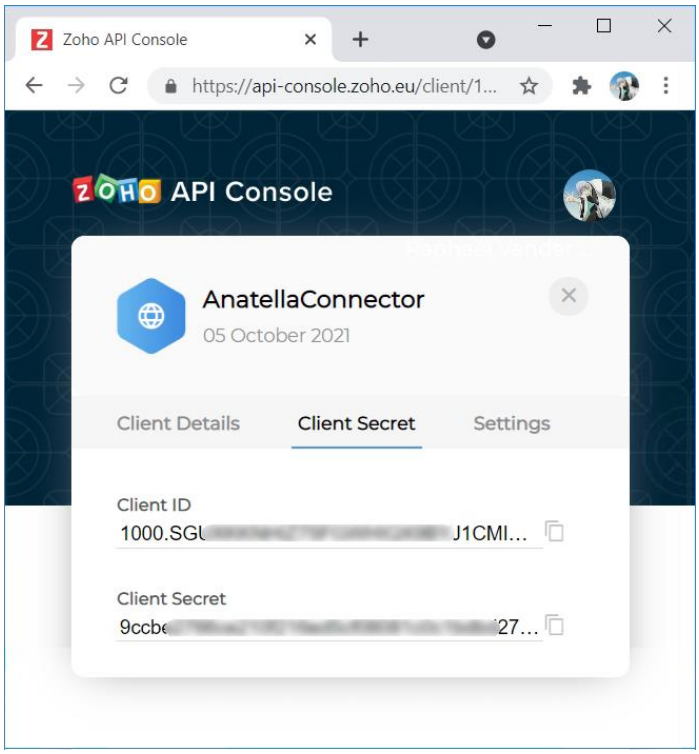
4. Fill-in the next form. You can use any value for the "Client Name" or the "Homepage url". The parameter named "Authorized Redirect URIs" must be:

https://timi.eu/oauth/zohoGetToken.php

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

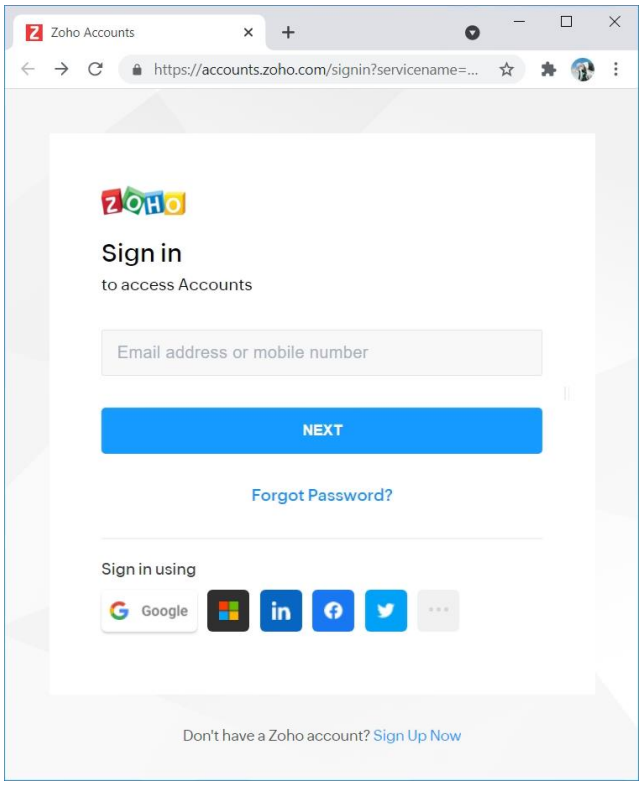


5. Click on the “Client Secret” menu.
   You get your “Client ID” (parameter **P1**) and your “Client Secret” (parameter **P2**) here:

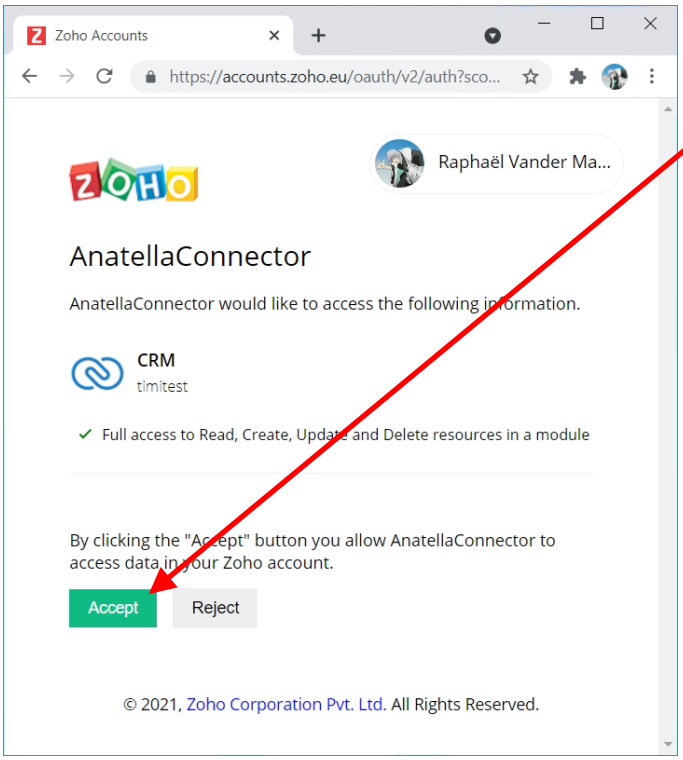


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. Copy/paste your newly found “Client ID” and “Client Secret” inside the parameters **P1**&**P2** of the zohoUnlock action inside Anatella and run the Action. Your browser opens. Login-in as usual inside your ZohoCRM Account:

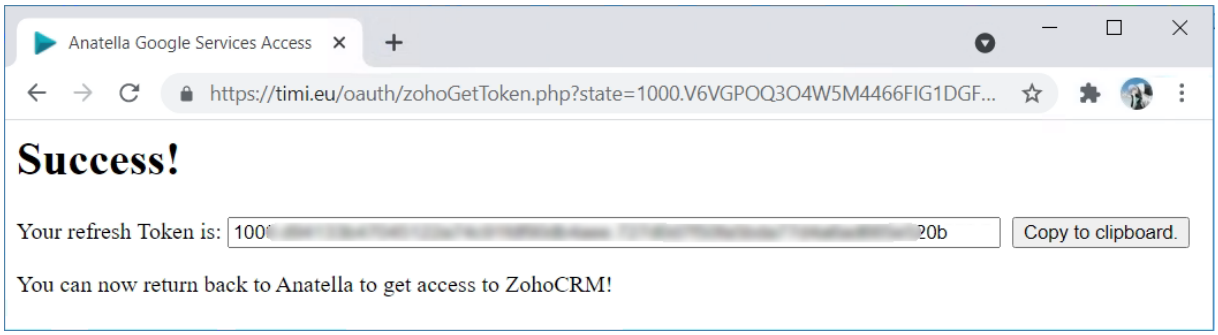


7. Click the “Accept” button to allow you to download your ZohoCRM data:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

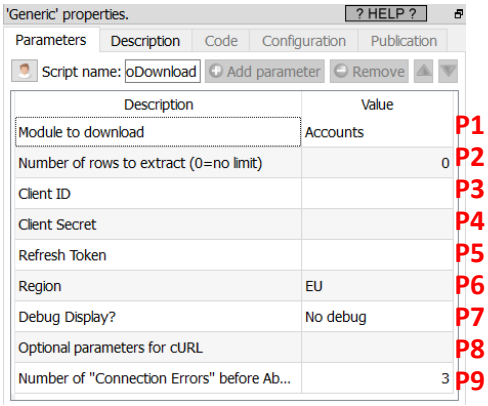
8. You finally get your “Refresh Token”:



5.23.45. Download data from Zoho CRM

Icon:

Property window:



Short description:
Download Data from Zoho CRM.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.
Before downloading data from Zoho CRM with Anatella, you need to get the 4 parameters:

- Client ID (Parameter **P3**)
- Cient Secret (Parameter **P4**)
- Refresh Token (Parameter **P5**)
- Region (Parameter **P6**)

…from the ZohoCRM website. Please refer to the previous section 5.23.44. for the exact procedure to get these 4 parameters.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.46. Twitter

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: Twitter | Add parameter | Remove

Description	Value	
Search query result type	Mixed(default)	P1
Operating search mode	mode 1: Standard search	P2
mode 1 : All of these words		P3
mode 1 : This exact phrase		P4
mode 1 : Any of these words		P5
mode 1 : None of these words		P6
mode 1 : These hashtags		P7
mode 1 : From these accounts		P8
mode 1 : To these accounts		P9
mode 1 : Mentioning these accounts		P10
mode 1 : Since date (yyyy-MM-dd)	> DateTool("yyyy-MM-...	P11
mode 1 : Before date (yyyy-MM-dd)	> DateTool("yyyy-MM-...	P12
mode 2 : Custom Search	from:twitterDev	P13
Limit (0=no limit)	200	P14
Enable Geocoded search	☐	P15
Geocoded search - latitude	50.812877	P16
Geocoded search - longitude	4.392626	P17
Geocoded search - Radius (km)	500	P18
Consumer API Key		P19
Consumer API Secret		P20
Optional parameters for cURL		P21
Debug Display?	No debug	P22
Number of "Connection Errors" before ...	5	P23
Error Management	Abort Graph Execution ...	P24

Short description:
Download tweets
From Twitter

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P21** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Twitter Website (i.e. you need the parameters **P19** and **P20**). Please see the next section 5.23.46.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.46.1, you can use the parameters from **P1** to **P18** to extract tweets from twitter.

5.23.46.1. Twitter first time setup

Before using the Twitter Action inside Anatella, you need to get these 2 parameters:

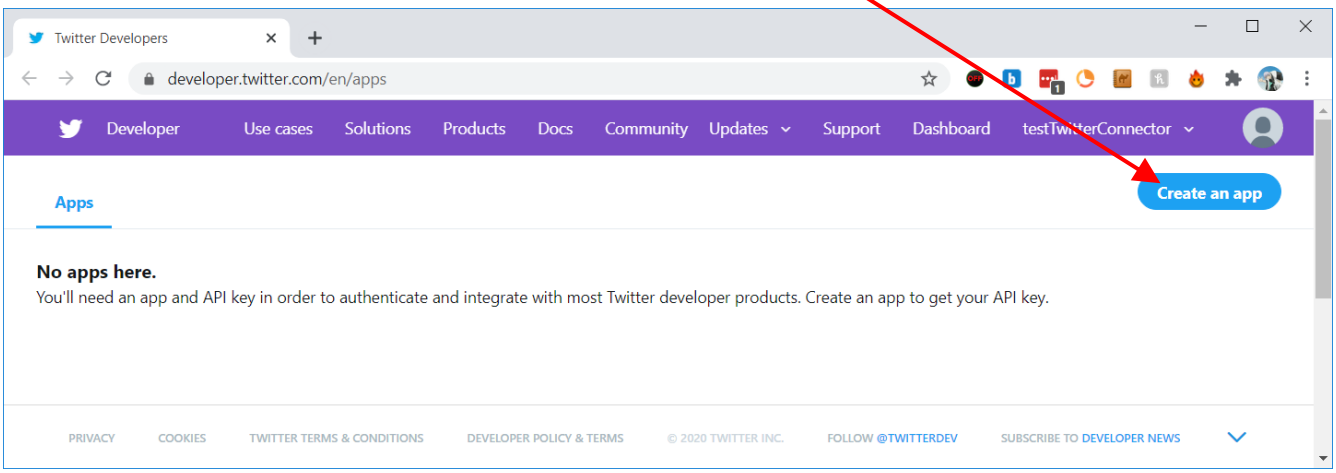
- Anatella Parameter **P19**: Consumer API Key
- Anatella Parameter **P20**: Consumer API Secret

…from the Twitter website.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

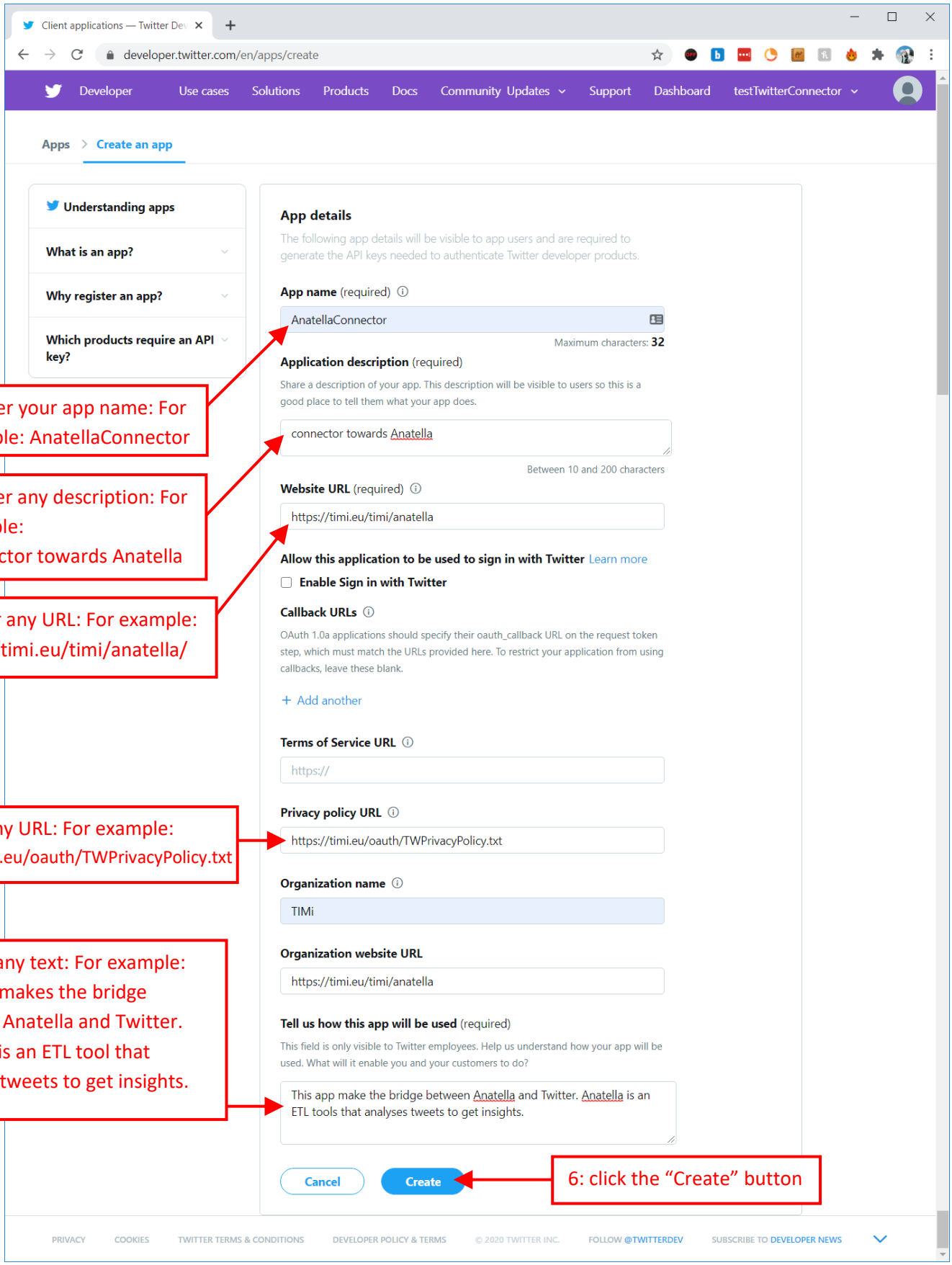
Here are the steps to get these 2 parameters:

1. Open the url https://developer.twitter.com/en/app (if necessary, login with your credentials). Once you are logged in, click on the "Create an app" button:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Enter the following details for the new app:



1: Enter your app name: For example: AnatellaConnector

2: Enter any description: For example: connector towards Anatella

3: Enter any URL: For example: https://timi.eu/timi/anatella/

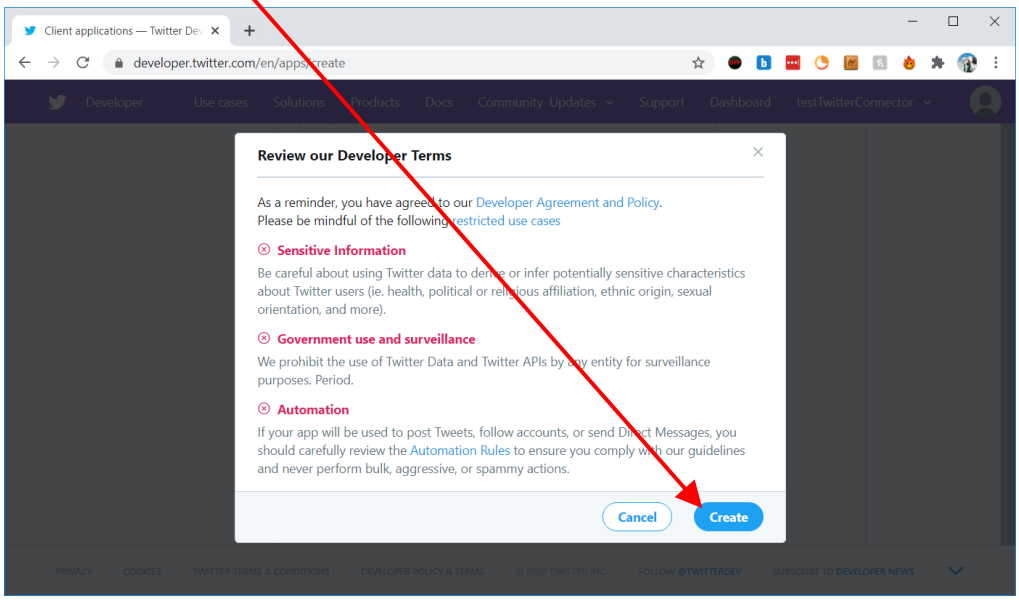
4: Enter any URL: For example: https://timi.eu/oauth/TWPrivacyPolicy.txt

5: Enter any text: For example: This app makes the bridge between Anatella and Twitter. Anatella is an ETL tool that analyses tweets to get insights.

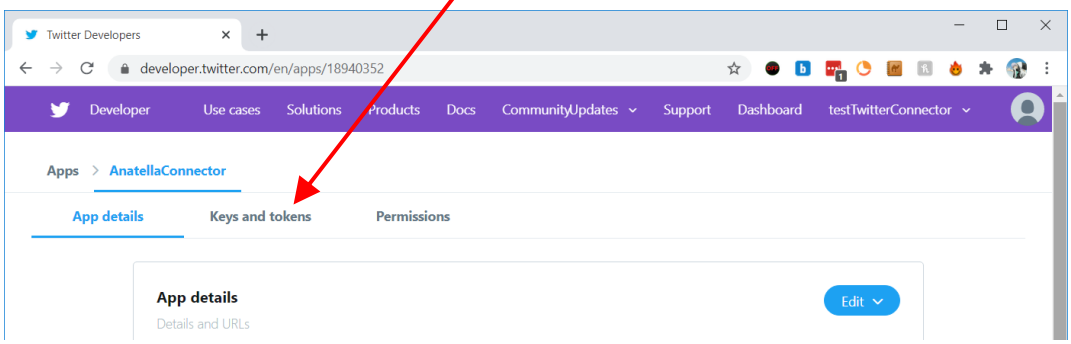
6: click the "Create" button

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

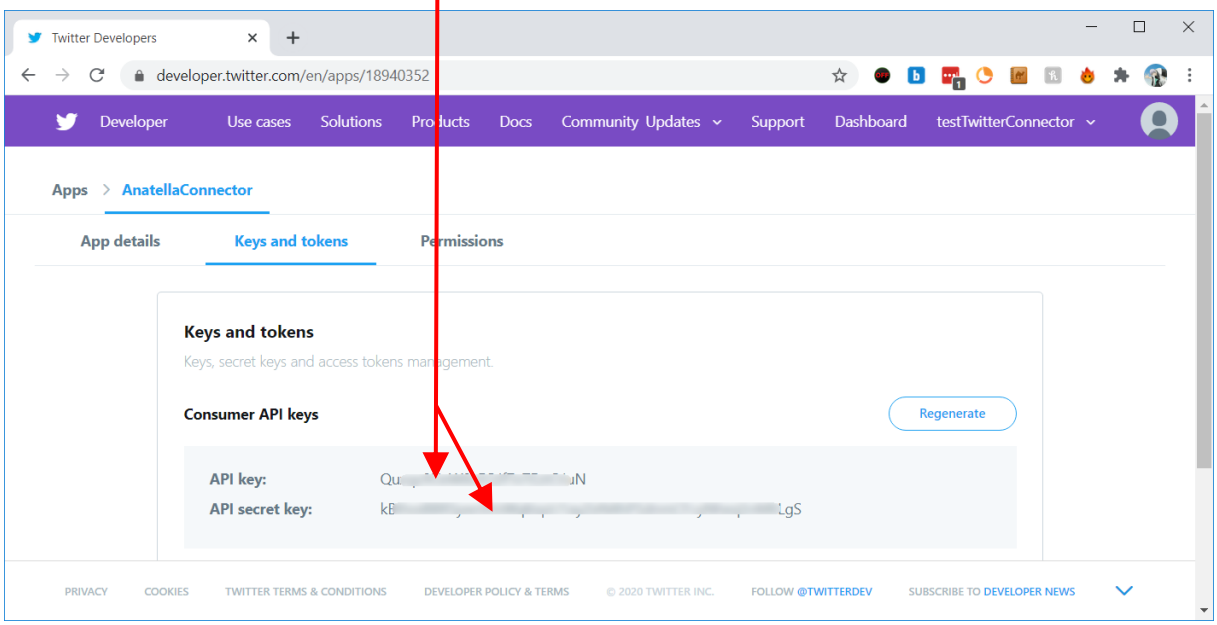
3. Click the “Create” button:



4. Open (i.e. click on) the “Keys and tokens” tab:



5. The Anatella Parameter **P19** (Consumer API Key) and the Anatella Parameter **P20** (Consumer API Secret) are directly visible here: Copy-paste them into Anatella!



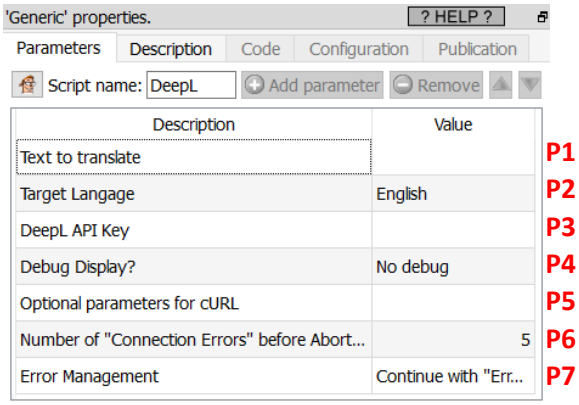
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. Your Twitter API Key is not immeditaly activated: You first need to wait a few minutes before it actually allows you to log-in into Tweeter with Anatella.

5.23.47. DeepL automated translation

Icon:

Property window:



Short description:
Use the DeepL API to compute some translations

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

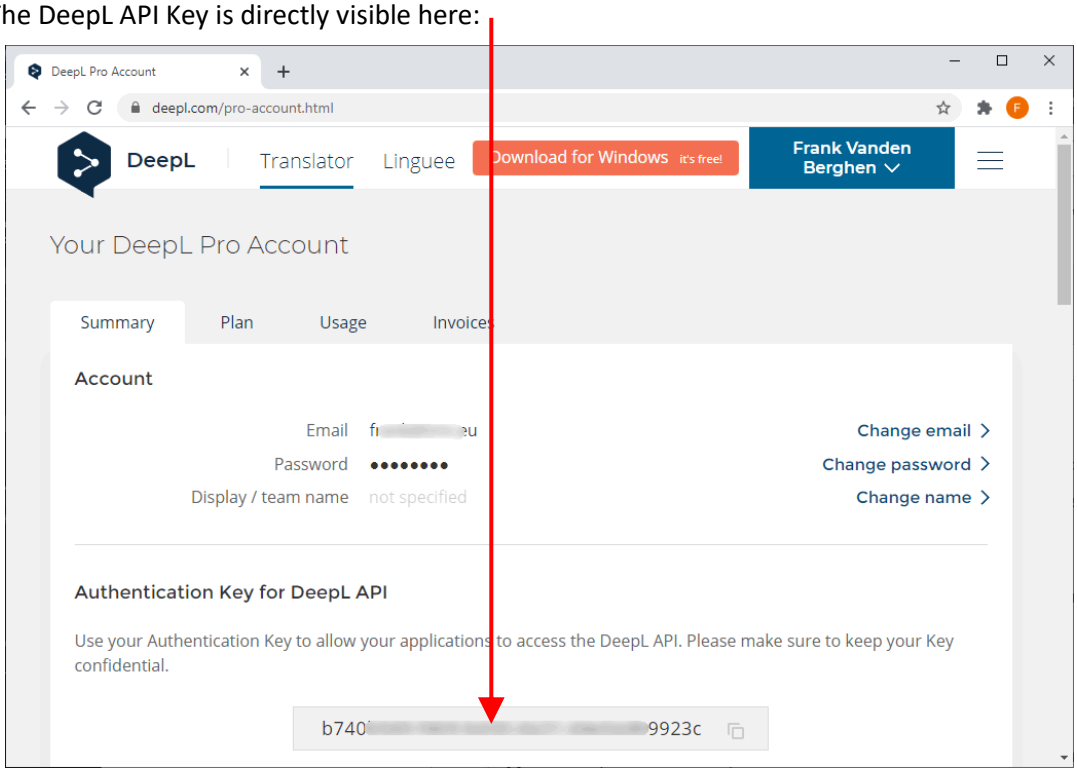
To use this Action, you'll need to get one parameter from the DeepL Website (i.e. you need the parameters **P3**). Please see the next section 5.23.47.1 for more details on how to get this parameter.

Once you have completed the "setup process" described in the section 5.23.47.1, you can use the parameter **P1** to compute the required translations.

5.23.47.1. DeepL first time setup

Before using the DeepL Action inside Anatella, you need to get the DeepL API Key from the DeepL website. Here are the steps to get this parameter:

1. Open the url: https://www.deepl.com/pro-account.html (if necessary, log in using your credentials)
2. The DeepL API Key is directly visible here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.48. Google Speech-to-Text

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: GoogleSpeechToText | Add parameter | Remove

Description	Value	
Audio file to upload		P1
(optional) All sound files are longer than 1 minute	☑	P2
Language spoken in the audio file	English (United States)	P3
Sample rate Hertz for .ogg files	16000	P4
Operating mode	mode 1 : local files less than 1min	P5
(optional) mode 3 : Filename on remote server		P6
mdoe 3 : Bucket name	my_bucket	P7
mode 3 : delete uploaded file when finished	☐	P8
Client ID		P9
Client Secret		P10
Refresh Token		P11
Optional parameters for cURL		P12
Debug Display?	No debug	P13
Number of "Connection Errors" before Abortin...	5	P14
Error Management	Continue with "Error" Status	P15

Short description:
Use the Google API to convert sound files to text

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P12** for web-access through a PROXY server.

To be able to use this Action, you need to get these 3 parameters from Google: (1) your “Client ID”, (2) your “Client Secret”, (3) your “Refresh Token”. To get these 3 parameters, you must use the “Unlock Google Services” action detailed in section 5.23.11.
Once you have completed the “setup process” described in the section 5.23.11, you can use this Action to convert your sound files (wav, mp3, ogg) to text.

5.23.49. Text Razor Topic Extraction

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: RazorTopics | Add parameter | Remove

Description	Value	
Text to analyze		P1
Primary Key Column		P2
API Key		P3
Debug Display?	No debug	P4
Optional parameters for cURL		P5
Number of "Connection Errors" before Ab...	5	P6
Error Management	Abort Graph Executi...	P7

Short description:
Use the TextRazor API to extract the Topics from your text corpus

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

To use this Action, you’ll need to get one parameter from the TextRazor Website (i.e. you need the parameters **P3**). Please see the next section 5.23.49.1 for more details on how to get this parameter.

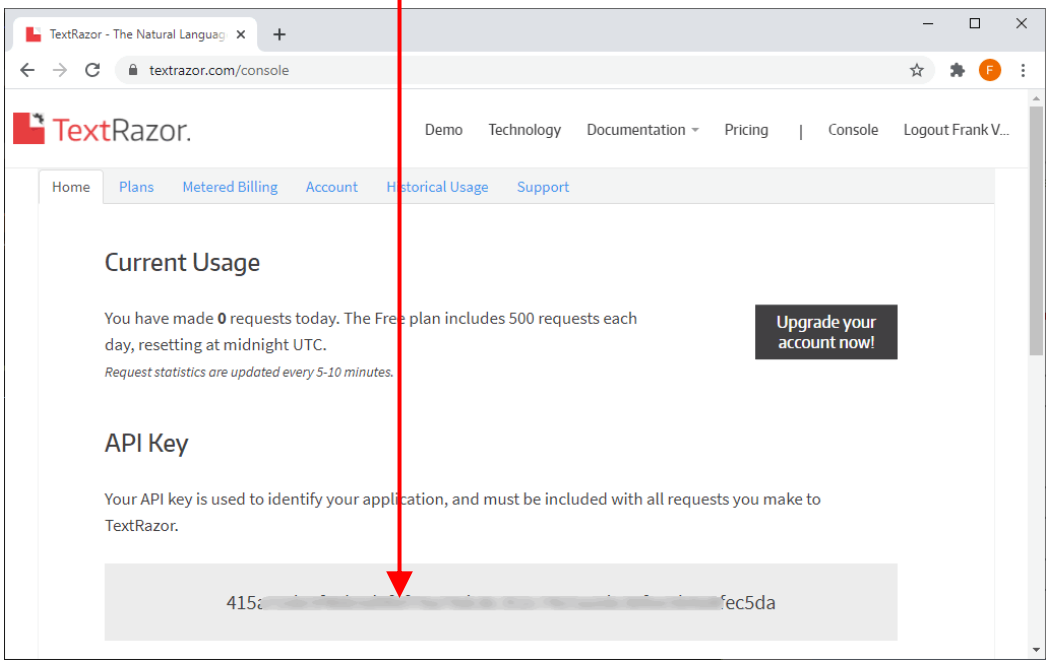
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Once you have completed the "setup process" described in the section 5.23.49.1, you can use the parameter **P1** and **P2** to extract the Topics from your text corpus.

5.23.49.1. TextRazor first time setup

Before using the TextRazor Action inside Anatella, you need to get the TextRazor API Key from the TextRazor website. Here are the steps to get this parameter:

1. Open the url: https://www.textrazor.com/console (if necessary, log in using your credentials)
2. The TextRazor API Key is directly visible here:



5.23.50. Unicheck plagiarism checker

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: Unicheck | Add parameter | Remove

Description	Value	
Filename to check for similarity		P1
(Optional) Local filepath for report (to save locally)		P2
(Optional) Remote Folder Id		P3
Perform similarity check	☑	P4
Download report	☑	P5
Delete file when finished	☑	P6
Source for similarity check	Internet+Library	P7
Client Key		P8
Client Secret		P9
Debug Display?	No debug	P10
Optional parameters for cURL		P11
Number of "Connection Errors" before Aborting...	5	P12
Error Management	Continue with "...	P13

Short description:
Use the Unichek API to check for Plagiarism in your text corpus

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P11** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

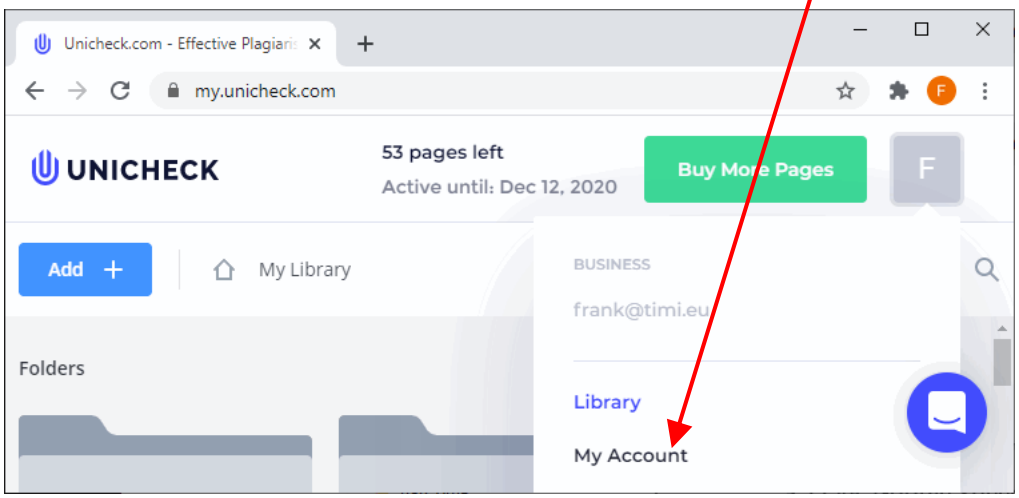
To use this Action, you'll need to get one parameter from the Unicheck Website (i.e. you need the parameters **P8** and **P9**). Please see the next section 5.23.50.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.50.1, you can use the parameter **P1** to **P7** to check for Plagiarism in your text corpus

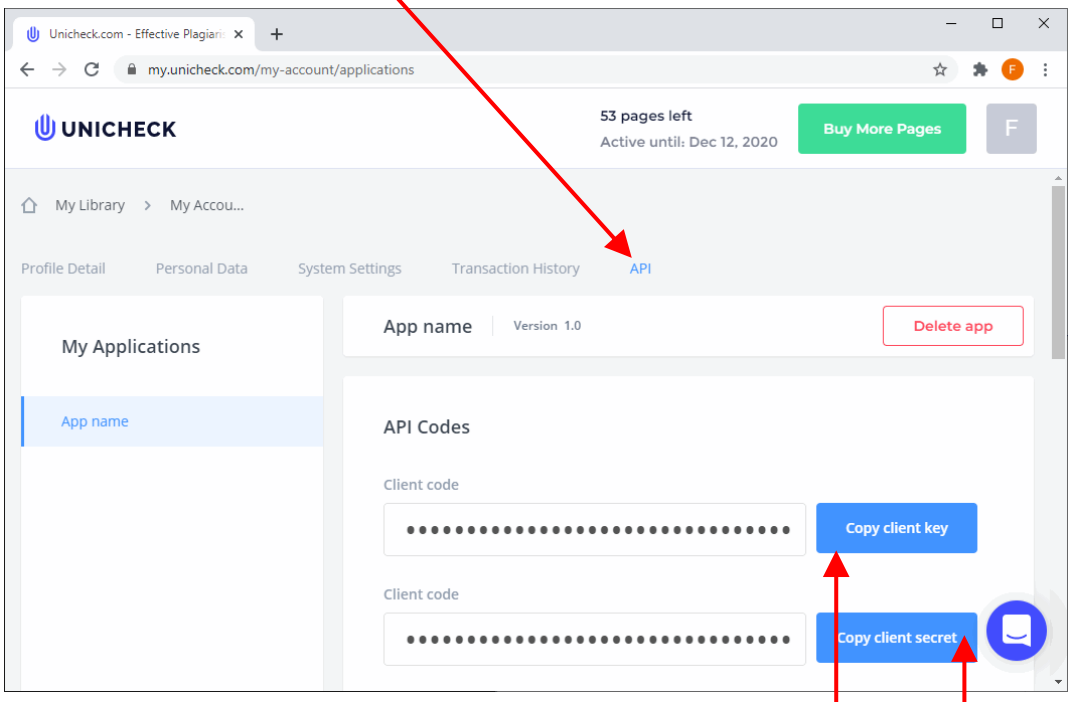
5.23.50.1. Unicheck first time setup

Before using the Unicheck Action inside Anatella, you need to get the Unicheck Client Key and the Unicheck Client secret from the Unicheck website. Here are the steps to get these 2 parameters:

1. Open the url: https://unicheck.com and log-in using your credentials.
2. Open the drop-down menu in the top-right corner and click "My account":



3. Click on the "API" menu in the middle:



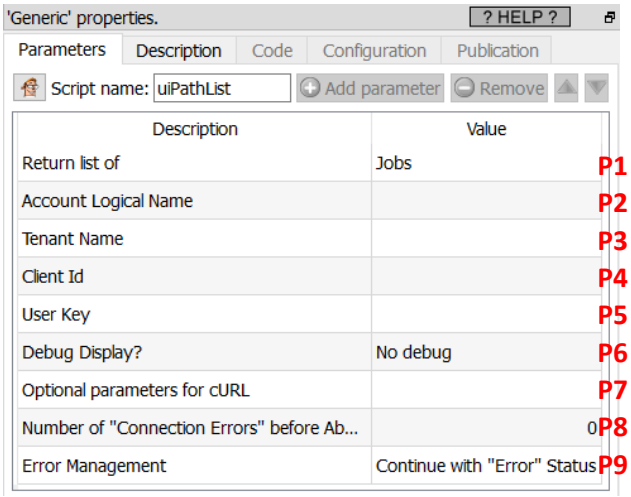
4. The "Unicheck Client Key" and the "Unicheck Client Secret" are accessible here and here

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.51. UI Path List

Icon:

Property window:



Short description:
List the processes and the jobs inside your UIPath Orchestrator

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

To use this Action, you'll need to get four parameters from the UIPath Orchestrator Website (i.e. you need the parameters **P2, P3, P4** and **P5**). Please see the next section 5.23.51.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.51.1, you can use the parameter **P1, P6, P7, P8, P9** to List the processes and the jobs inside your UIPath Orchestrator.

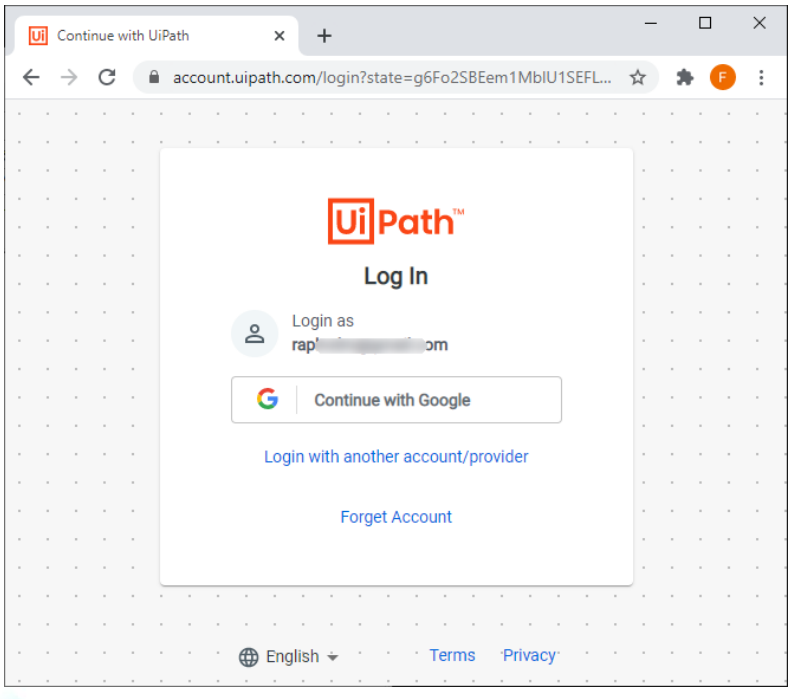
5.23.50.1. UIPath first time setup

Before using the UIPath Actions inside Anatella, you need to get:

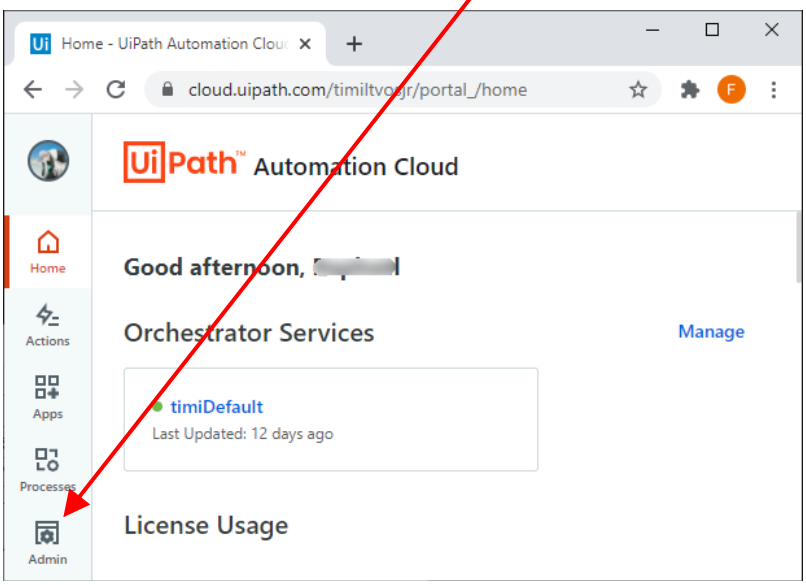
- the Acccount Logical Name
- the Tenant Name
- the Client ID
- the User Key

…from the UIPath Orchestrator website. Here are the steps to get these 4 parameters:

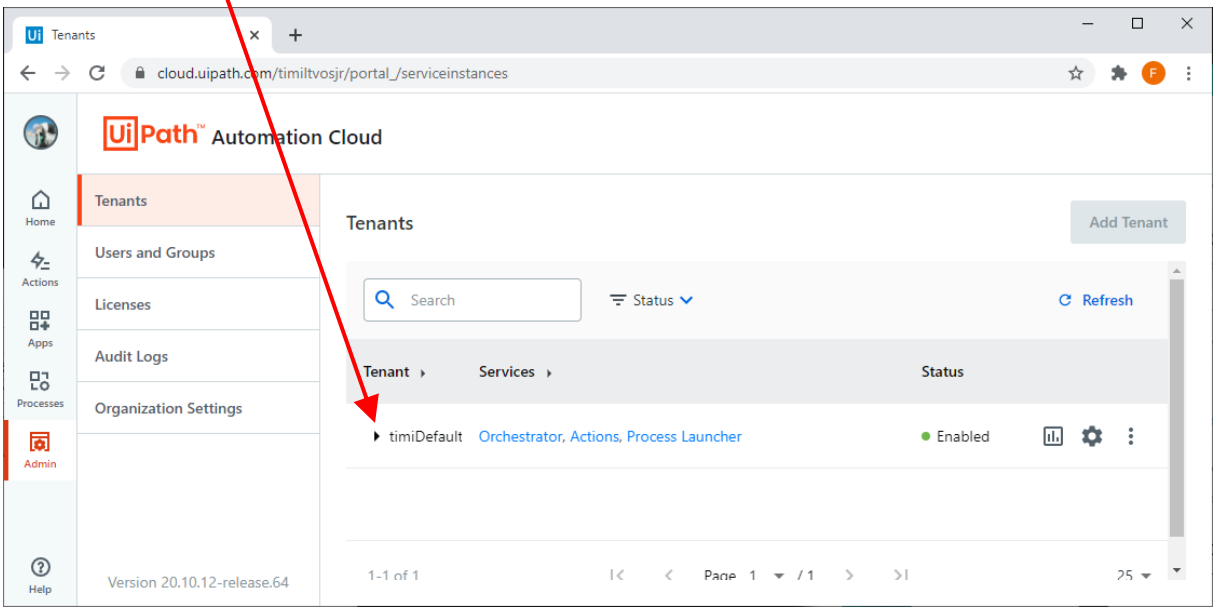
1. Open the url: https://cloud.uipath.com/ and log-in using your credentials:



2. Click on the “Admin” button on the left:

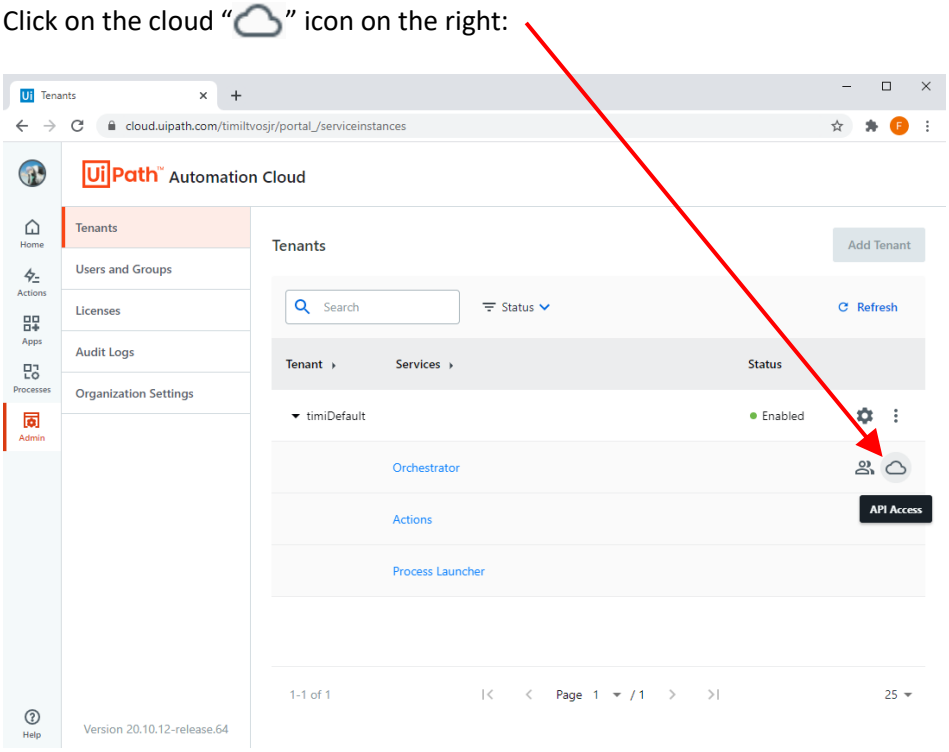


3. Click on the little arrow “ ▶ ” to open the details of the tenant that will be used to connect to the orchestrator:

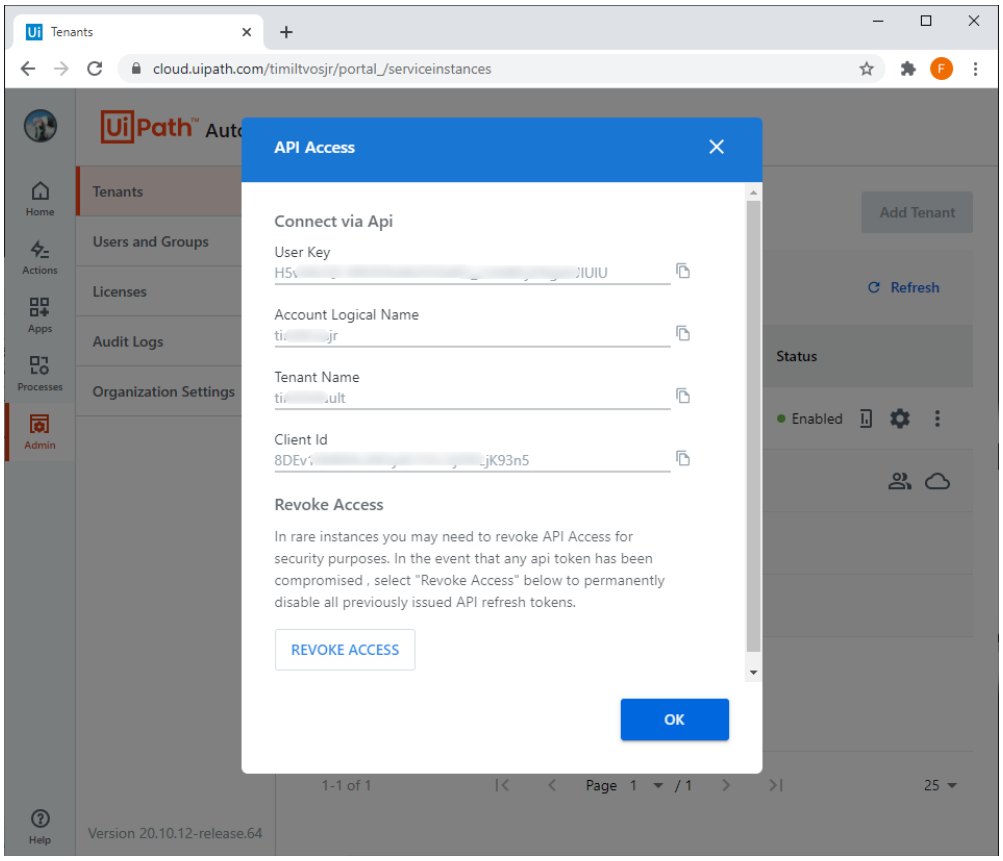


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. Click on the cloud "☁" icon on the right:



5. The four required parameters to connect to the UIPath ochestrator are now visible:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.52. UI Path Run Processes

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: uiPathRun | Add parameter | Remove

Description	Value	
Process Name	UI_Path_Processes	P1
Process Arguments		P2
Account Logical Name		P3
Tenant Name		P4
Client Id		P5
User Key		P6
Debug Display?	No debug	P7
Optional parameters for cURL		P8
Number of "Connection Errors" before Ab...		P9
Error Management	Abort Graph Execution	P10
Timeout (s)	60	P11

Short description:
Create and run jobs inside your UIPath Orchestrator

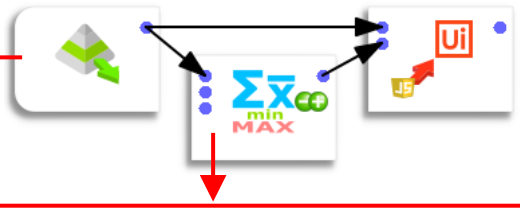
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get four parameters from the UIPath Orchestrator Website (i.e. you need the parameters **P3, P4, P5** and **P6**). Please see the previous section 5.23.51.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.51.1, you can use this action to create and add jobs inside your UIPath Orchestrator (and directly run these jobs).

A typical setup looks like this:

A table with:
- The "UI_Path_Processes" column that contains the name of all the UI Path processes to execute.
- Some other additional columns whose column's name are identical to the argument's names of the given UIPath processes to execute.



The list of "UI Path processes" to execute (without duplicates). To compute this list:

'aggregate' properties. ? HELP ?

Use in-RAM algorithm for small output tables | Add Aggregate | Remove

Aggr.1 | Adv. Parameters

Group by: | Add "Group by" variable | Remove

Var
UI_Path_Processes

Let's give a small example:

Let's assume that we have two processes (named "P1" and "P2") with the following arguments: P1(arg1,arg2) and P2(arg1,arg3). In other words:

- The table on input pin 0 should ideally contains the columns named "UI_Path_Processes", "arg1", "arg2", "arg3".
- The table on input pin 1 should be:

UI_Path_Processes
P1
P2

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If one of the arguments (arg1, arg2 or arg3) is missing from the input (on pin 0), then Anatella displays a warning message and the default value for the missing argument is used.

When running a process (P1 or P2), Anatella sends the highest quantity of arguments possible that exist inside the input table for any given process.

For example, if the input table (on pin 0) looks like this:

UI_Path_Processes	arg1	arg2
P1	a1	a2
P2	b1	b2
P1	c1	c2

…Then:

- Both arguments “arg1”, “arg2” are sent to P1 (since the process P1 accepts both arguments).
- The process P2 only receives "arg1" (Since the process P2 only accepts the "arg1" and "arg3" arguments).
- Anatella creates and runs 2 jobs based on the process P1 (these 2 jobs have different values for the arguments “arg1” and “arg2”).
- Anatella creates and runs 1 job based on the process P2.

How does this Action operate?

To execute any job within “UI Path”, you always need to a “ROBOT”. So, this Anatella action starts by listing all the “ROBOTS” available inside your Orchestrator to run all the required jobs (i.e. to run all the jobs given on “input pin 1”). This “ROBOT LIST” is very important and will be used to execute one-by-one all the processes given in “input pin 0”.

To add a new job inside the UI Path Orchestrator, Anatella proceeds in this way:
Step 1: Read from pin 0 the name of the next process to execute and the value of all the arguments related to this process.
Step 2: Open/select the next entry inside the “ROBOT LIST”.
Step 3: Test if the currently selected ROBOT inside the “ROBOT LIST” is able to execute the current process (i.e. not all robots can execute all processes). If that’s not the case, go back to step 2 (to select&test the next robot).
Step 4: Create and add a new job inside the UI Path Orchestrator: i.e. Use the currently selected ROBOT to execute the currently selected Process. The creation of this new job might fail because the currently selected ROBOT is still busy to finish some previous job. If that’s the case, Anatella detects the error condition and Anatella re-tries to create the new job during **P11** seconds (one re-try per second). If it’s still impossible to create the new job after **P11** seconds, Anatella will either (1) abort the process or (2) return a “fail” flag on the output pin (depending on the value of the **P10** parameter).
Step 5: If we have read all the rows/processes from the input pin 0, stop. Otherwise, go to step 1.

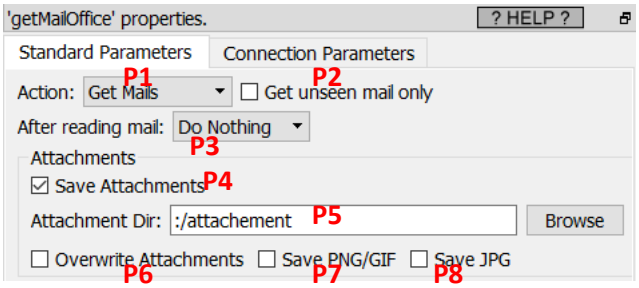
To summarize, the input table (on pin 0) contains your queue of “*processes to execute*” (with the accompanying parameters). Anatella create a new job for each process inside your queue. Anatella assigns the new job to the right ROBOT (i.e. to the ROBOT that is the most likely to be idle).

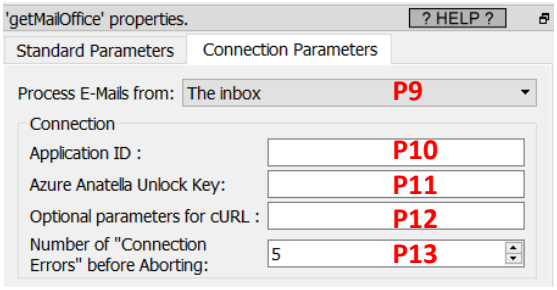
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.53. Get Emails using your Office 365 subscription

Icon:

Property window:





Short description:
Get Emails using your Office 365 subscription

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P12** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Azure Website (i.e. you need the your "Application ID" -parameter **P10**- and your "Anatella Unlock Key" -parameter **P11**). Please see the section 5.23.27. for more details on how to get these parameters.

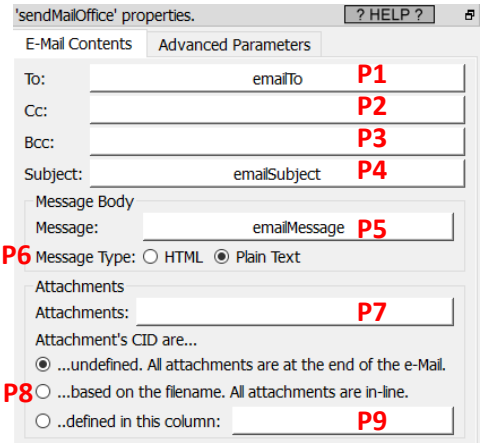
Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P1** to **P9** to download/list/purge the emails from your office 365 subscription.

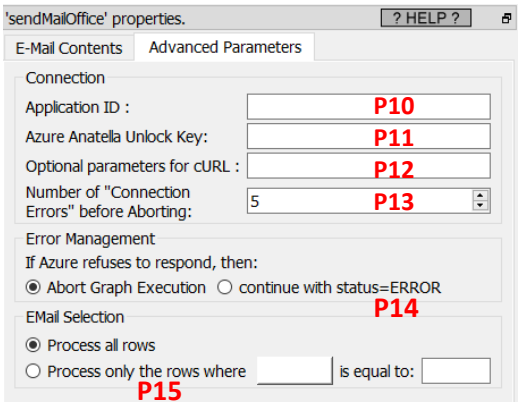
5.23.54. Send Emails using your Office 365 subscription

Icon:

Property window:

Short description:
Send Emails using your Office 365 subscription





Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P12** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Azure Website (i.e. you need the your "Application ID" -parameter **P10**- and your "Anatella Unlock Key" -parameter **P11**). Please see the section 5.23.27. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P1** to **P9** , **P14** and **P15** to send emails using your office 365 subscription.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Using the parameter **P8**, you chose one of the 3 operating modes that are related to the e-mail attachments:

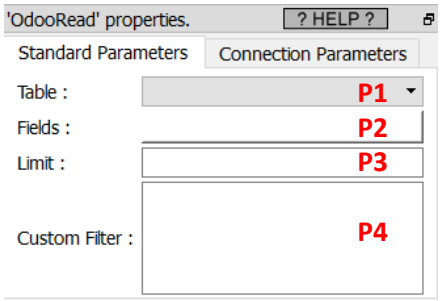
1. Attachment CID are undefined. All attachement are at the end of the e-mail.
2. Attachment CID are based on a filename. All attachements are in-line.
3. Attachment CID are defined in an input column.

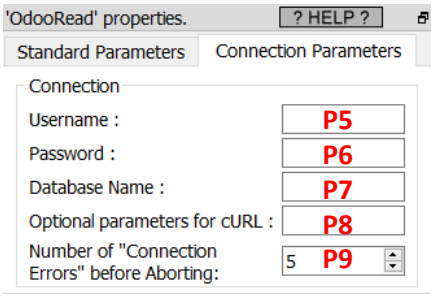
The exact differences that exists between these 3 operating modes (for the parameter **P8**) are explained inside the section 5.27.22 about the OutlookSend Action.

5.23.55. Download data from Odoo

Icon:

Property window:





Short description:
Download Data from Odoo

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Odoo Website (i.e. you need the parameters **P5**, **P6** and **P7**). Please see the section 5.23.55.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.55.1, you can use the parameters **P1** to **P4** to download data from your Odoo server. The parameters **P1** to **P2** are self-explanatory. The parameter **P3** is the number of rows to extract from Odoo (leave empty to extract all the rows).

The parameter **P4** is an expression that allows to select&download some of the rows of the input table (instead of downloading *all* the rows). For example, let's assume that we want to download the rows that have an "id" ("id" is one of the column of the input table to download) that is between 30 (included) and 50 (excluded). In other words, our row-filter is:

```
(id >= 30) && (id<50)
```

Unfortunately, you cannot directly write such an expression inside the parameter **P4** because this parameter expects a different syntax based on XML. The same row-filter expression translated to the Odoo XML syntax is:

```
<value><string>&amp;</string></value>

<value><array><data><value><string>id</string></value><value><string>&gt;=</string></value
><value><int>30</int></value></data></array></value>

<value><array><data><value><string>id</string></value><value><string>&lt;</string></value>
<value><int>50</int></value></data></array></value>
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You'll find here a complete documentation on how to write these row-filter expressions using the Odoo XML syntax:

https://www.odoo.com/documentation/master/reference/orm.html#reference-orm-domains

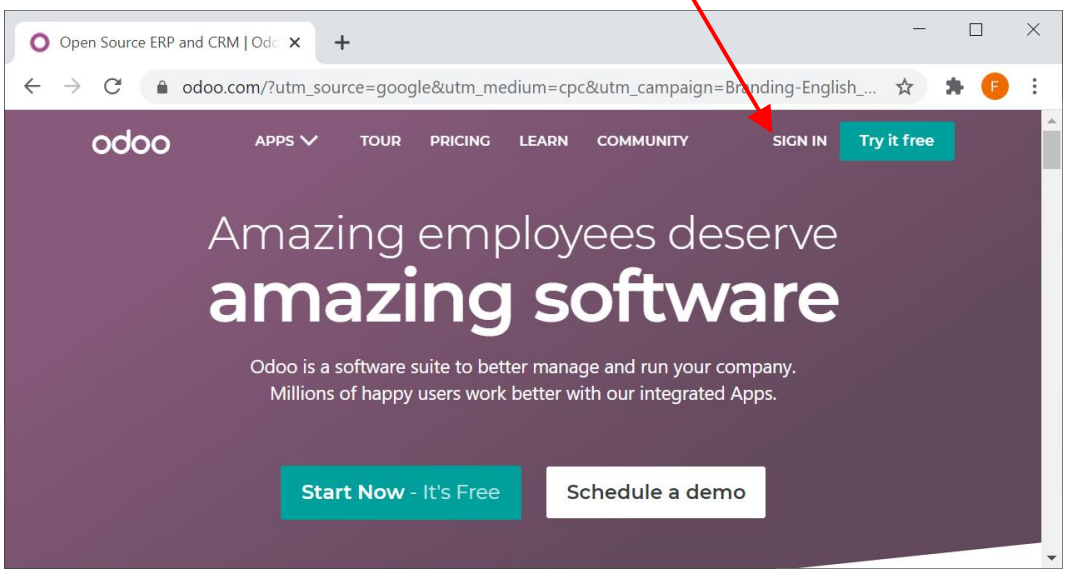
5.23.55.1. Odoo first time setup

Before using the Odoo Actions inside Anatella, you need to get:

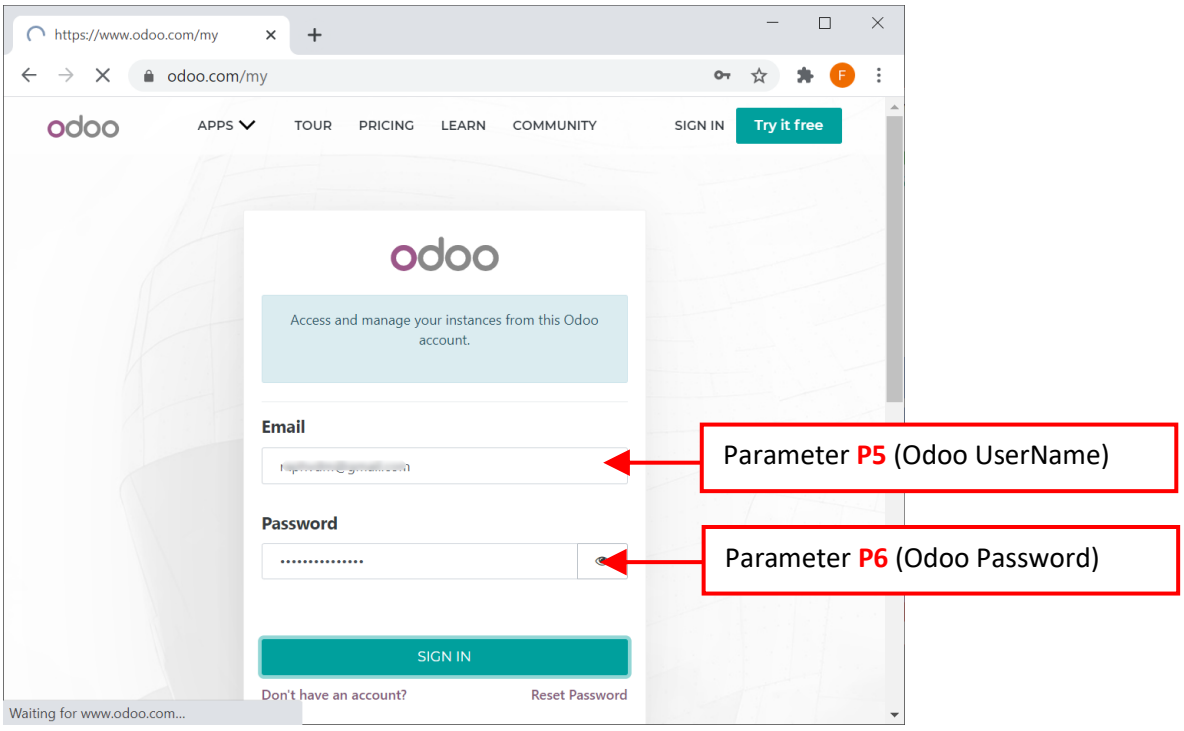
- Your username
- Your password
- Your Database name

…from the Odoo website. Here are the steps to get these 3 parameters:

1. Open the url: https://odoo.com/ and click on "sign in":

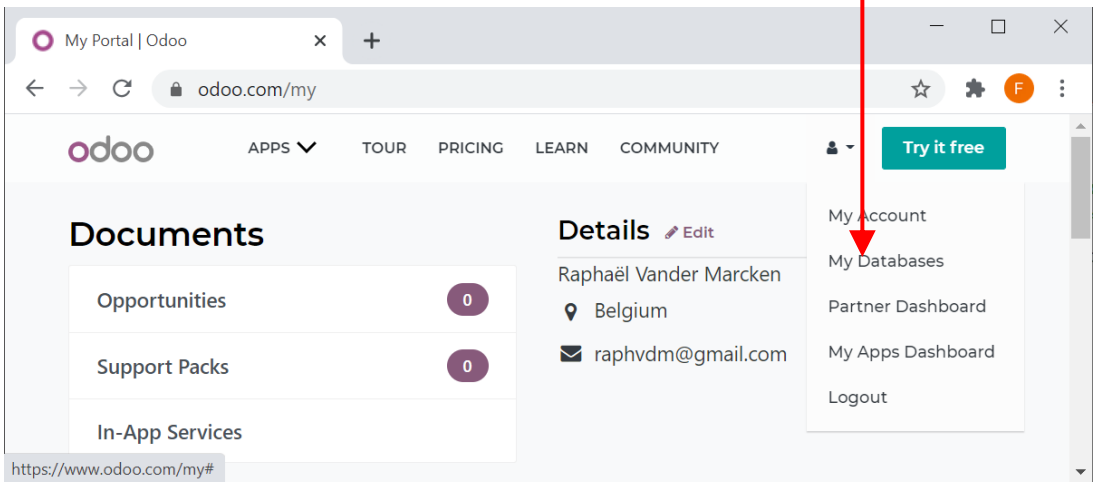


2. Sign-in using your Odoo credentials: These are the parameters **P5** (Odoo UserName) and **P6** (Odoo Password):

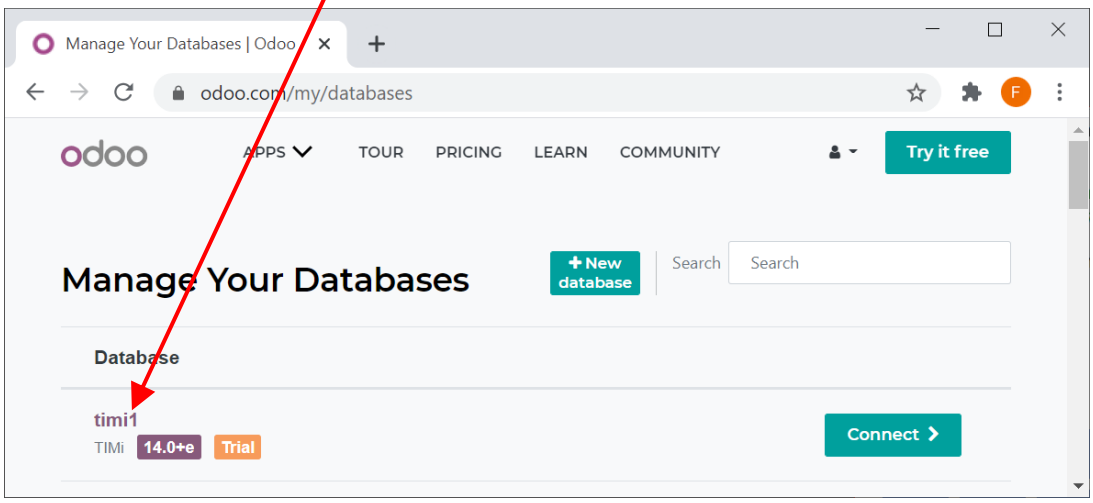


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Click on the " " icon and select "My databases" inside the drop-down menu:



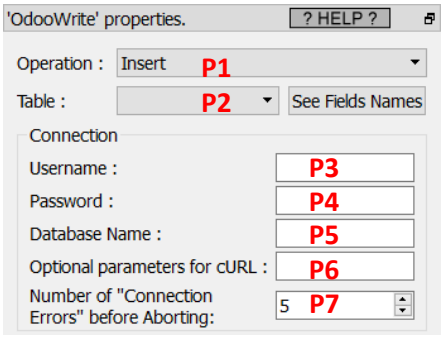
4. The last parameter **P7** (i.e. your Odoo database name) is now visible.
In the example below, it's "timi1":



5.23.56. Upload data to Odoo

Icon:

Property window:



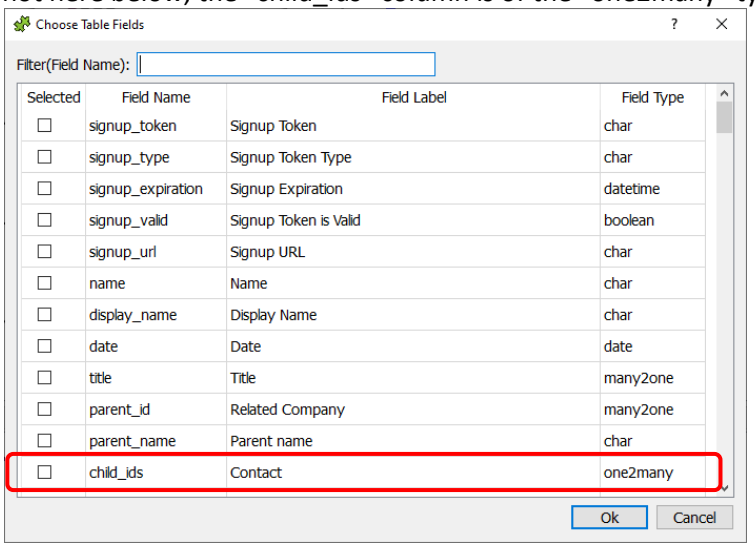
Short description:
Upload Data to Odoo

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To use this Action, you'll need to get several parameters from the Odoo Website (i.e. you need the parameters **P3**, **P4** and **P5**). Please see the section 5.23.55.1 for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.55.1, you can use the parameters **P1** and **P2** to upload data to your Odoo server. The column's names inside the input table of the OdooWrite action in Anatella must match the column's names of the table inside the Odoo server. Inside some Odoo tables, some columns have the "one2many" or the "many2many" types. For example, in the screenshot here below, the "child_ids" column is of the "one2many" type:



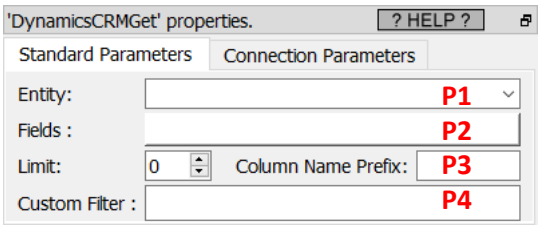
For these types of columns, Odoo expects to get cells that each contain a XML structure that represents an array. For example, the array [4,5,6] is represented in "Odoo XML" as:

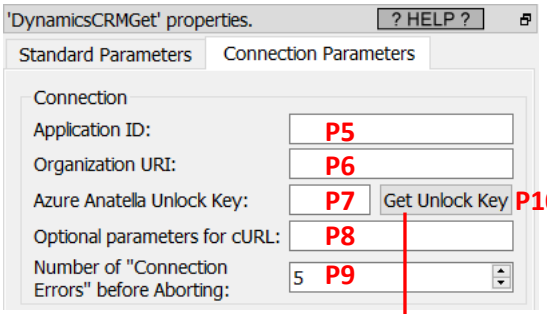
```
<array><data>
<value><int>4</int></value>
<value><int>5</int></value>
<value><int>6</int></value>
</data></array>
```

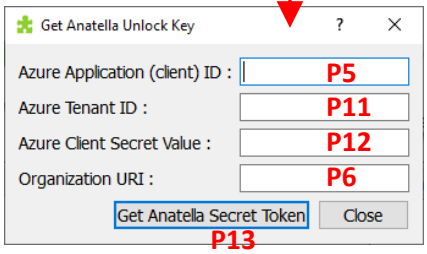
5.23.57. Get data from Microsoft Dynamics CRM

Icon:

Property window:







Short description:
Download Data from Microsoft Dynamics CRM

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To use this Action, you'll need to get several parameters from the Azure Website (i.e. you need the parameters **P5**, **P6** and **P7**). Please see the next two sections 5.23.57.1 and 5.23.57.2 for more details on how to get these parameters.

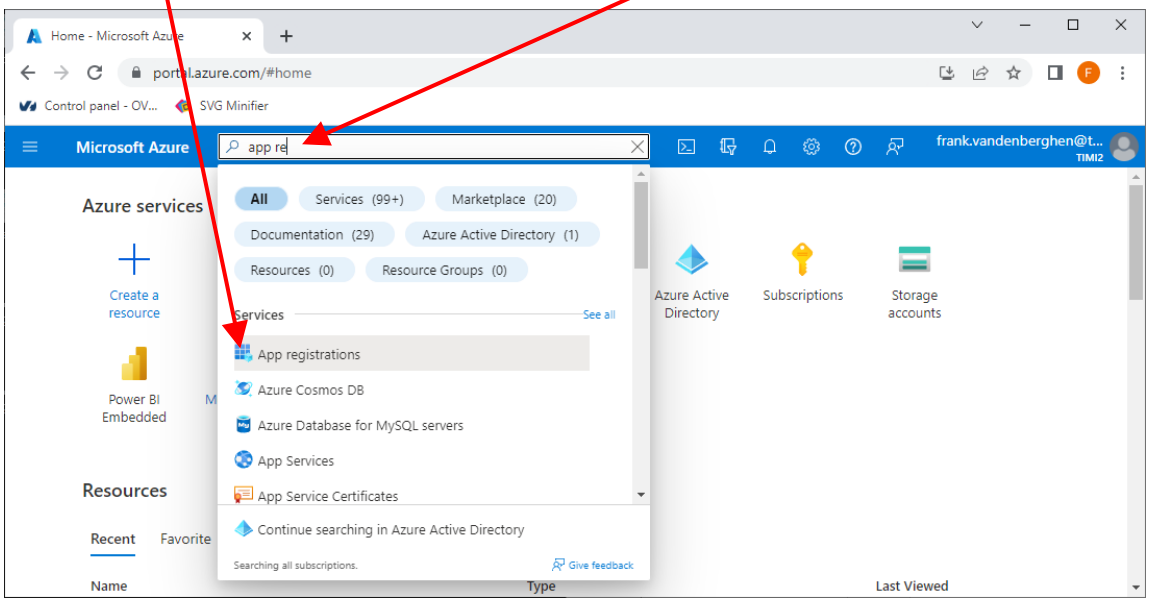
Once you have completed the "setup process" described in the sections 5.23.57.1 and 5.23.57.2, you can use the parameters **P1** to **P4** to download data from your Microsoft Dynamics CRM system.

The parameters **P5** ("Application (Client) ID") and **P6** will never expires. In opposition, your "Anatella Unlock Key" (parameter **P7**) will expire after a few months (that depends on the settings that you selected at the step 20.3 from the next section 5.23.57.1.). The maximum duration of an "Anatella Unlock Key" is 2 years (this limitation is common to all the tools accessing Azure since it's imposed by Azure). When your "Anatella Unlock Key" expires, you just need to redo the easy 3-steps procedure given inside section 5.23.57.2 (there is no need to redo the whole procedure from section 5.23.57.1!!).

5.23.57.1. MS-Dynamics First Time Setup

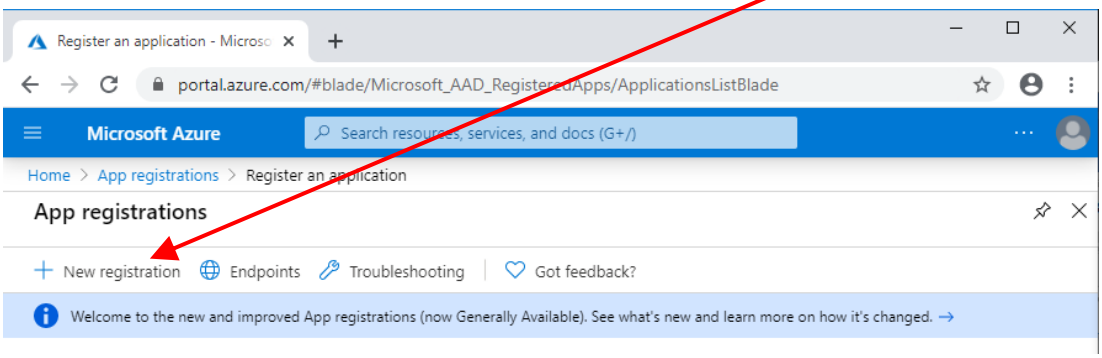
The procedure that is given inside this section must only by performed **once** by an Azure Administrator. The first-time-setup procedure to connect to MS-Dynamics on Azure is the following:

1. Open the URL https://portal.azure.com and "log-in" using your normal "Login" and "Password" for Azure. Then, type "App Registration" in the searchbar and click on the corresponding icon:

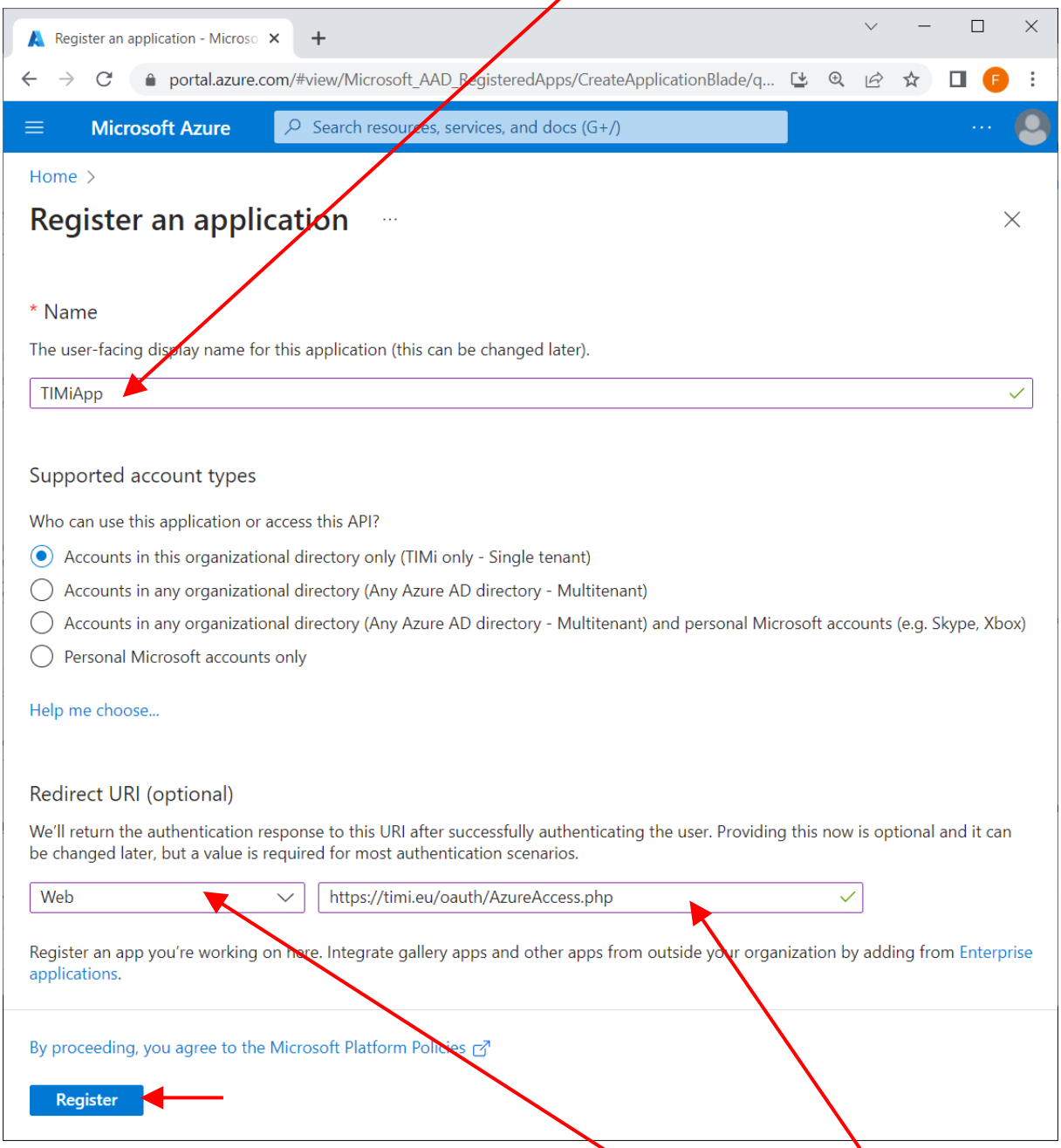


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. We will create a new Azure App: Click the “New registration” button:



3. Give a name to your Azure App (you can use any name)



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. For the redirect URI: select the “Web” type of application here and enter as URL: https://timi.eu/oauth/AzureAccess.php

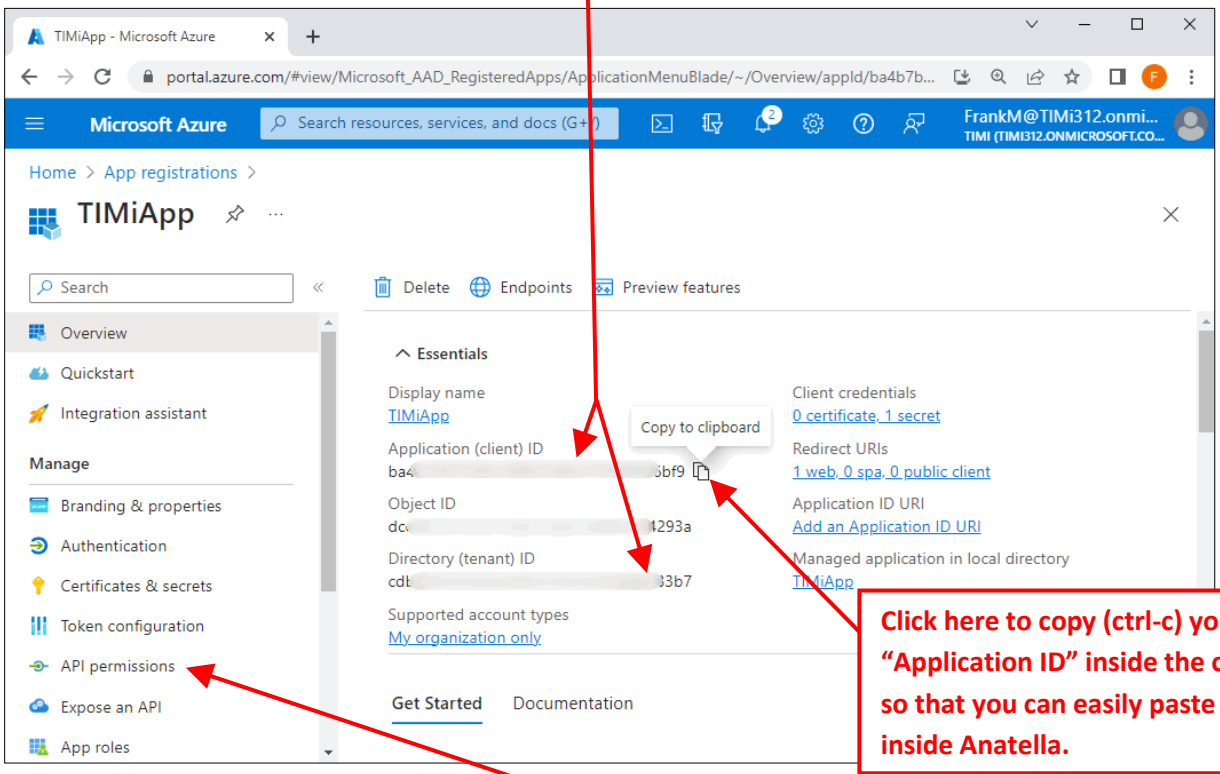
   …as the “redirect URI”.

   WARNING: This is a very important step: The URI must be exactly the one given here above (this is case sensitive) and the type of the application must be “Web”.

5. Click the “Register” button at the bottom of the webpage to go to the next page.

6. Inside the “Advanced Parameters” panel of the Dynamics action inside Anatella, click on the Button parameter **P10** to open the “Get Anatella Unlock Key” windows.

7. You are now receiving the Anatella parameter **P5** and **P11** (i.e. your “Application (client) ID” and your “Tenant ID”): They are here: Copy/paste these 2 parameters inside the “Get Anatella Unlock Key” windows

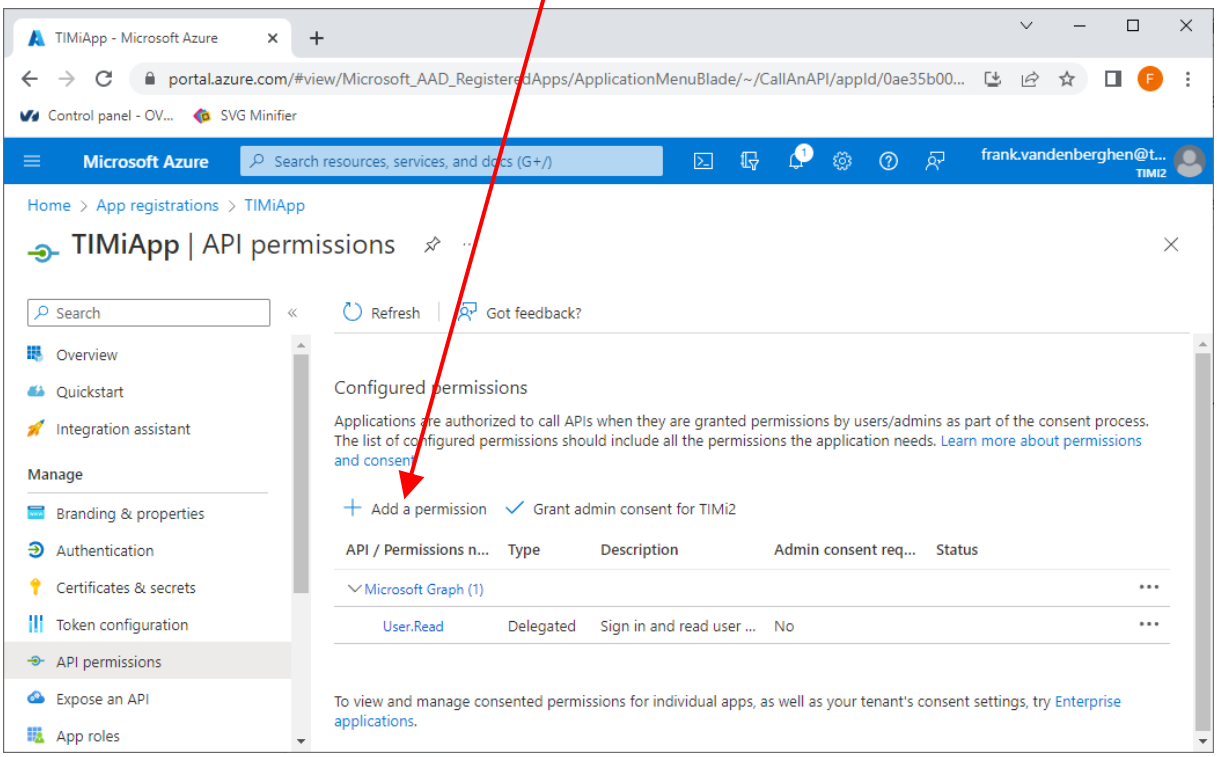


8. Click on “API Permissions” in the left panel here.

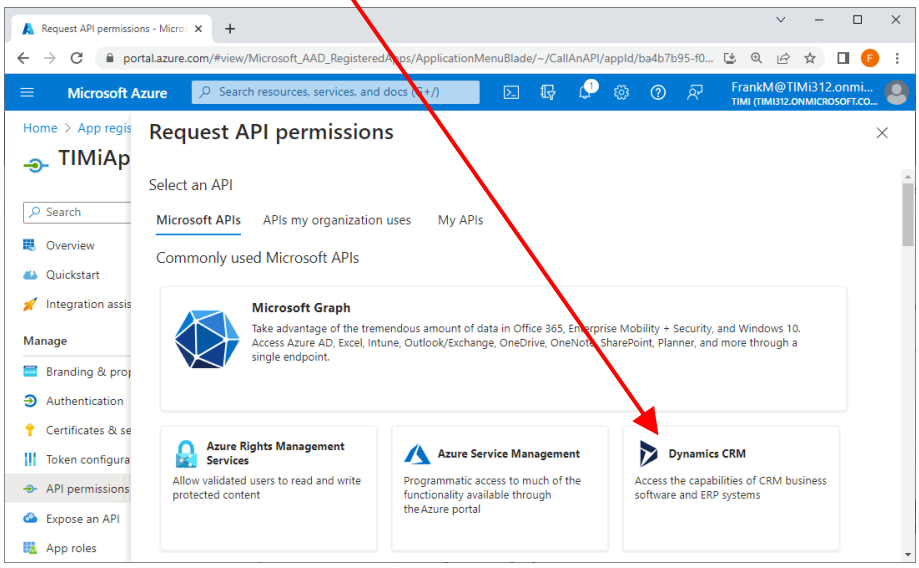
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9. **Grant Permission to access MS-Dynamics**

9.1. Click the "+ Add permission" button here:

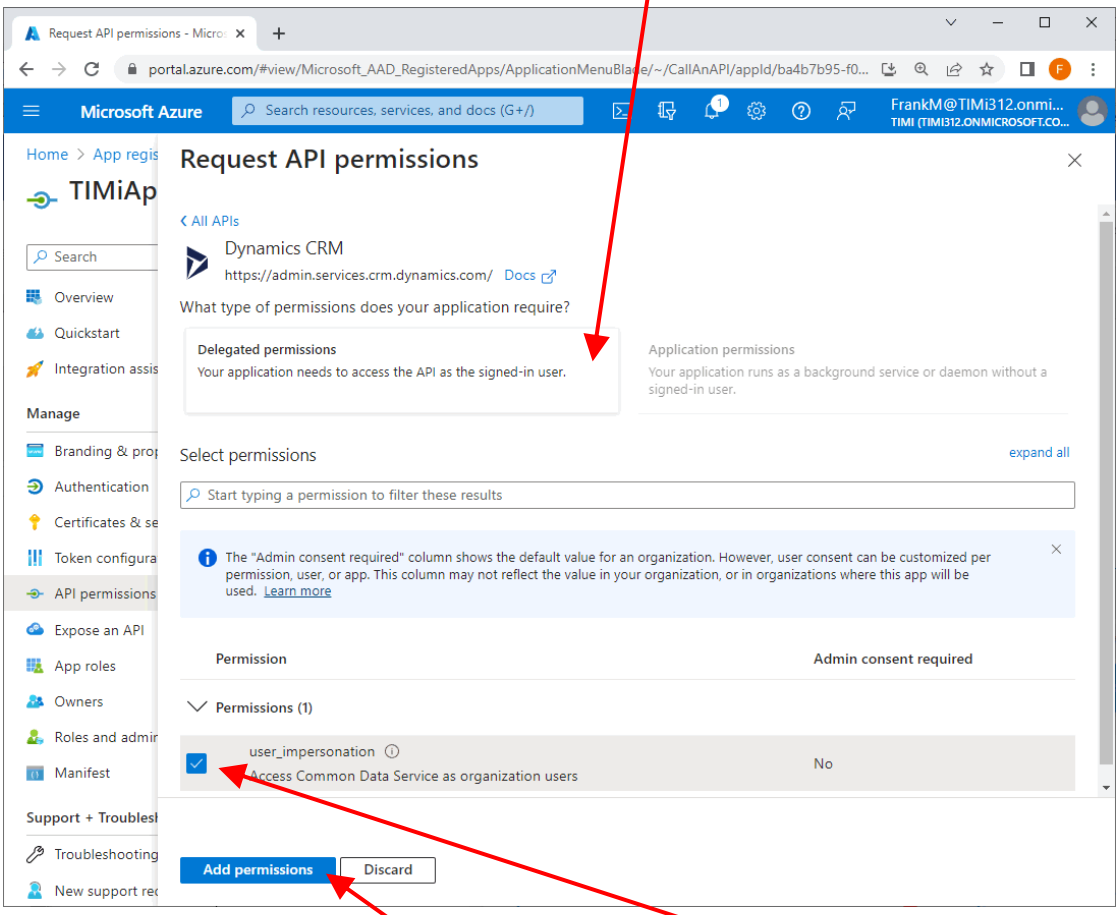


9.2. Click on "Dynamics CRM":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

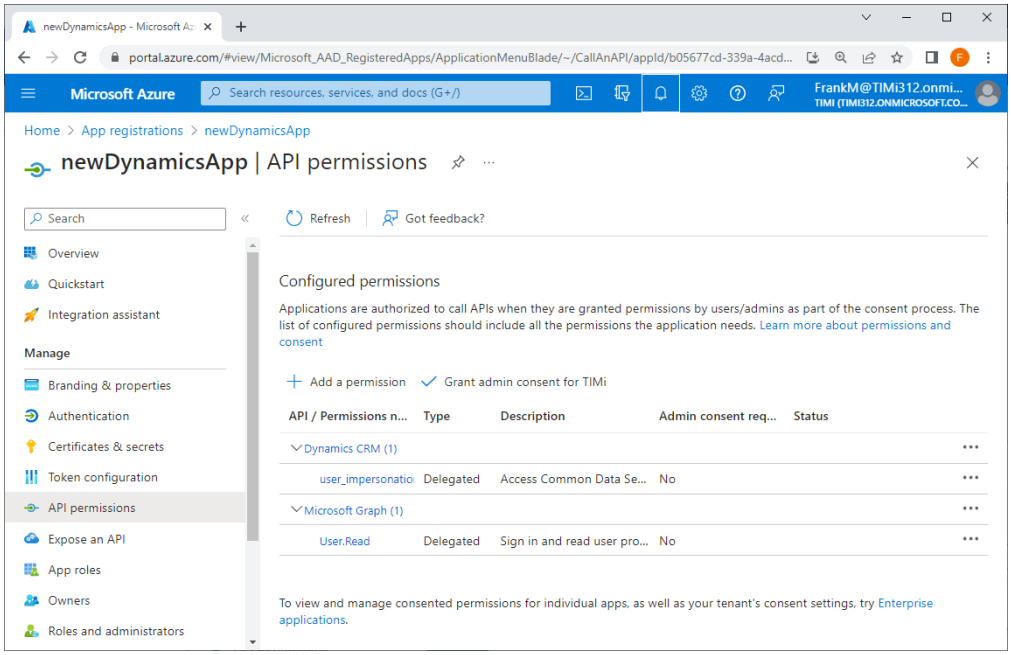
9.3. Click on the “Delegated Permissions” button:



9.4. Tick the checkbox named “user_impersonation” here:

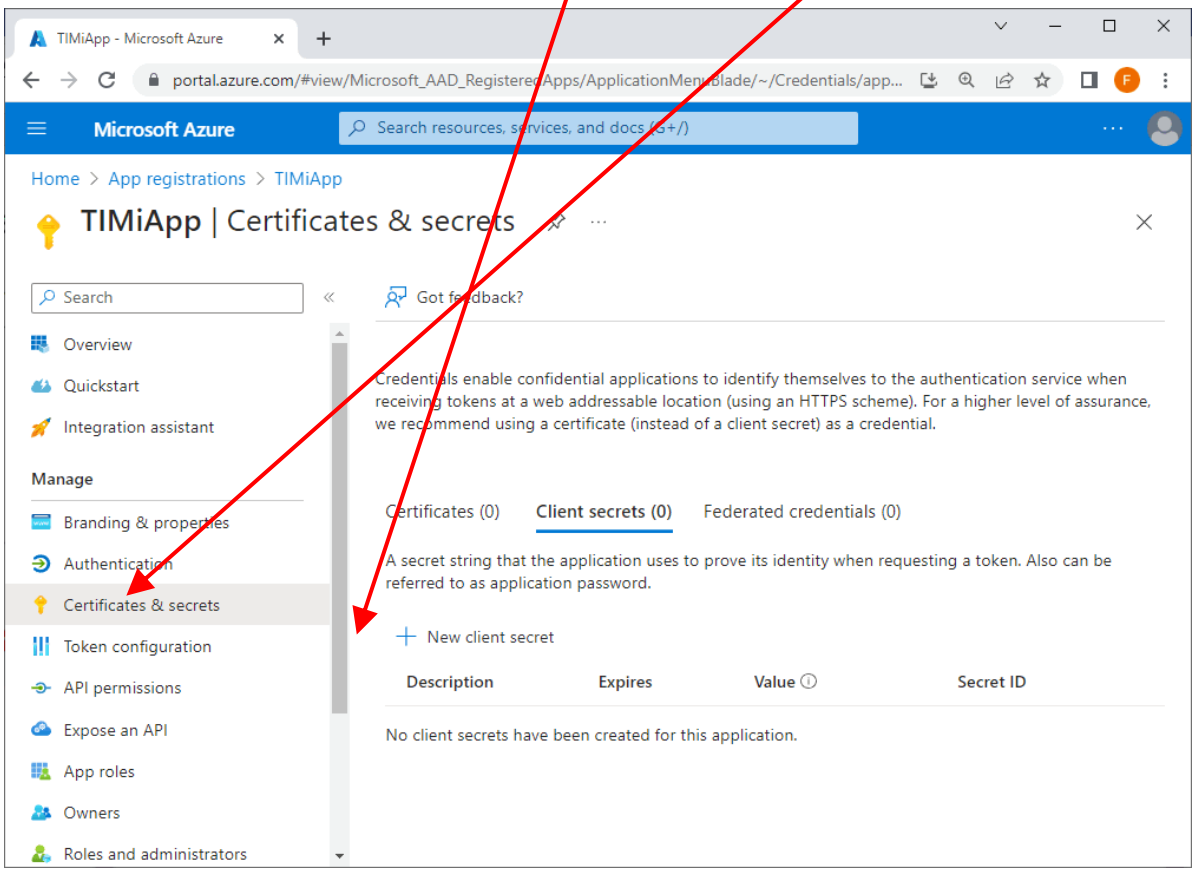
9.5. Click on the “Add permissions” button:

10. If you correctly enabled the required permissions, you should now see:

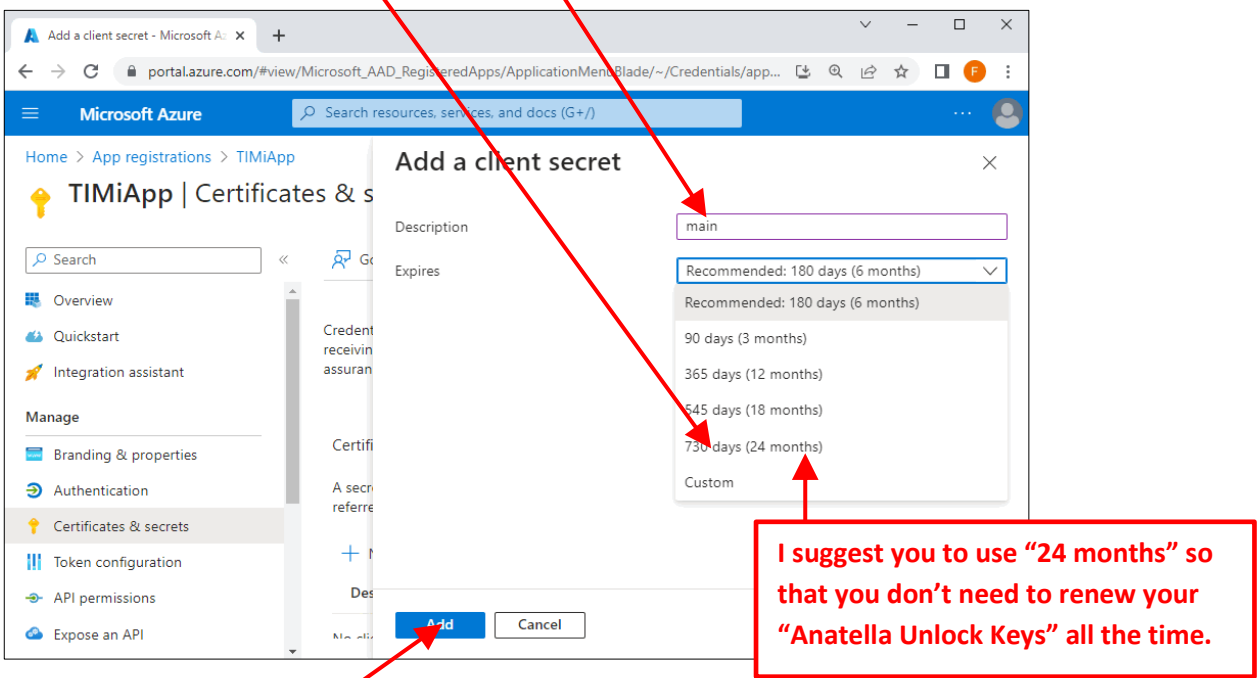


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

11. Get the parameter **P12** “Azure Client Secret”.
    11.1. Click on the section “Certificates and secrets” inside the Left-Menu
    11.2. Click on the “+ New client secret” button:



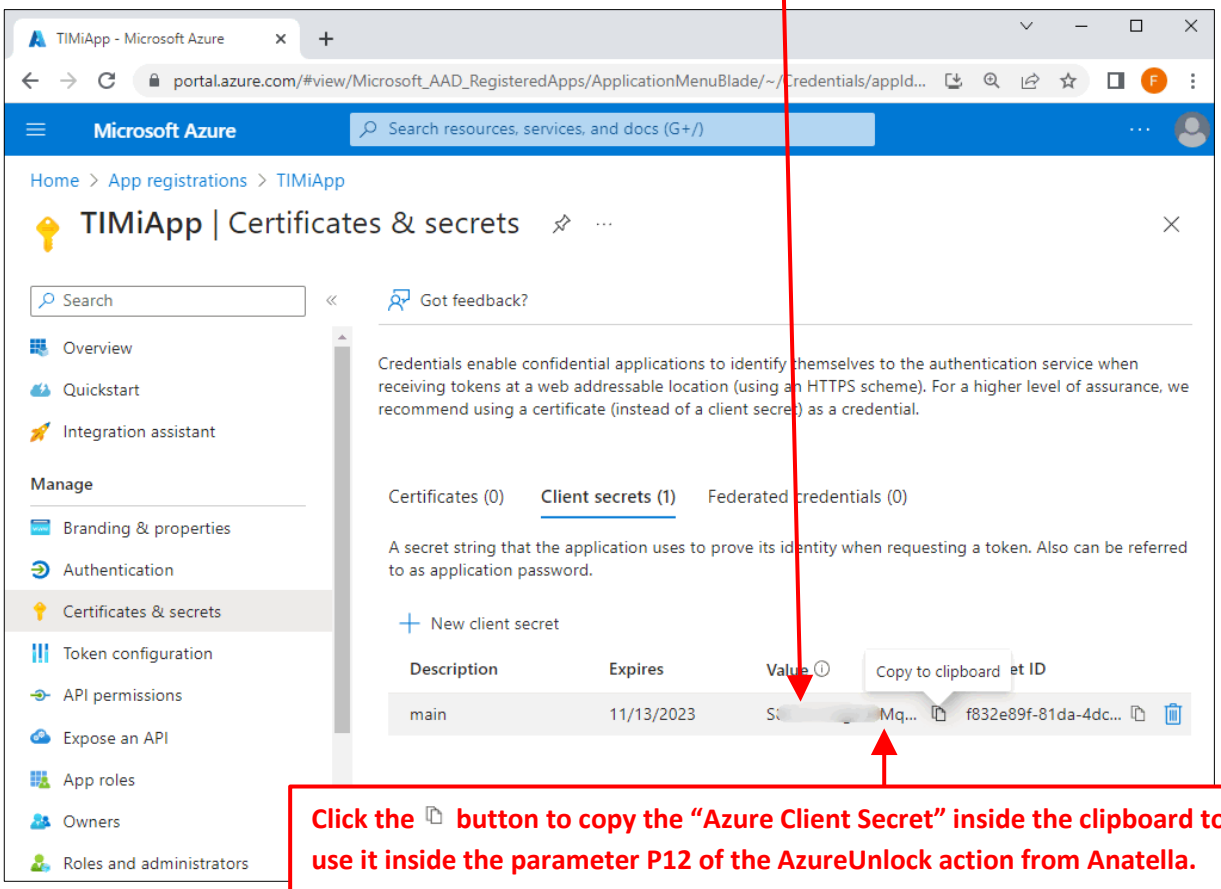
    11.3. Given any name for the description and select the maximum validity period of your “Anatella Unlock Key”:



    11.4. Click the “Add” button:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

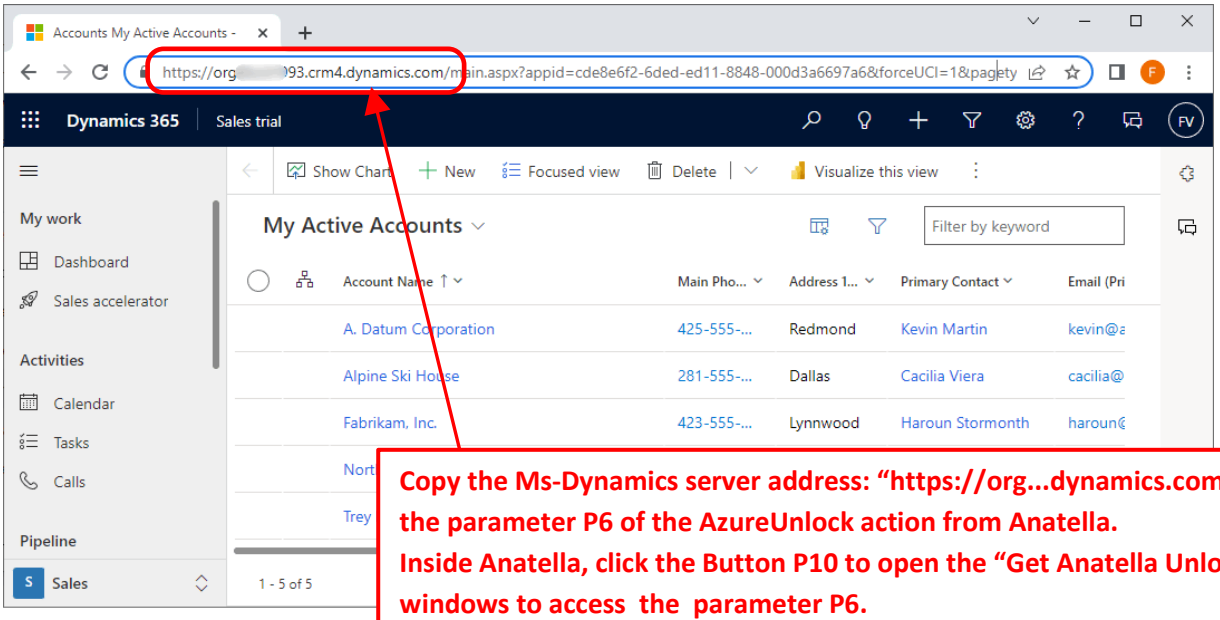
11.5. Finally, your parameter **P12** (“Azure Client Secret”) is here:



Click the button to copy the “Azure Client Secret” inside the clipboard to use it inside the parameter P12 of the AzureUnlock action from Anatella. Inside Anatella, click the Button P10 to open the “Get Anatella Unlock Key” windows to access the parameter 12.

11.6. There is no way to retrieve back the parameter **P12** (“Azure Client Secret”) if you lose it: i.e. Store this parameter preciously!

12. Open MS-Dynamics inside your Browser and look at the URL address of your MS-Dynamics server:



Copy the Ms-Dynamics server address: “https://org...dynamics.com” inside the parameter P6 of the AzureUnlock action from Anatella. Inside Anatella, click the Button P10 to open the “Get Anatella Unlock Key” windows to access the parameter P6.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

13. Proceed to the next section to get your first “Anatella Unlock key” to unlock the integration between Anatella and MS-Dynamics!

5.23.57.2. Get or Renew your Anatella Unlock Key for MS-Dynamics.

Follow the procedure given in this section to get your first “Anatella Unlock key” for MS-Dynamics!

Also: Each different Azure user that wants to access his own MS-Dynamics data must get his own personal “Anatella Unlock key” by following the procedure given here below.

Also, after a few months, your “Anatella Unlock key” will expire (this depends on the settings that you selected at step 20.3 from the previous section 5.23.27.1.). The maximum duration of an “Anatella Unlock Key” is 2 years (this is common to all tools accessing Azure since it’s imposed by Azure). When your “Anatella Unlock Key” expires, you just need to redo the easy procedure given below inside the current section.

The procedure to get (or renew) your “Anatella Unlock key” for MS-Dynamics is:

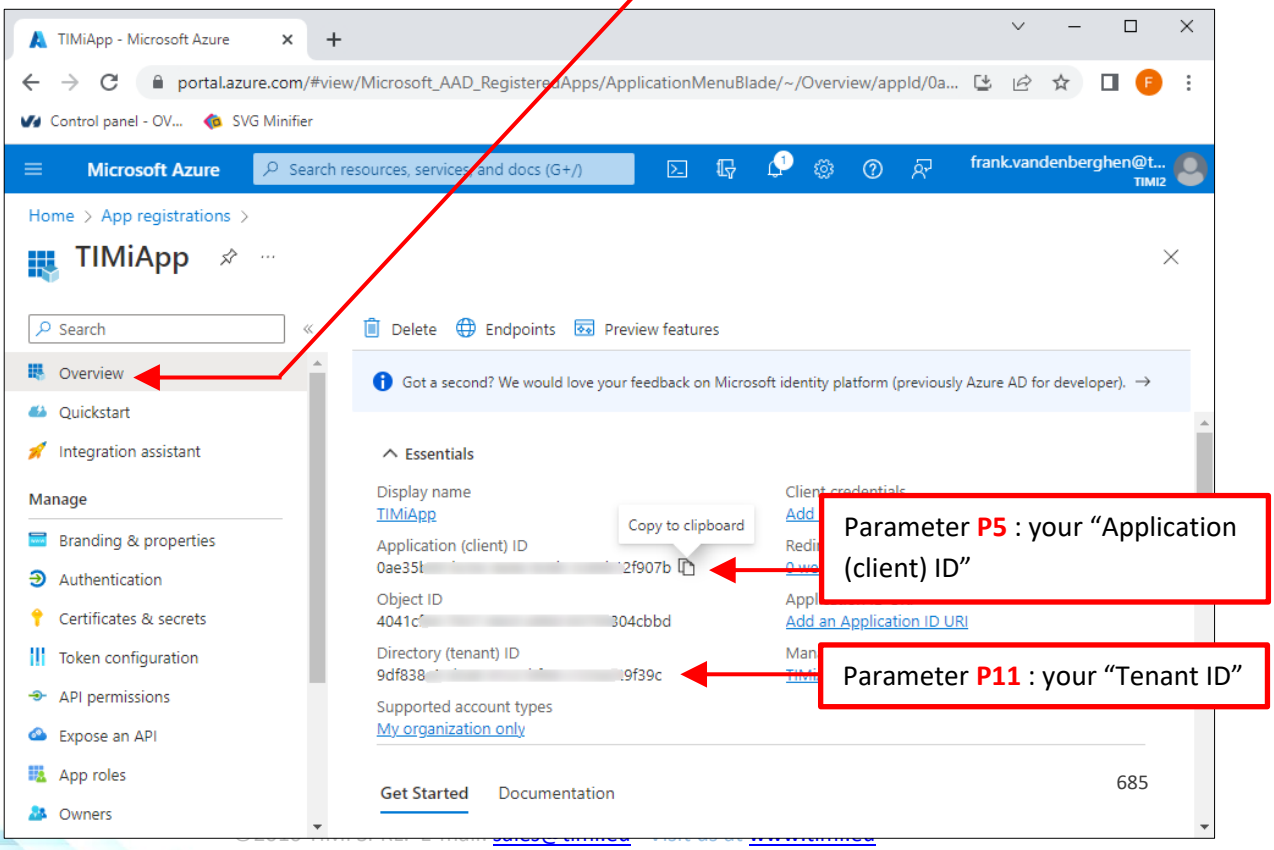
1. Go back inside Anatella. Place a “DyncamicsCRMGet” action inside your graph and open its properties. Inside the parameters of the “DyncamicsCRMGet” action, click the button Parameter **P10** to open the “Get Anatella Unlock Key” Window. Inside this window, you need to fill-in the parameters **P5**, **P11**, **P12** and **P6** using the values obtained while following the procedure from the previous section 5.23.57.1.

   More precisely:

   - The parameters **P5** and **P11** (i.e. your “Application (client) ID” and your “Tenant ID” were obtained during the step 7 of the procedure given in the previous section.

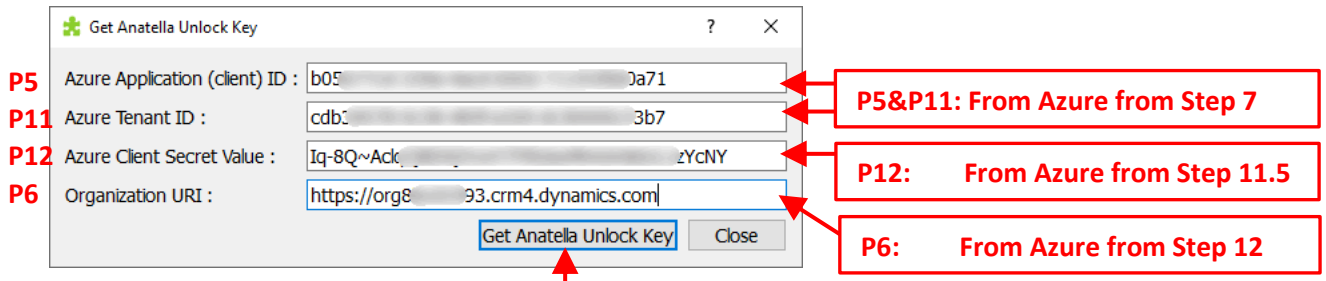
     Just in case: you can click on the

     button on the top of the left panel inside the Azure portal to see again these two parameters **P5** and **P11**:

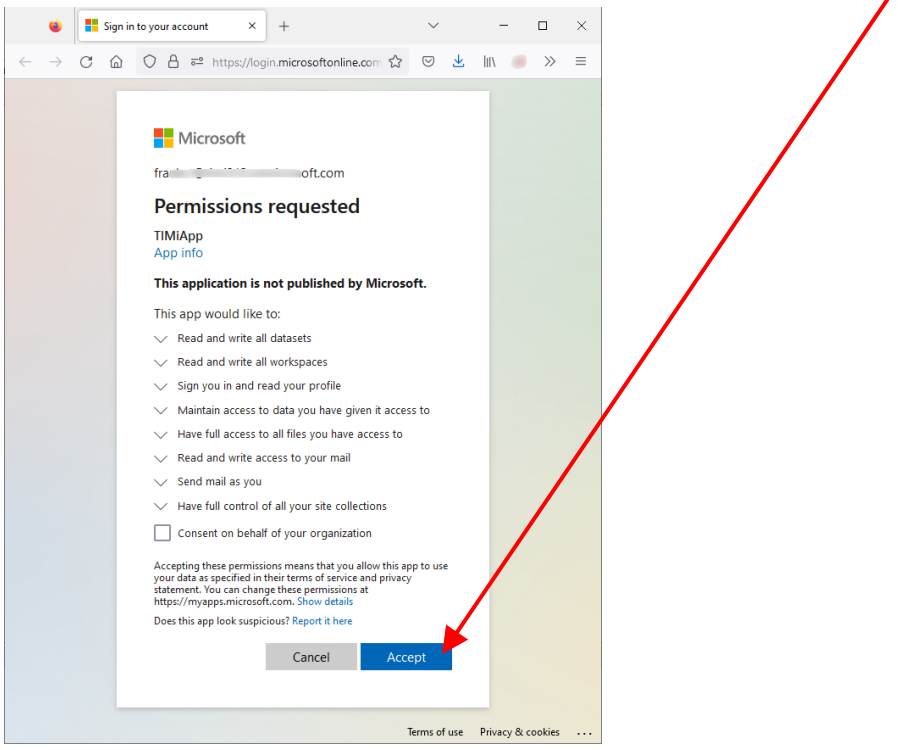


- The parameter **P12** (i.e. your "Azure Client Secret") was obtained during the step 11.5 of the procedure given in the previous section.
- The parameter **P6** (i.e. your "Organization URI") was obtained during the step 12 of the procedure given in the previous section.

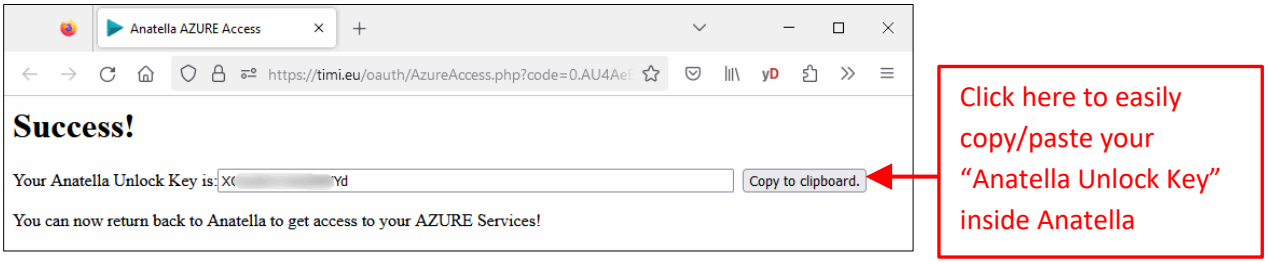
You should now see something like:



2. Click the button "Get Anatella Unlock Key": . An internet browser opens. Sign-in into the account that possesses the MS-Dynamics account that you want to access. Click on "Accept":

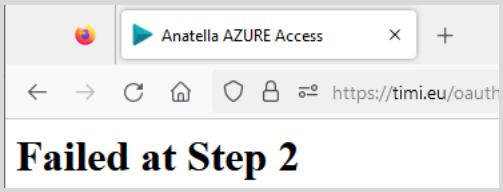


3. You finally get your "Anatella Unlock Key":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Sometime, instead of the "Success !" page (that you see here just above), you get:



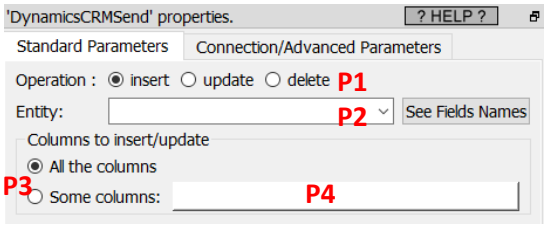
…followed by a bunch of garbage codes.

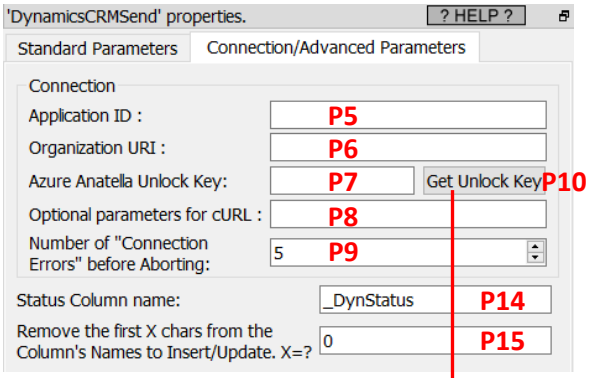
Don't worry: That's just Azure that is a little bit slow. Your new application settings need a little bit of time to propagate. Just wait a few seconds and click again the "Get Anatella Unlock Key" button (parameter **P13**) and, at some point, it will give you the required "Anatella Unlock Key".

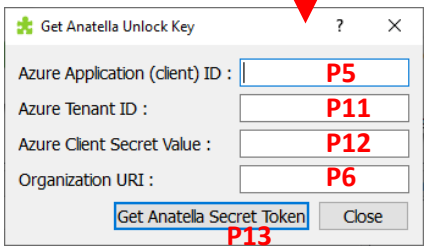
5.23.58. Send data to Microsoft Dynamics CRM

Icon:

Property window:







Short description:
Upload Data to Microsoft Dynamics CRM

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the Azure Website (i.e. you need the parameters **P5**, **P6** and **P7**). Please see the sections 5.23.57.1 and 5.23.57.2 for more details on how to get these parameters.

Once you have completed the "setup process" described in the sections 5.23.57.1 and 5.23.57.2, you can use the parameters **P1** to **P4** to insert/delete/update data to your Microsoft Dynamics CRM system.

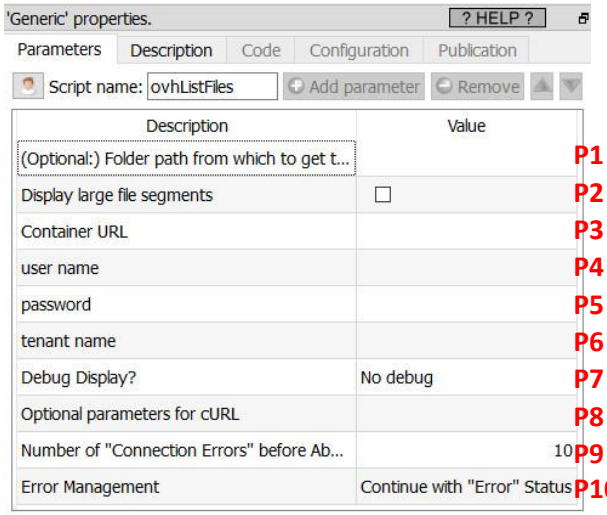
The column's names inside the input table of the DynamicsCRMSend action in Anatella must match the column's names of the table inside the Microsoft Dynamics CRM system. To help you make a correct match, you can use the parameter **P15** and the button "See Fileds Names".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.59. OVH Cloud Object Storage List Files

Icon:

Property window:



Short description:
List files in a container in an OVH Cloud Object Storage

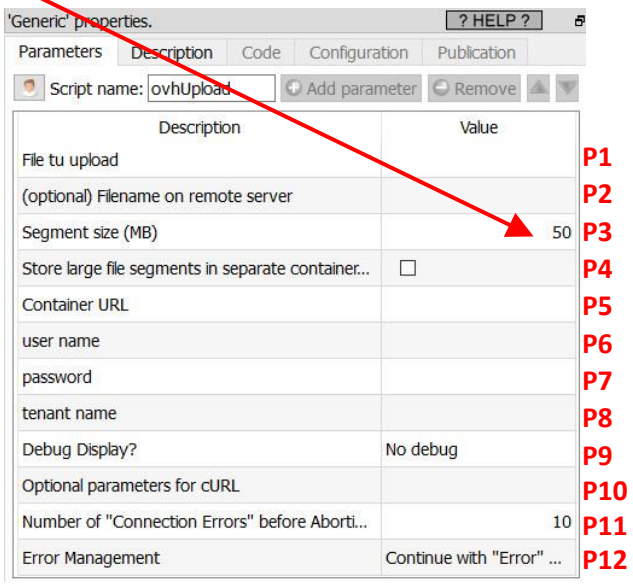
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the OVH Website (i.e. you need the parameters **P3**, **P4**, **P5** and **P6**). Please see the section 5.23.59.2. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.59.2., you can use the parameter **P1** and **P2** to list the files in your OVH Cloud Object storage system.

5.23.59.1. OVH Cloud Storage particularities

The OVH Cloud object storage has some very special particularities when it comes to manipulating large files. What's a "*large*" file? That's a file that is larger than the parameter **P3** ("Segment Size") from the "ovhUpload" action:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

What's happening when you upload a file that is larger than the segment size? (i.e. when you upload a file that is larger than 50MB, in the example above). For example, let's see what's happening if we upload a 120MB file that is named "big.txt". The following happens:

- The "large" file named "big.txt" is splitted in many 50MB files.
  Each of these 50MB file is named a "segment".
- Each segment is uploaded to the OVH Cloud object storage.
  Thus, we'll have inside the OVH Cloud object storage, 3 file-segments that are:

Filename of the segment in the cloud storage	Size
big.txt|1	50 MB
big.txt|2	50 MB
big.txt|3	20 MB

- Anatella uploads to the OVH Cloud object storage a "Manifest File" named "big.txt" that simply announce that the 3 file-segments "big.txt|1", "big.txt|2" and "big.txt|3" are actually the different parts of a single 120 MB "logical" file that is named "big.txt".

From now on, if you attempt to download back from the OVH cloud the file named "big.txt", Anatella will transparently download a 120MB file (that is the concatenation of all the segment-files listed inside the "Manifest File" named "big.txt").

If you use the OVH Cloud List File Action to list the files inside your OVH Cloud storage, you might see these 4 files:

Filename in the cloud storage	Size
big.txt	120 MB
big.txt|1	50 MB
big.txt|2	50 MB
big.txt|3	20 MB

For many use-cases, we actually don't want to see the individual file-segments that are composing the "large" files (e.g. we don't want to see "big.txt|1", "big.txt|2" and "big.txt|3"). So, when listing the files inside the OVH Cloud storage, there are 2 solutions to "hide" the individual file-segments:

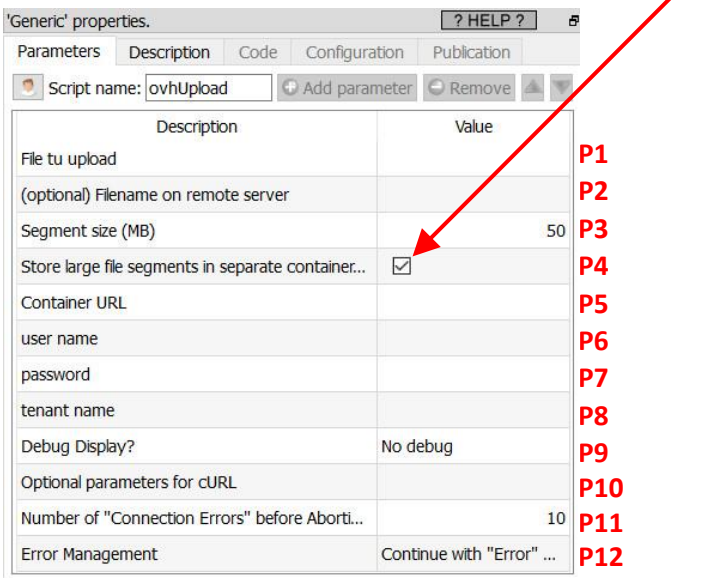
- **Solution 1 (the simplest solution)**
  You can uncheck the parameter **P2** of the OVH Cloud List File Action. This will remove from the display all the files with a filename that ends with the character "|" followed by a number, so that we finally see:

Filename in the cloud storage	Size
big.txt	120 MB

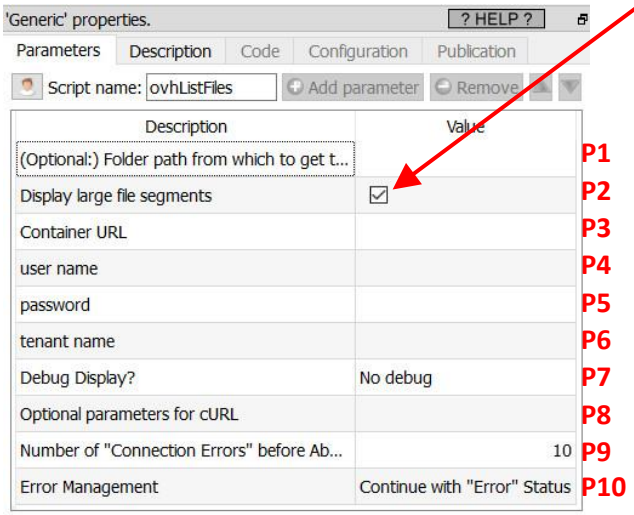
  This solution has one small disadvantage: i.e. it's not always 100% accurate. For example, this solution will remove from the output filename table a file that is named "myfile.txt|9" despite the fact that this file is a perfectly valid file (i.e. it's not a segment-file). The removal occurs just because this filename unfortunately ends with the character "|" followed by a number (i.e. it just has the same "look" as a segment-file).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- **Solution 2 (the more complex solution)**
  This second solution involves using two different OVH Cloud object containers. The first container contains all the files, with the exception of the segment files. The second container only contains the segment files. For this solution to work, you need the following:
    - The name of the second container (that only contains the segment files) must be the name of the first container followed with "_segments". For example, if the name of the first container is "*my_container*", then the name of the second container is "*my_container_segments*".
    - You must check the parameter **P4** inside the "ovhUpload" action:



    - You must check the parameter **P2** inside the "ovhListFiles" action:



    This solution is better because it's always 100% accurate: i.e. there is no risk of "hiding" a valid, non-segment file: You just display ALL the files inside the first container and you are good! 😊

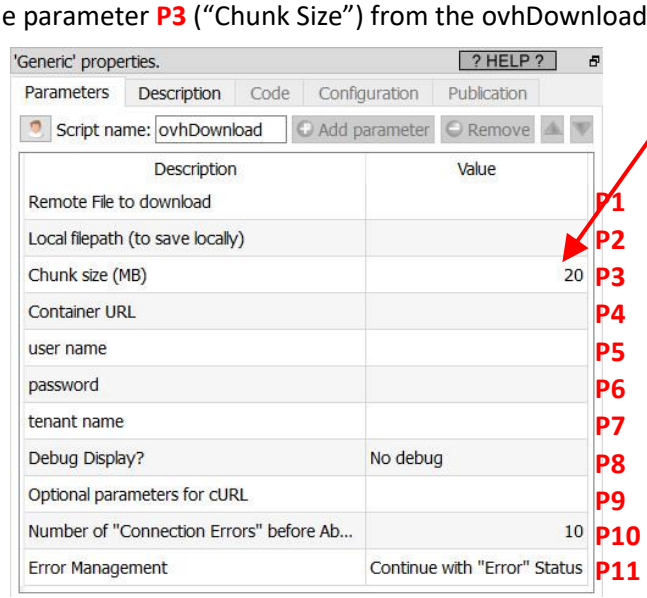
All the Anatella OVH Cloud Storage actions are built to handle flawlessly both the first and the second solution described here above. In particular, if you are using the second solution, this means that:

- When Anatella must download the file "big.txt" (this file is actually just a small "manifest file" that is stored in the first container), Anatella will actually transparently connect to the second container to download the appropriate file-segments.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- When Anatella must delete the file “big.txt” (that is stored in the first container), it will actually also connect transparently to the second container to delete from this second container the appropriate file-segments.
- Everything is totally transparent for the final user: i.e. it looks like all the files are simply stored inside the first container.

One final note about the parameter **P3** (“Chunk Size”) from the ovhDownload Action:



This parameter **P3** (“Chunk Size”) from the ovhDownload Action is usefull when your internet connection is unstable. Anatella always downloads your files chunk-by-chunk. If the internet connection is lost during the download of one chunk, Anatella will “*throw away*” the incomplete/corrupted chunk and attempt to re-download again the same chunk (the number of “retries” is defined in parameter **P10**). A small “chunk size” means that, when the internet connection is lost, we don’t have to “*throw away*” a large quantity of bytes. This means that, when your internet connection is bad, you should decrease the value of the “chunk size”.

What’s happening if we want to download the 120 MB file named “big.txt” from the OVH Cloud storage and the parameter **P3** (“Chunk Size”) is 20 MB? At first sight, we expect Anatella to run 6 downloads to download “big.txt” (since the file-size is 120MB and the chunk-size is 20 MB, we should have 120/20=6 downloads) but Anatella actually runs 7 downloads:

#	Donwload from	Download size	Size of the local “big.txt” file after the completion of the download
1	big.txt|1	20 MB	20 MB
2		20 MB	40 MB
3		**10** MB	50 MB
4	big.txt|2	20 MB	70 MB
5		20 MB	90 MB
6		**10** MB	100 MB
7	big.txt|3	20 MB	120 MB

We have 7 downloads (instead of the expected 6) because the downloads #3 and #6 are only 10 MB (instead of 20 MB) because we used a 50MB segment-size during the “upload”. So, to avoid un-necessary extra downloads, it’s better to use a segment-size that is a multiple of the chunk-size.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

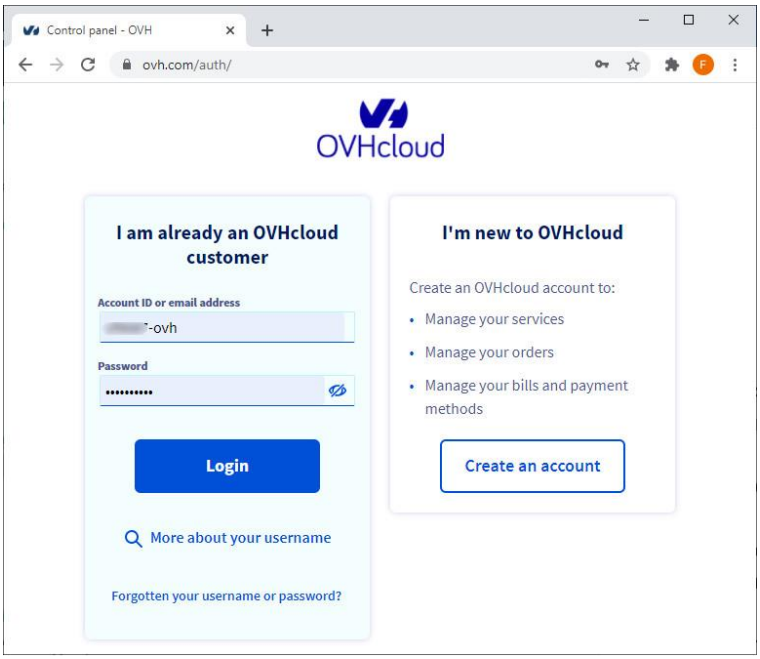
5.23.59.2. OVH Cloud Storage first time setup

Before using the OVH Cloud Object storage Actions inside Anatella, you need to get the 4 parameters:

- User name (Parameter **P4**)
- Password (Parameter **P5**)
- Tenant Name (Parameter **P6**)
- Container URL (Parameter **P3**)

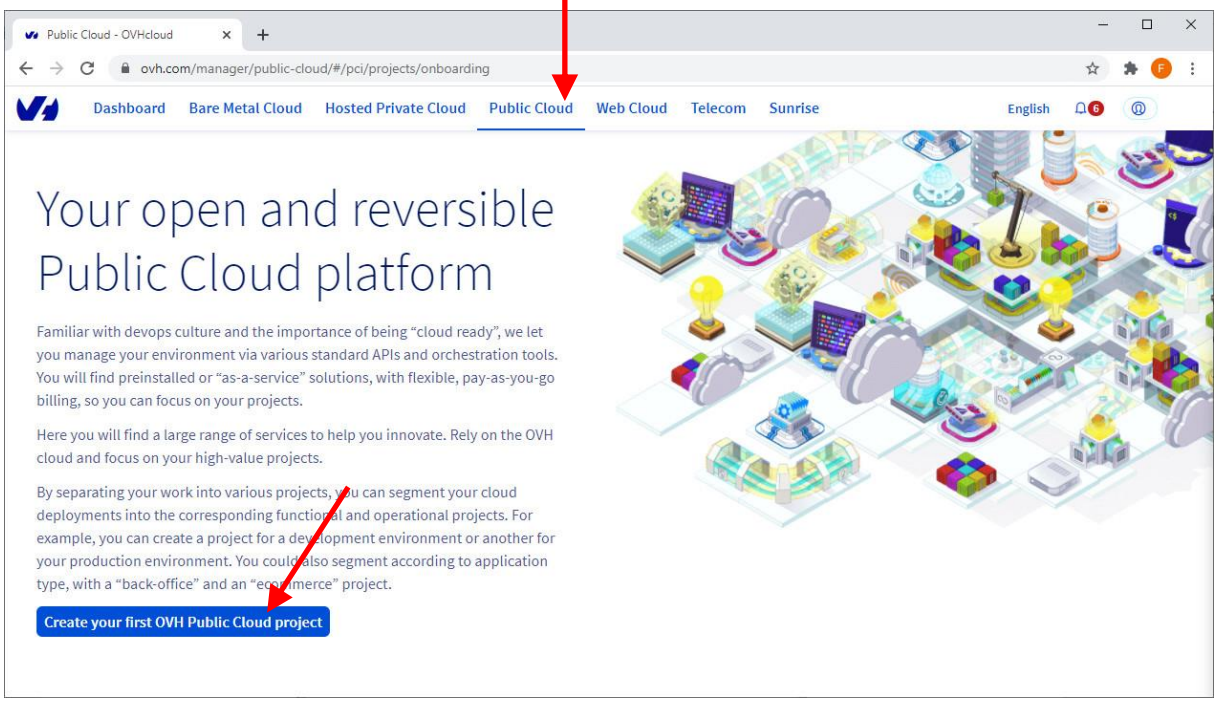
…from the OVH website. Here are the steps to get these 4 parameters:

1. Open the URL https://www.ovh.com/auth/ and "log-in" using your normal "Login" and "Password" for OVH:

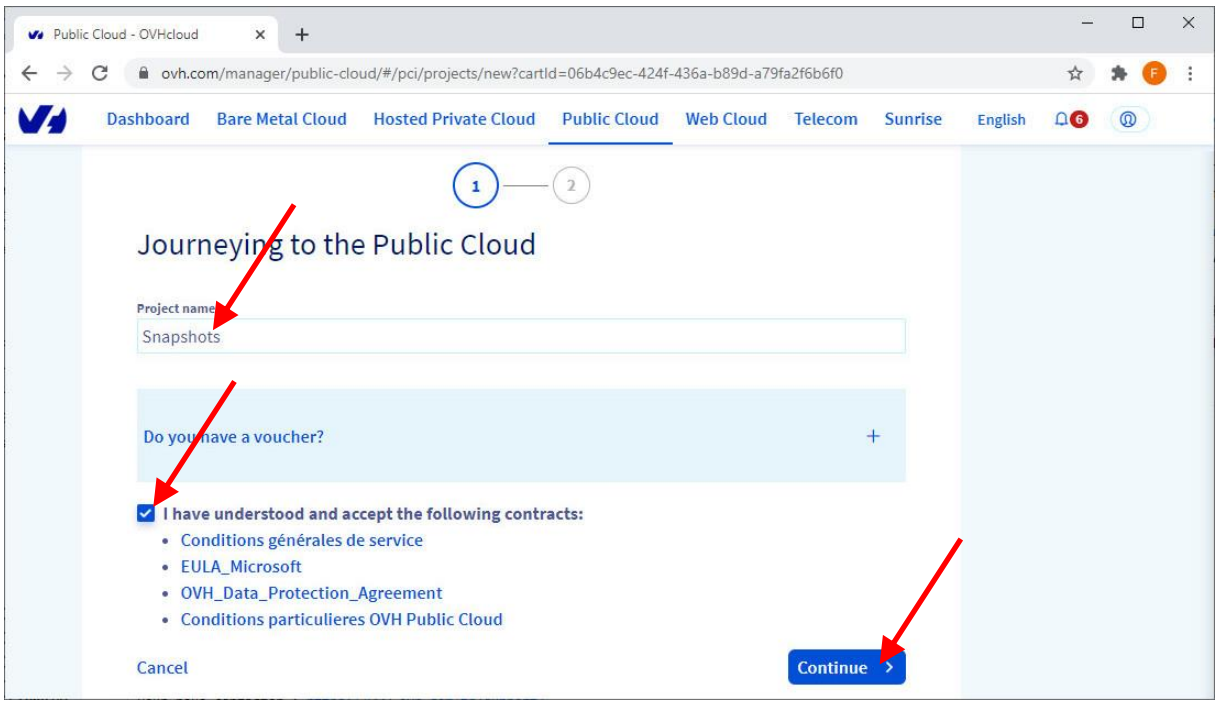


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. If this is the first time that you are using the OVH Cloud Object storage:
   3.7. Create an OVH project
   - In the top menu, click on “Public Cloud”, and then click on the “Create ..project” button:

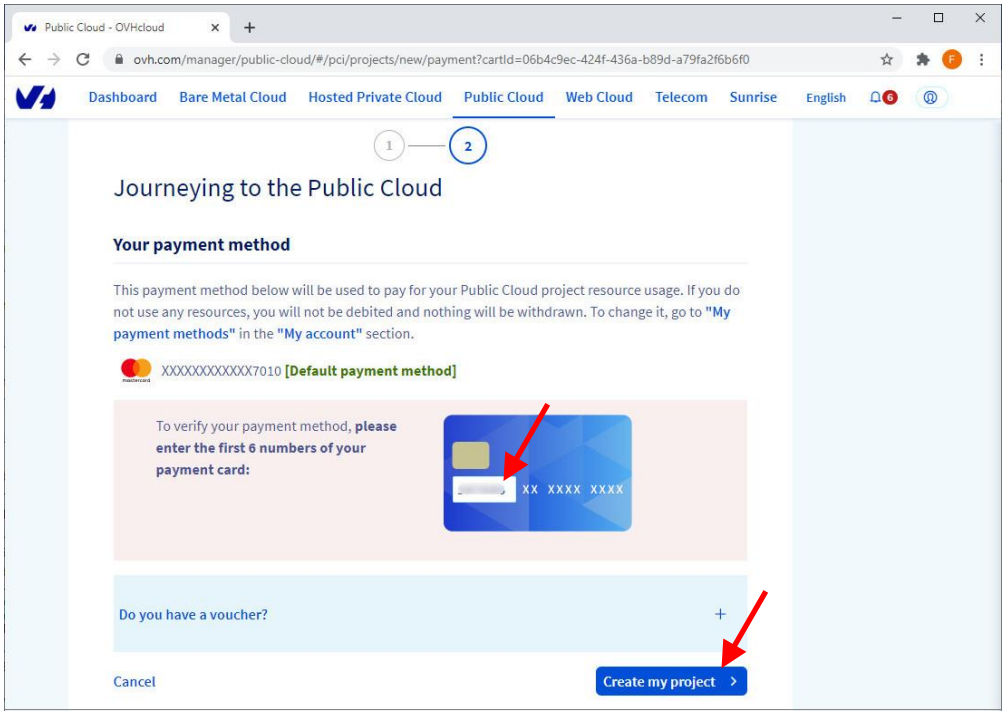


   - Enter a project name (in the screenshot below, we used “snapshots” but you can use any name), check the checkbox and click “Continue”:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

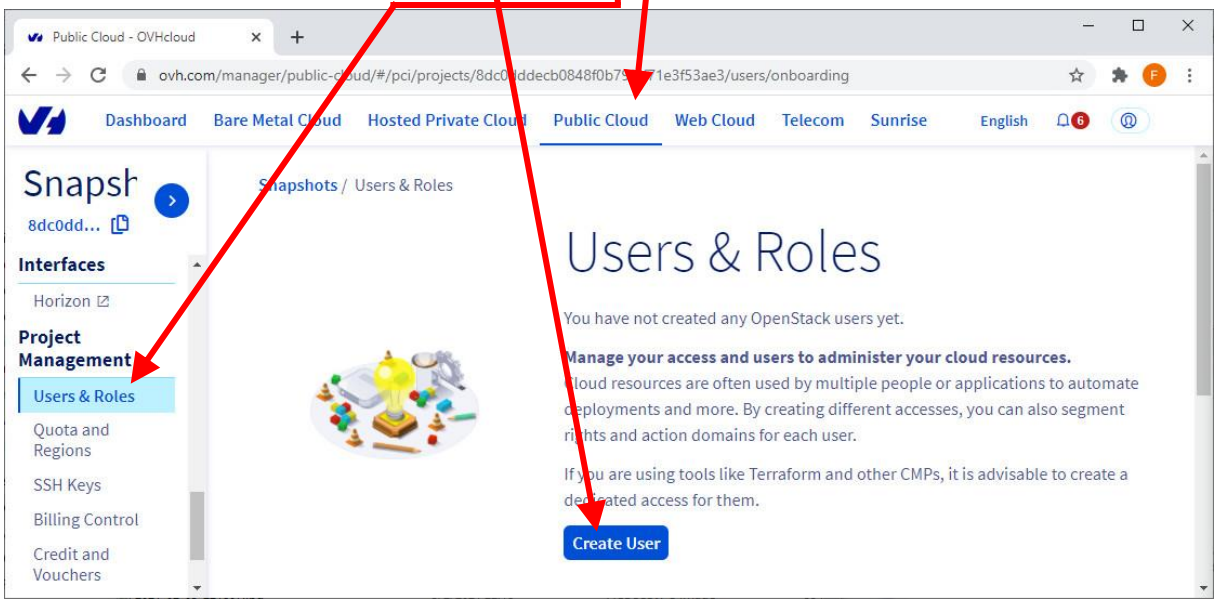
- Confirm your payment method and click on the “Create my project” button:



- After a few seconds, your project is created.

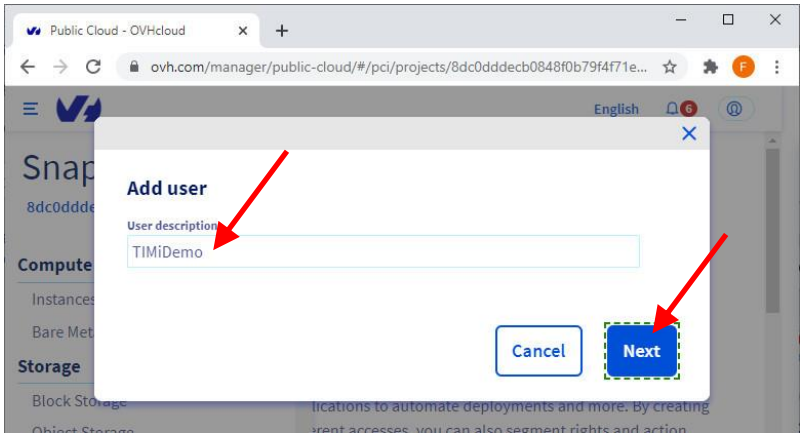
3.8. Create a new user:

- In the top menu, click on “Public Cloud” :
- In the left menu, click on “Users & Roles”:
- Click the “Create User” button:

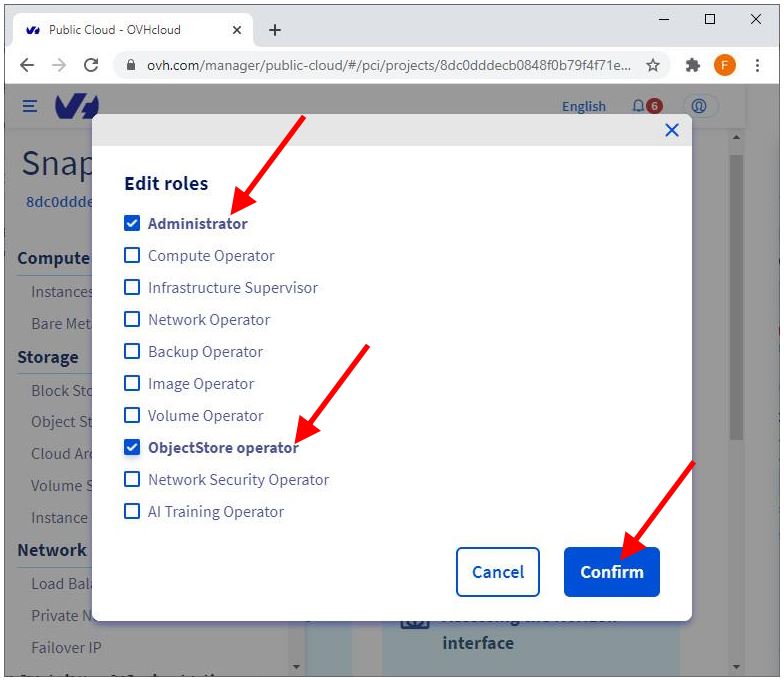


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

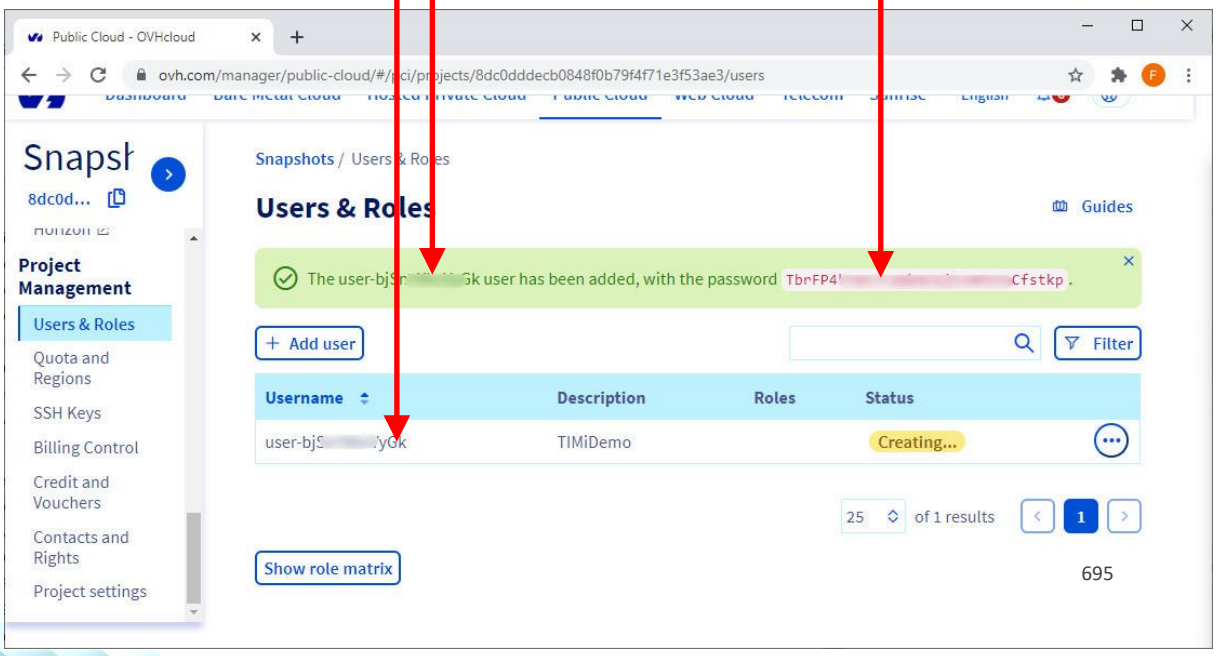
- Give a name to the new user and click the “Next” button:



- Give the “Administrator” role and “ObjectStore Operator” role to this new user:

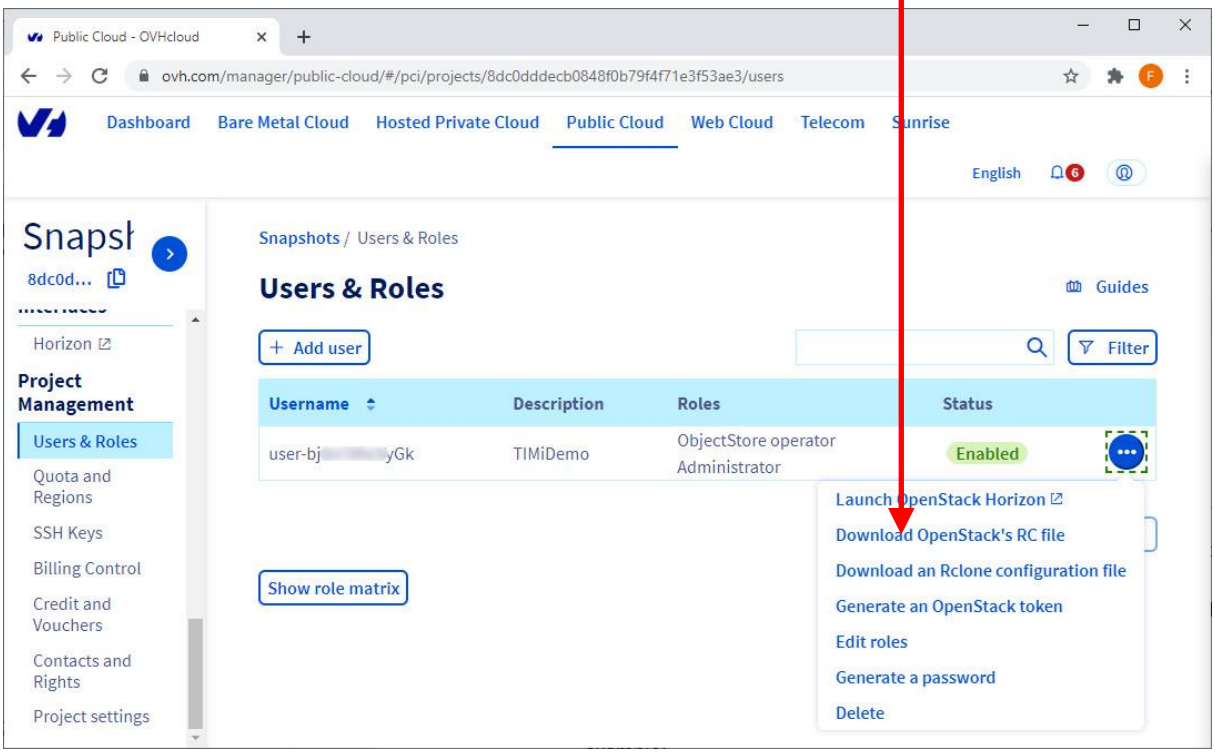


- You finally get your new user.
  At this point, please note carefully the Anatella parameter **P5** (password) and the Anatella parameter **P4** (user name).

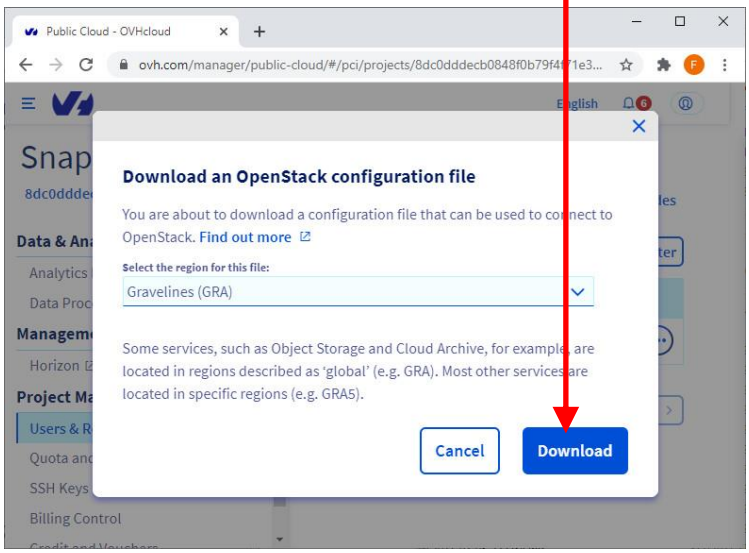


You should store very carefully the above password since there is no way to retrieve it later (you can generate a brand-new password but not retrieve an old one).

- Click on the ••• icon and select "Download Openstack's RC file":

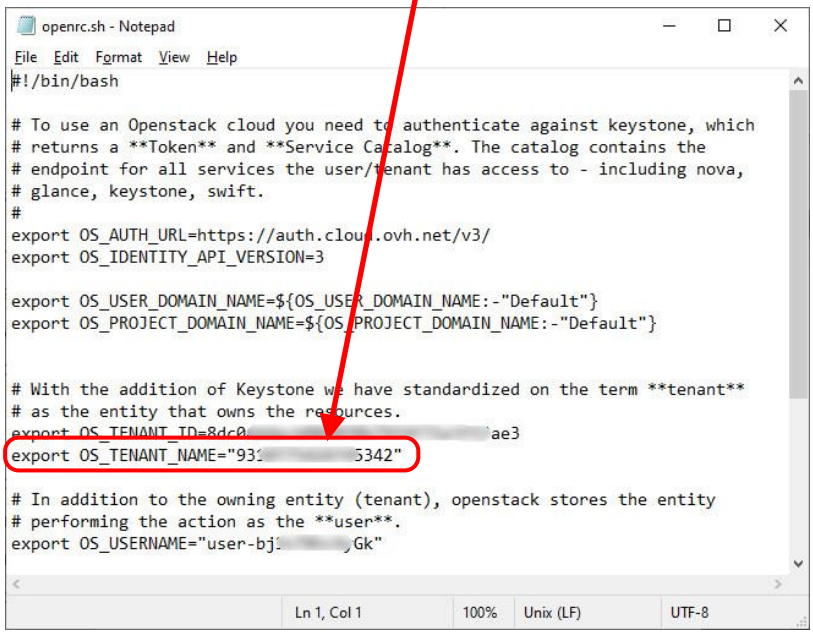


- Select the Data Center that hosts your files and click the "Download" button:



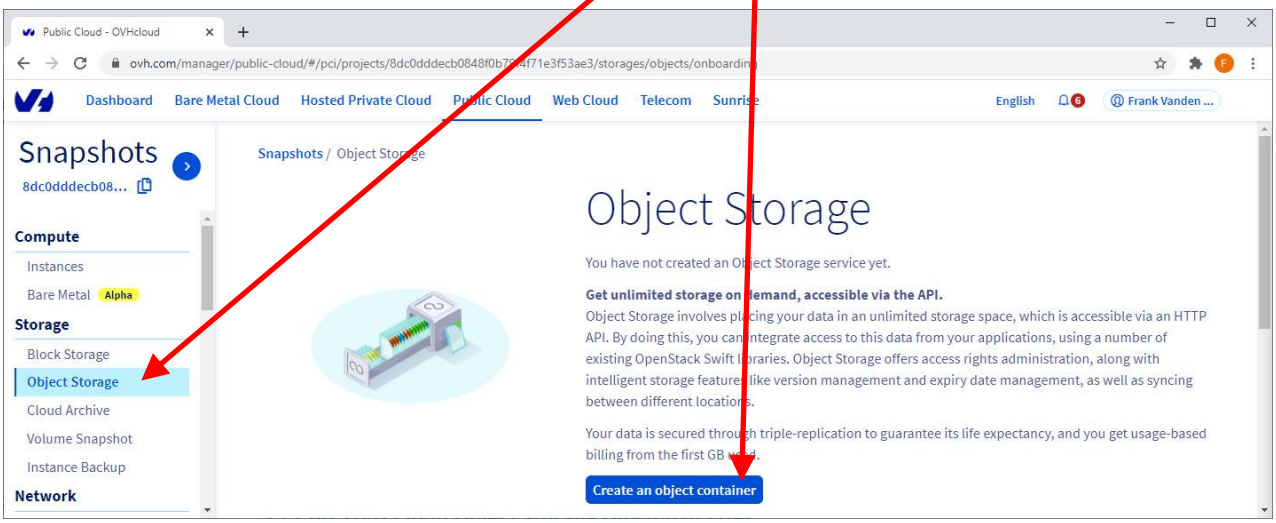
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Open the downloaded file (“openrc.sh”) inside notepad.
  The Anatella parameter **P6** (tenant name) is visible here:



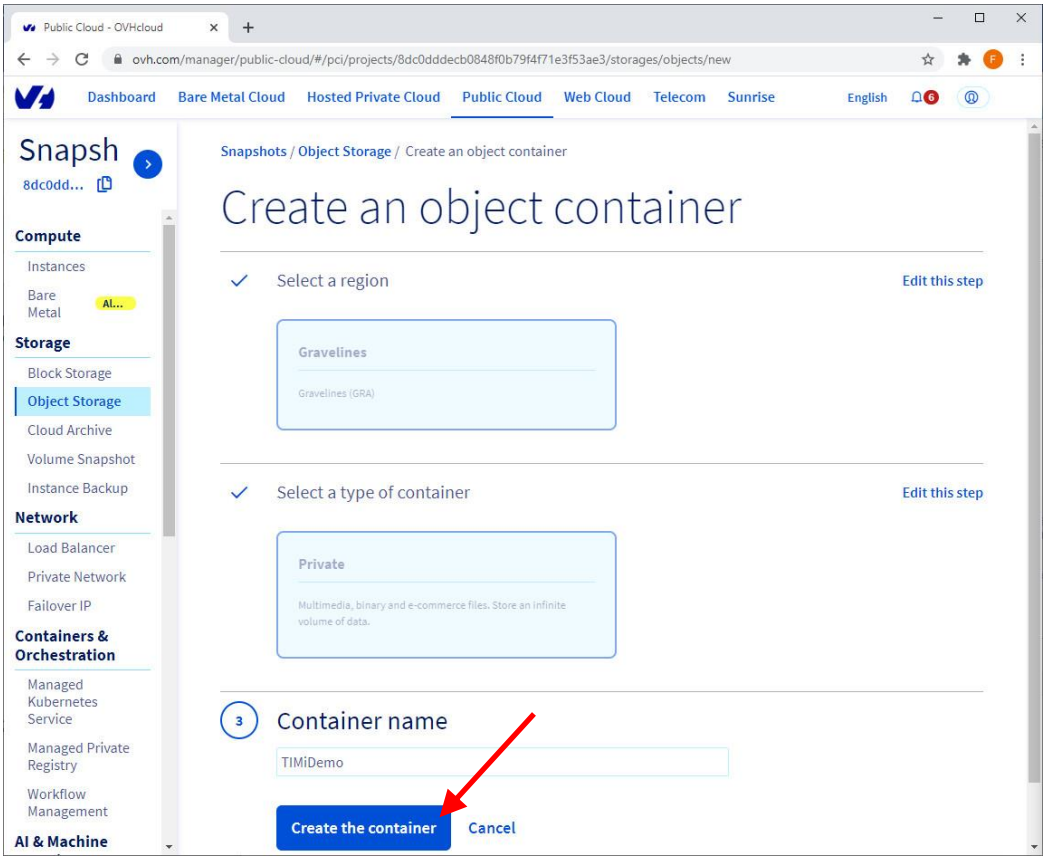
3.9. Create a new Container:

- In the left menu, click on “Object Storage”:
- Click on the “Create an object container” button:

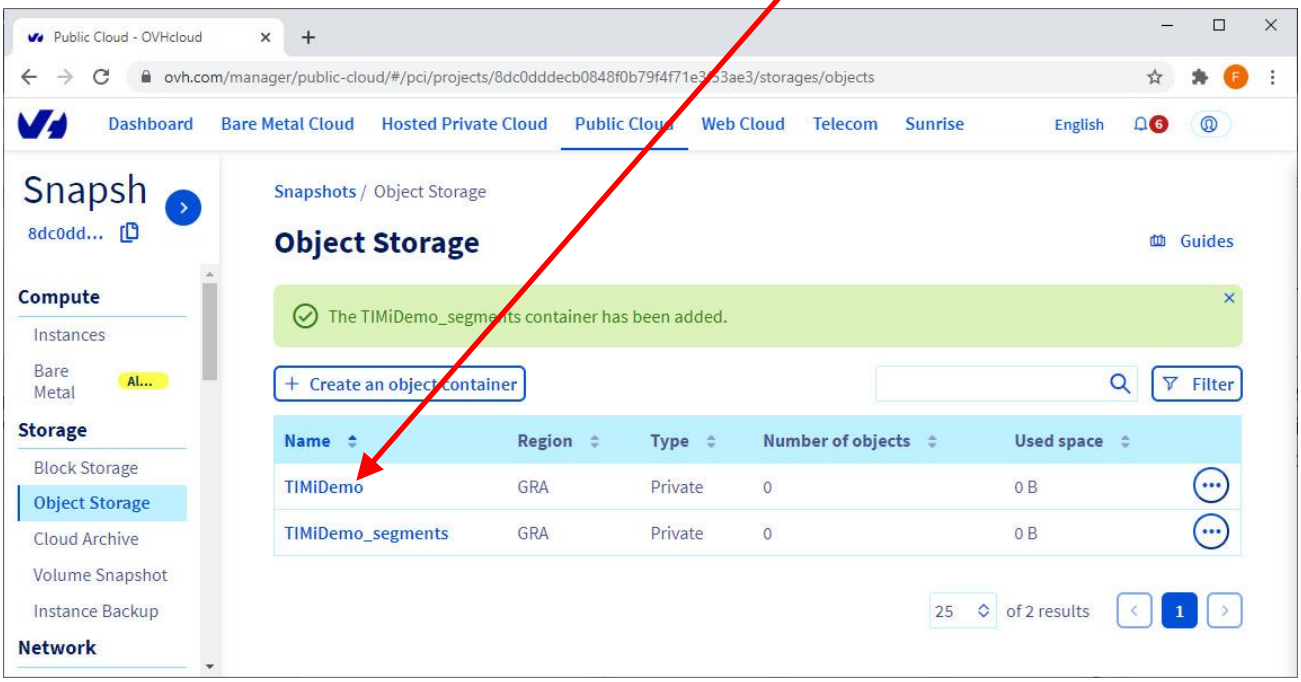


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Select the Data Center("Region") that will hold your files, Select the "type of container", give a name to the container and, finally, click the "Create Container button":

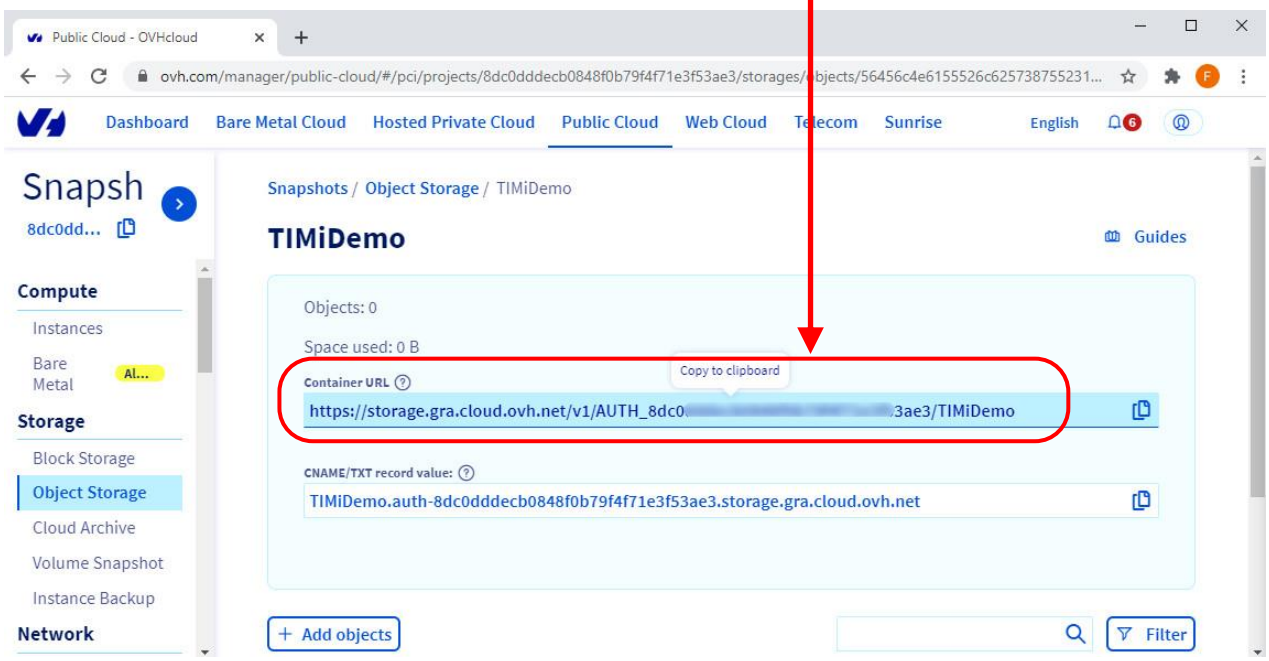


- You have now a brand-new container.
  If you opted for the "*Solution 2 (the more complex solution)*" described in the previous section 5.23.59.1., you'll need to create a second container whose name ends with "_segments". We did exactly that (i.e. we created a second container) in the screenshot below. Click now on your (first) container:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- The Anatella parameter **P3** (Container URL) is visible here:

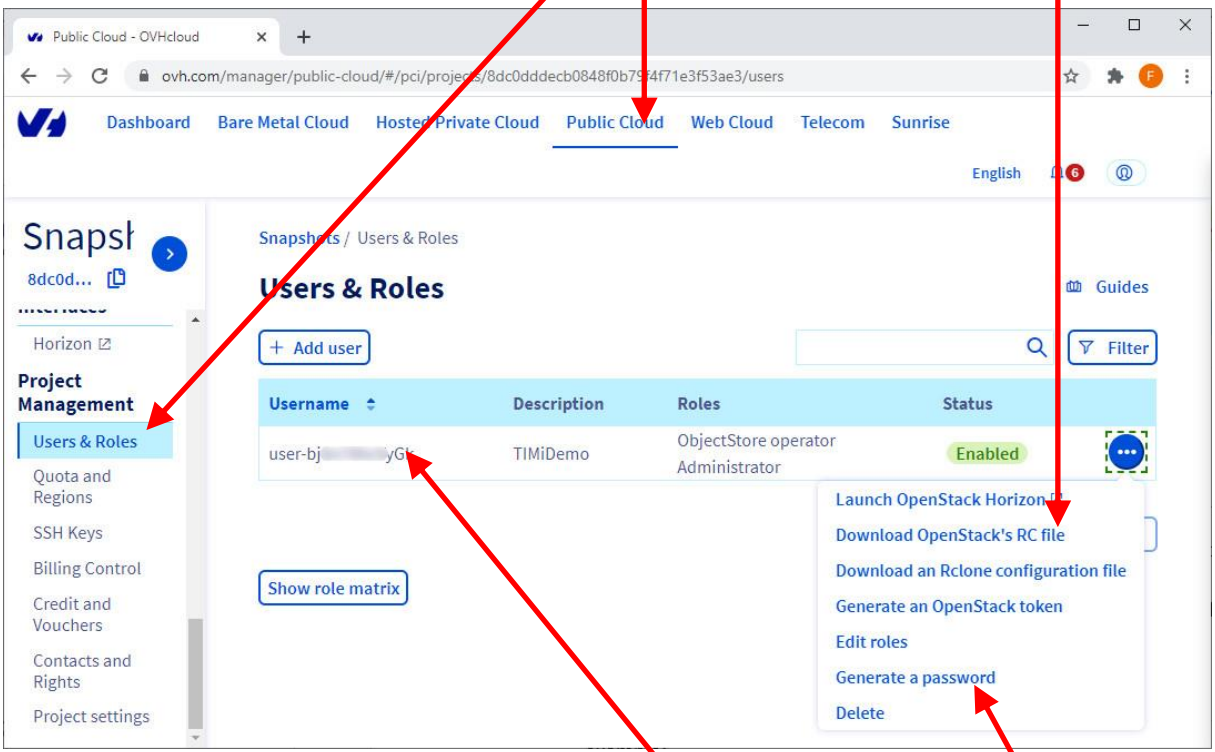


You now have the 4 required parameters to connect to your OVH Cloud Object Storage.

4. You want to connect to an already existing OVH Cloud Object storage:

4.7. Get your credentials:

- In the top menu, click on “Public Cloud” :
- In the left menu, click on “Users & Roles”:
- Click on the ••• icon and select “Download Openstack’s RC file”:

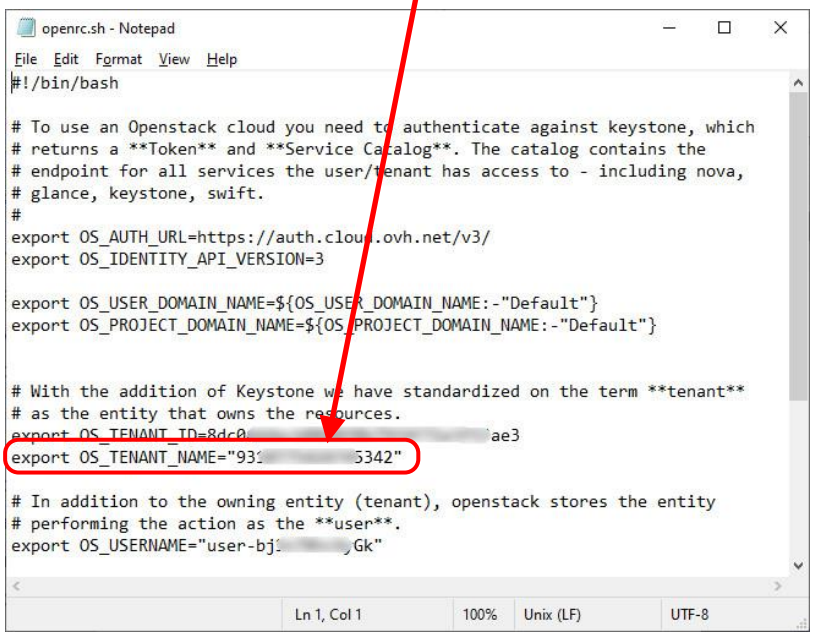


- The Anatella parameter **P4** (user name) is here:
- If you lost your parameter **P5** (password), you can create a new password here

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

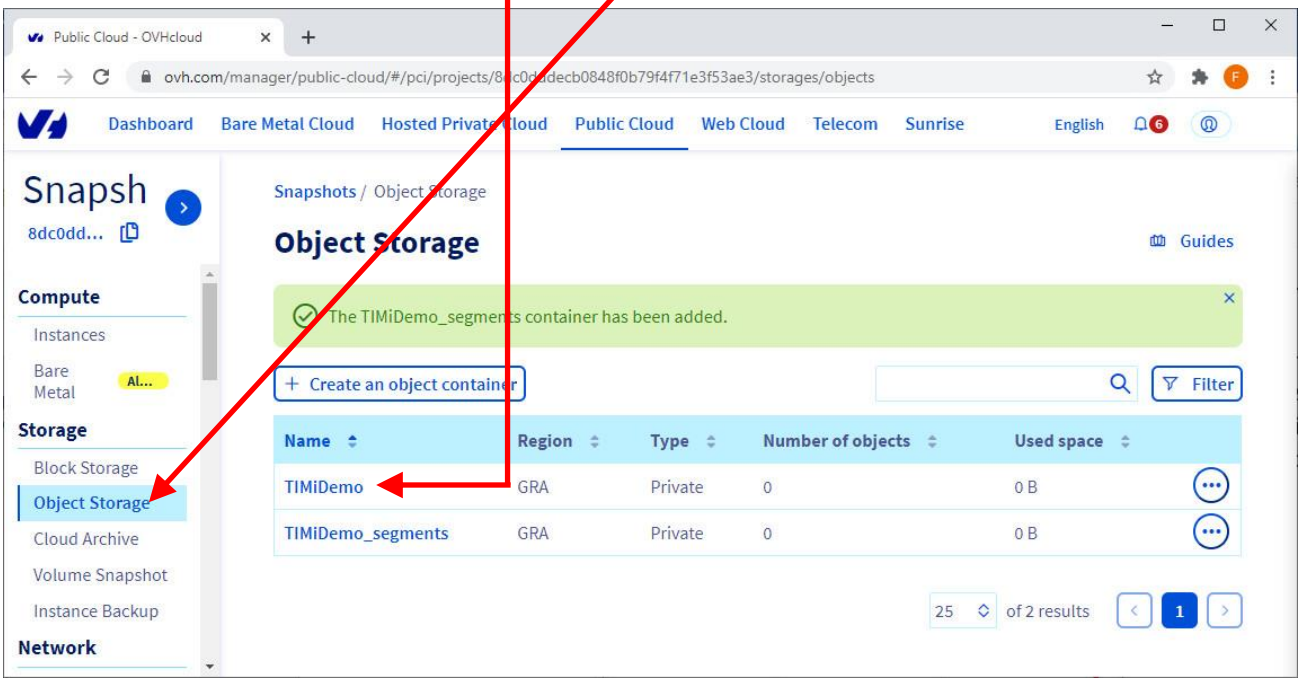
Please make sure that you selected a user that can access the Object Storage (i.e. the "roles" of this user must contain "ObjectStore Operator"). Alternatively, create a brand-new user with the correct "roles", following the procedure given in the point 2.2. here above.

- Open the downloaded file ("openrc.sh") inside notepad.
  The Anatella parameter **P6** (tenant name) is visible here:



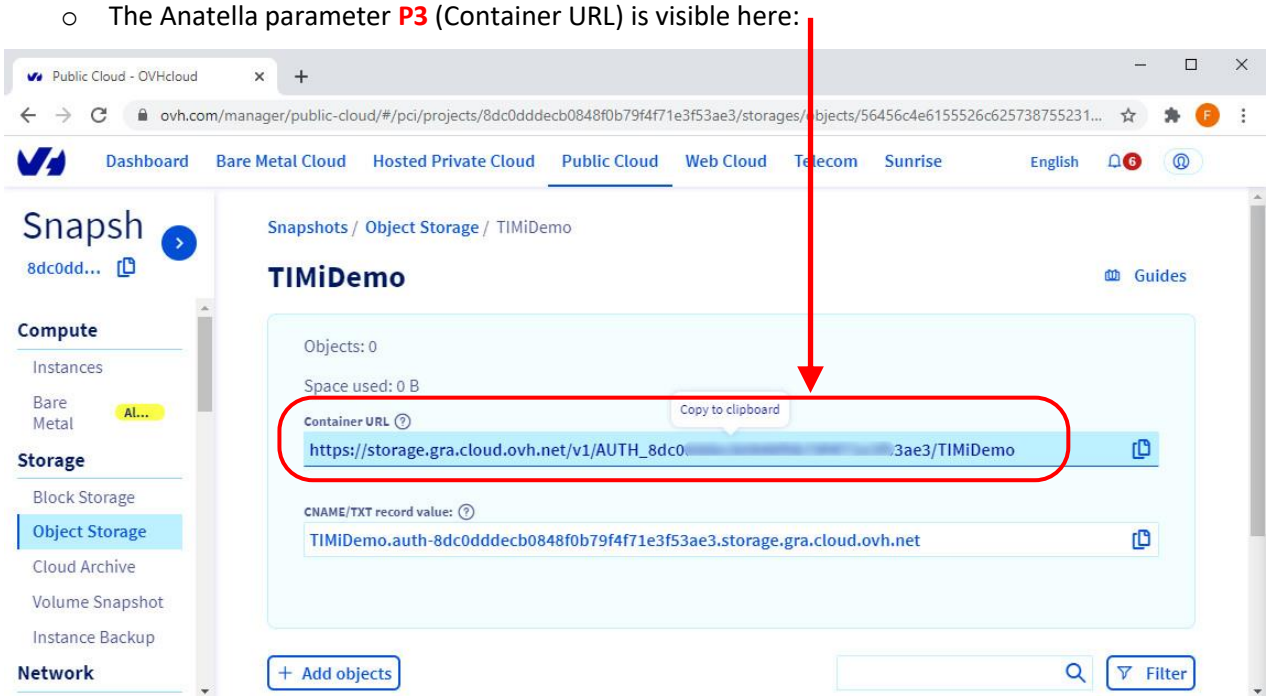
4.8. Get the Container URL:

- In the left menu, click on "Object Storage":
- Click on your (first) container:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- The Anatella parameter **P3** (Container URL) is visible here:



You now have the 4 required parameters to connect to your OVH Cloud Object Storage.

5.23.60. OVH Cloud Object Storage Download Files

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: ovhDownload | Add parameter | Remove

Description	Value	
Remote File to download		P1
Local filepath (to save locally)		P2
Chunk size (MB)	20	P3
Container URL		P4
user name		P5
password		P6
tenant name		P7
Debug Display?	No debug	P8
Optional parameters for cURL		P9
Number of "Connection Errors" before Ab...	10	P10
Error Management	Continue with "Error" Status	P11

Short description:
Download files from a container in an OVH Cloud Object Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the OVH Website (parameters **P4**, **P5**, **P6** and **P7**). Please see the section 5.23.59.2. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.59.2., you can use the parameters **P1** and **P2** to download the required files from your OVH Cloud Object storage system.

The parameter **P3** (i.e. the Chunk Size) is usefull when downloading very large files. The file is downloaded chunk-by-chunk. If one connection error happens during the download, only the last

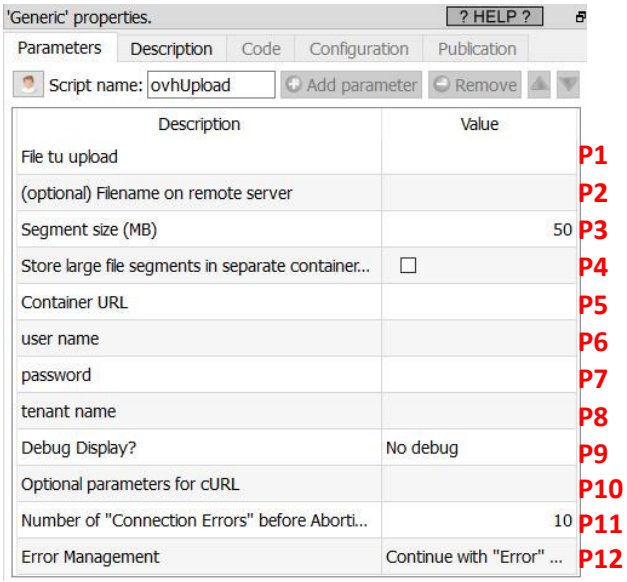
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

chunk needs to be re-downloaded. For more information about the parameter **P3**, see the section 5.23.59.1.

5.23.61. OVH Cloud Object Storage Upload Files

Icon:

Property window:



Short description:
Upload files to a container in an OVH Cloud Object Blob Storage

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P10** for web-access through a PROXY server.

To use this Action, you'll need to get several parameters from the OVH Website (parameters **P5**, **P6**, **P7** and **P8**). Please see the section 5.23.59.2. for more details on how to get these parameters.

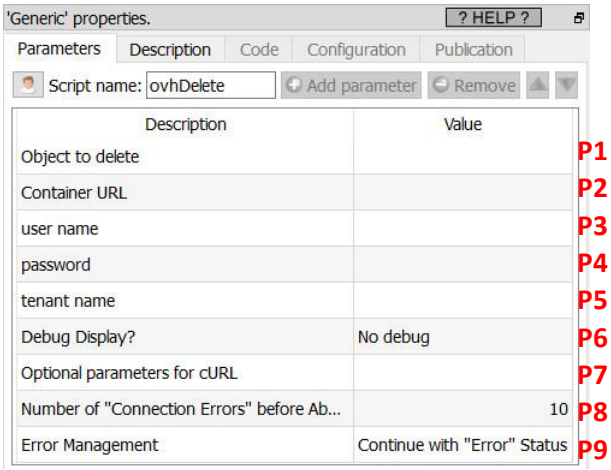
Once you have completed the "setup process" described in the section 5.23.59.2, you can use the parameters **P1** and **P2** to upload the required files to your OVH Cloud Object storage system.

The parameter **P3** (i.e. the segment Size) and **P4** are usefull when uploading very large files. The file is uploaded segment-by-segment. If one connection error happens during the upload only the last segment needs to be re-uploaded. For more information about these 2 paramaters, see the section 5.23.59.1.

5.23.62. OVH Cloud Object Storage Delete Files

Icon:

Property window:



Short description:
Delete files/Objects from a container in an OVH Cloud Object Storage

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

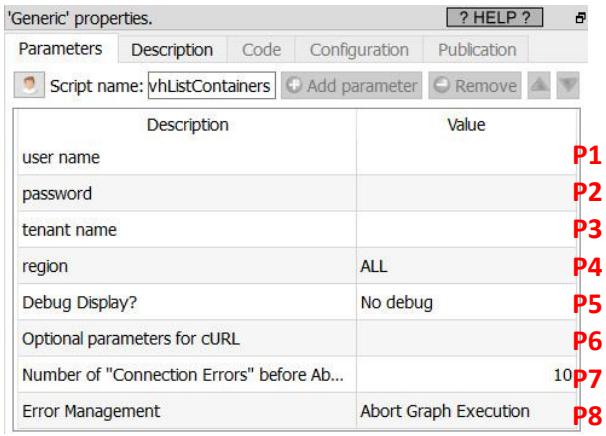
To use this Action, you'll need to get several parameters from the OVH Website (parameters **P2**, **P3**, **P4** and **P5**). Please see the section 5.23.59.2. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.59.2., you can use the parameter **P1** to delete the required files in your OVH Cloud Object storage system.

5.23.63. OVH Cloud Object Storage List Containers

Icon:

Property window:



Short description:
List all containers in a OVH Cloud Object Blob Storage system

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

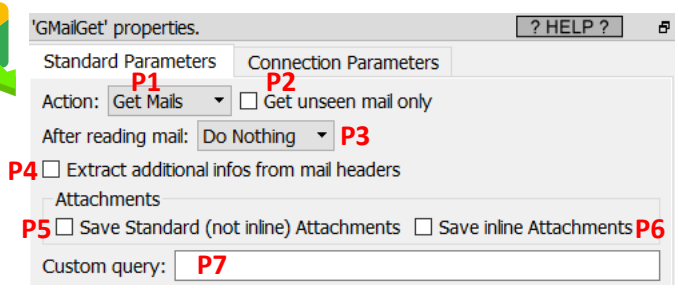
To use this Action, you'll need to get several parameters from the OVH Website (parameters **P1**, **P2**, and **P3**). Please see the section 5.23.59.2. for more details on how to get these parameters.

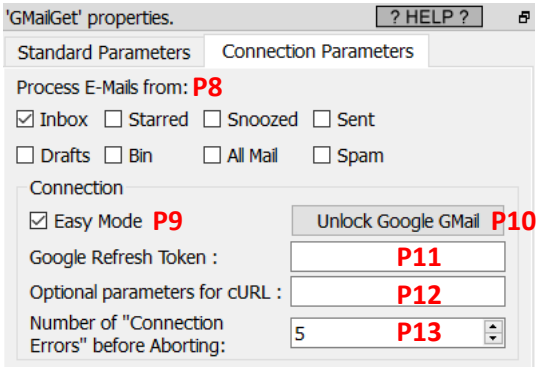
Once you have completed the "setup process" described in the section 5.23.59.2., you can use this action to list the available containers in your OVH Cloud Object storage system.

5.23.64. Get Emails from GMail

Icon:

Property window:





Short description:
Get Emails from Gmail

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P12** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

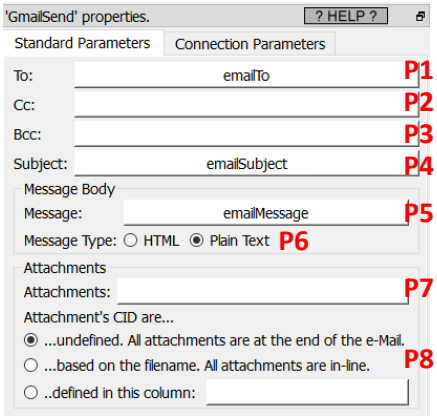
To use this Action, you'll first need to get the "Refresh Token" parameter from the Google Website (i.e. you need the parameter **P11**). To get this "Refresh Token", simply click on the button **P10**. Alternatively, you can also uncheck the "Easy Mode" (parameter **P9**) and follow the procedure from section 5.23.11 to get your 3 connection parameters (clientId, clientSecret, refreshToken).

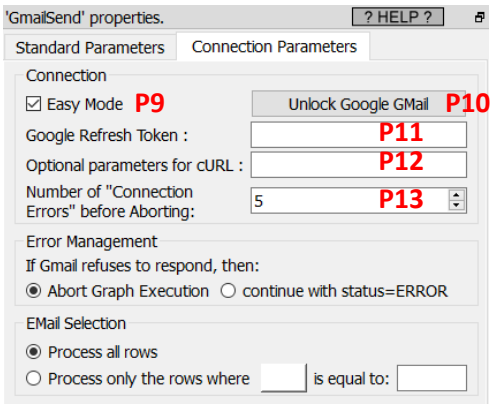
Once you have completed the "setup process", you can use the parameters **P1** to **P8** to download/list/purge the emails from your Gmail subscription.

5.23.65. Send Emails using GMail

Icon:

Property window:





Short description:
Send Emails using Gmail

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P12** for web-access through a PROXY server.

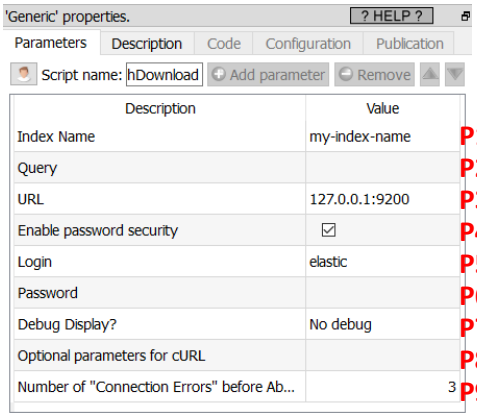
To use this Action, you'll first need to get the "Refresh Token" parameter from the Google Website (i.e. you need the parameter **P11**). To get this "Refresh Token", simply click on the button **P10**. Alternatively, you can also uncheck the "Easy Mode" (parameter **P9**) and follow the procedure from section 5.23.11 to get your 3 connection parameters (clientId, clientSecret, refreshToken).

Once you have completed the "setup process", you can use the parameters **P1** to **P8** to send your emails using your Gmail subscription.

5.23.66. Download JSON documents from Elastic Search

Icon:

Property window:



Short description:
Download JSON documents from Elastic Search

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameter P2 allows to define a filter to download some specific JSON documents from the index. The syntax of the filter is given here: https://lucene.apache.org/core/2_9_4/queryparsersyntax.html

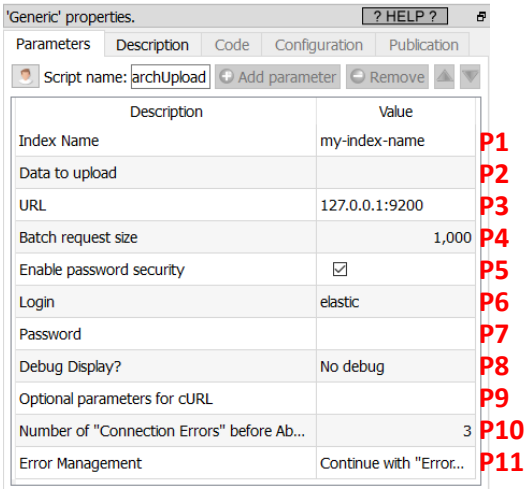
Currently, Anatella does not support the more advanced filters described here: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-filter-context.html
Drop us a line if you need this!

Typically, this Action is directly connected to the input pin of the "readJSON" Action (see section 5.2.11. for more details about this Action) to extract the required data from the returned JSON files.

5.23.67. Upload JSON documents to Elastic Search

Icon:

Property window:



Short description:
Upload JSON documents to Elastic Search

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

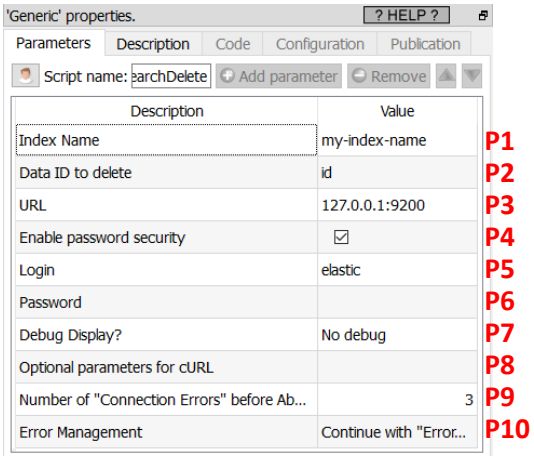
Depending on the value of the parameter **P2**, this action has 2 different operating modes:

- The parameter **P2** contains one column: The given column contains a complete JSON file to upload to ElasticSearch.
- The parameter **P2** contains several columns: For each input row, Anatella construct a small JSON dictionary based on the column's names and column's contents (the dictionary entries are the column's name and the dictionary values are the column's contents). This small dictionary is then sent to the ElasticSearch server.

5.23.68. Delete JSON documents from an Elastic Search Server

Icon:

Property window:



Short description:
Delete JSON documents from an Elastic Search Server.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

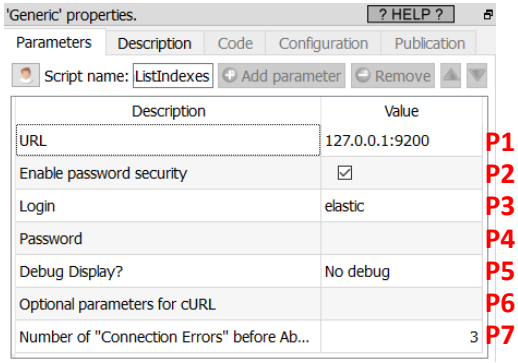
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

You can obtain the "ID" of the JSON document to delete (i.e. the parameter **P2**) using the ElasticSearchDownload Action (see section 5.23.66 for more details on this Action).

5.23.69. List all the Indexes on an Elastic Search Server

Icon:

Property window:



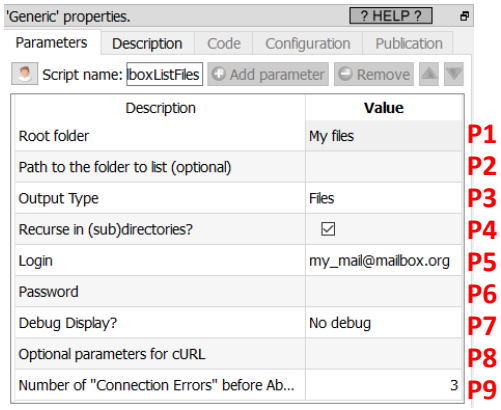
Short description:
List Indexes on an Elastic Search Server.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

5.23.70. List the Files on a Mailbox.org drive

Icon:

Property window:



Short description:
List files on a Mailbox.org drive.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

The "root folder" parameter **P1** must be chosen amongst the following choice :[My files,Other files,Public files,Shared files]. In addition to that choice, the parameter **P2** can be optionally provided to list files inside a particular folder. Here are some examples:

- I want to list all the files in "My files" :
  Parameter **P1** is: "My Files"
  Parameter **P2** is empty
- I want to list the files of the folder : "My files/myFolder1/myFolder2"
  Parameter **P1** is: "My Files"
  Parameter **P2** is: "/myFolder1/myFolder2/"

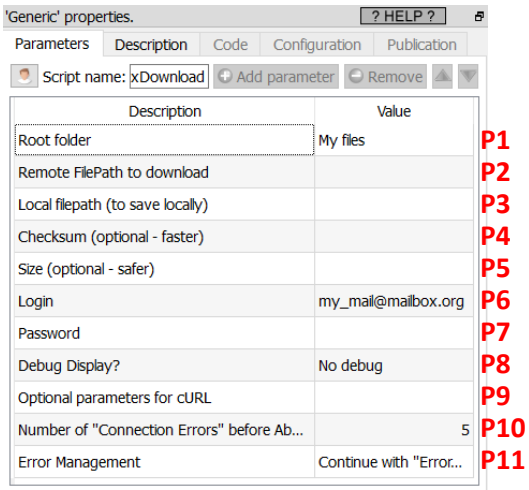
Leading and trailing slash (/) can be omitted for the parameter **P2**.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.71. Download Files from a Mailbox.org drive

Icon:

Property window:



Short description:
Download Files from a Mailbox.org drive.

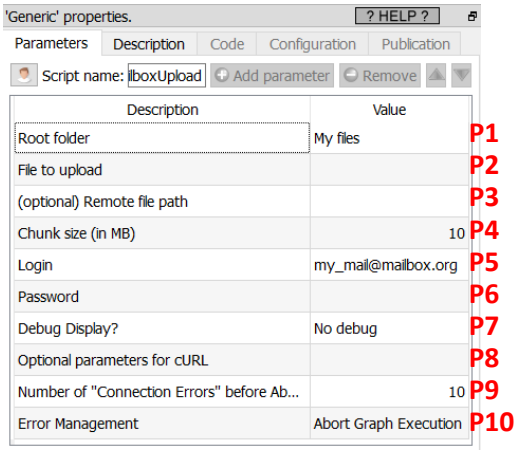
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

The parameters **P4** (checksum) and **P5** (size) can be optionnaly given. You can get these 2 parameters from the MailboxListFiles Action (see previous section 5.23.70 for more details about this Action). If the parameter **P4** is given, then the download will be significatively faster.

5.23.72. Upload Files to a Mailbox.org drive

Icon:

Property window:



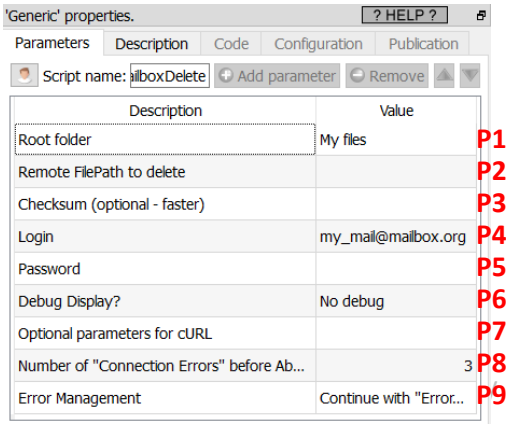
Short description:
Upload Files to a Mailbox.org drive.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

5.23.73. Delete Files on a Mailbox.org drive

Icon:

Property window:



Short description:
Dete Files on a Mailbox.org drive.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

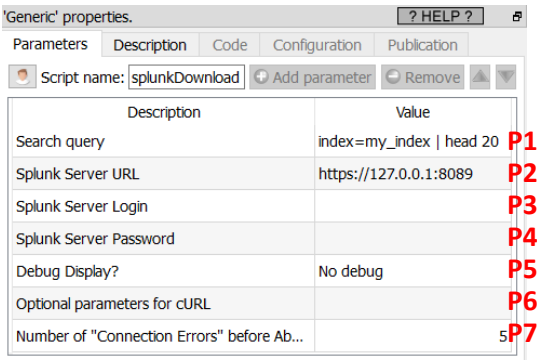
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

The parameters **P3** (checksum) can be optionnaly given. You can get this parameter from the MailboxListFiles Action (see section 5.23.70 for more details about this Action). If the parameter **P3** is given, then the delete procedure will be significatively faster.

5.23.74. Download logs from a Splunk Server

Icon:

Property window:



Short description:
Download logs from a Splunk server.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

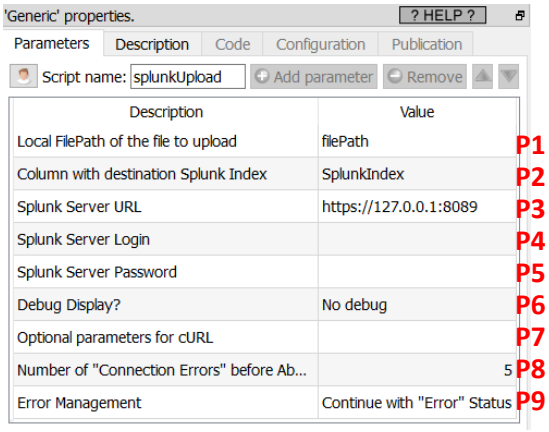
For this action to work you need to enable the “Splunk Management Port” inside the Splunk server settings. This is usually port 8089.

The parameter **P1** is the “Search Query”inside the Splunk server. The syntax of the “Search Query” is described here: https://docs.splunk.com/Documentation/SCS/current/SearchReference/Introduction

5.23.75. Upload logs to a Splunk Server

Icon:

Property window:



Short description:
Download logs from a Splunk server.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

For this action to work you need to enable the “Splunk Management Port” inside the Splunk server settings. This is usually port 8089.

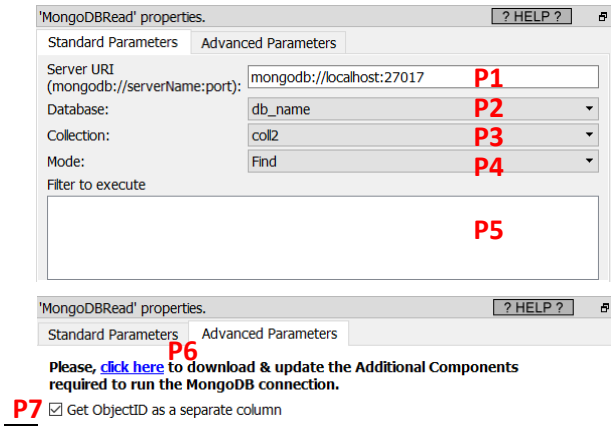
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.76. Download JSON from a MongoDB Server

Icon:

Property window:

Short description:
Download JSON from a MongoDB server.



Long Description:
Before using this Action, you need to install the Additional Components that runs the MongoDB connection: click on the link in the parameter **P6**.

The parameter **P1** is the "Server URI" to connect to your MongoDB server. If you are already using the "MongoDB Compass" tool, that's exactly the same "Server URI": You can directly copy/paste it into Anatella. The structure of this URI is:

```
mongodb://<username>:<password>@<servername>:<port>/<authorizationDB>?<options>
```

The server URI for an authenticated connection (with a login/password) typically looks like:

```
mongodb://myLogin:myPass@192.168.1.200:27017/test?authMechanism=SCRAM-SHA-1
```

You'll find more details on the syntax of the parameter **P1** here:
https://docs.mongodb.com/manual/reference/connection-string/

As soon as you have defined the parameter **P1**, Anatella automatically fills-in the combo box for the parameters **P2** and **P3**. To extract the whole collection inside a MongoDB server, just let the parameter **P5** empty and set the parameter **P4**="Find".

The value of the parameter **P5** can either be a "filter expression" or an "aggregate expression" (depending on the value of the parameter **P4**). Here is an example of "filter expression" (parameter **P5**) that returns all the JSON documents that contains an attribute named "myInt" whose value is lower than 25:

```
{"myInt":{"$lt":25}}
```

The syntax of the "filter expression" (parameter **P5**) is the syntax of the "filter" property described here:
https://docs.mongodb.com/manual/reference/command/find/

Here is an example of "aggregate expression" (parameter **P5**) that computes a "count" aggregate :

```
{ "pipeline":
  [
    {"$group":
      {
        "_id": null,
        "count": {"$sum":1}
      }
    }
  ]
}
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

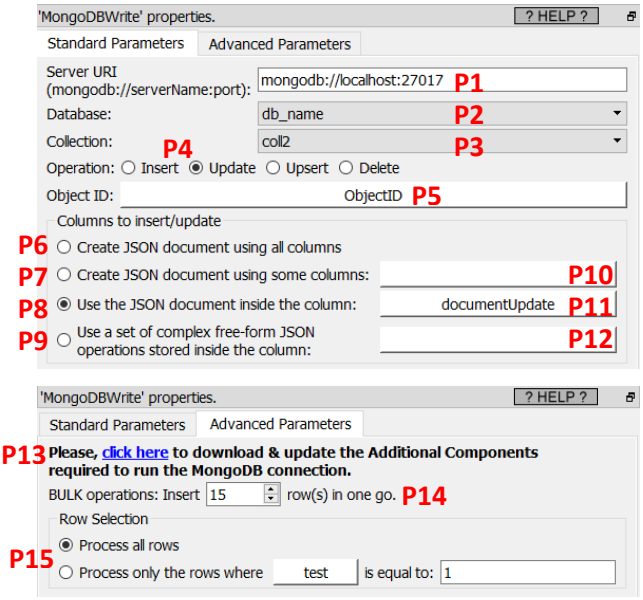
The syntax of the “aggregate expression” (parameter **P5**) is described here:
https://docs.mongodb.com/manual/reference/command/aggregate/

5.23.77. Upload JSON documents to a MongoDB Server

Icon:

Property window:

Short description:
Upload JSON documents to a MongoDB Server

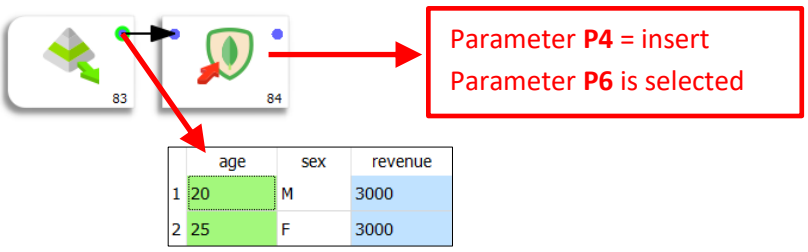


Long Description:
Before using this Action, you need to install the Additional Components that runs the MongoDB connection: click on the link in the parameter **P13**. As soon as you defined the parameter **P1**, Anatella automatically fills-in the combo box for the parameters **P2** and **P3**.

The option “upsert” parameter **P4** means that MongoDB will try to update a document with new values coming from Anatella and if the document does not exist yet, MondoDB will perform an “insert” instead of an “update”. The document to update/delete is identified using it’s “Object ID”. The “Object ID’s” of the document to update/delete can be specified using the parameter **P5**. Most of the time, the parameter **P5** is the column “Object ID” received from the MongoDBRead action (described in the previous section 5.23.76).

Depending on which parameter is selected amongst the parameters **P6** to **P9**, there are 4 different ways to generate the different JSON documents that are sent to MondoDB:

- Create a JSON document using all columns (Parameter **P6** is selected)
  Here is an example:

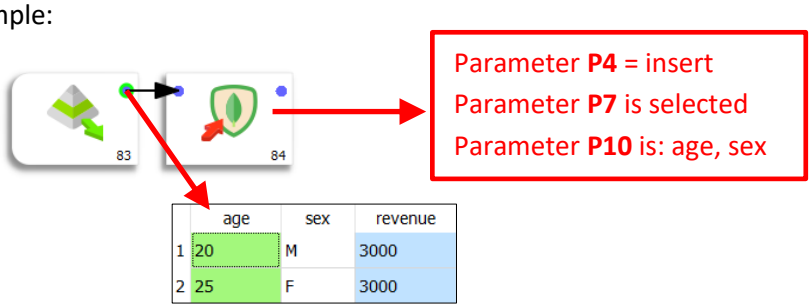


The above graph inserts inside the MongoDB database the following 2 JSON documents:

```
{ "age": 20, "sex": "M", "revenue": 3000 }
{ "age": 25, "sex": "F", "revenue": 3000 }
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

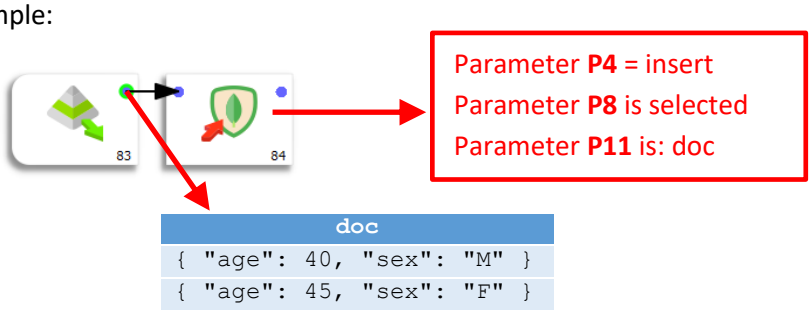
- Create a JSON document using some columns (Parameter **P7** is selected)
  Here is an example:



The above graph inserts inside the MongoDB database the following 2 JSON documents:

```
{ "age": 20, "sex": "M" }
{ "age": 25, "sex": "F" }
```

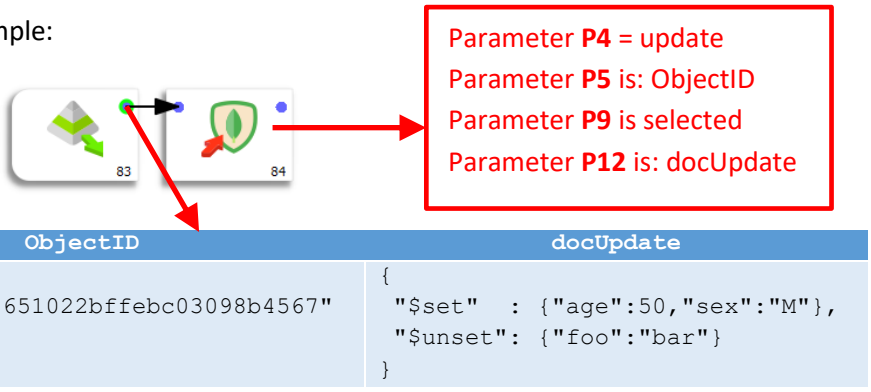
- Use the JSON document inside the columns (Parameter **P8** is selected)
  Here is an example:



The above graph inserts inside the MongoDB database the following 2 JSON documents:

```
{ "age": 40, "sex": "M" }
{ "age": 45, "sex": "F" }
```

- Use a set of complex free-form JSON operations stored inside the column (Parameter **P9** is selected)
  Here is an example:



The above Anatella graph updates the document with the oid=`54651022bffebc03098b4567` inside the MongoDB database. More precisely, this document is updated so that:

  - It contains the fields “age” and “sex”.
    Furthermore, these 2 fields values are set, respectively, to 50 and “M”.

```
{ "age": 50, "sex": "M" }
```

  - It does not contain the field “foo” anymore.

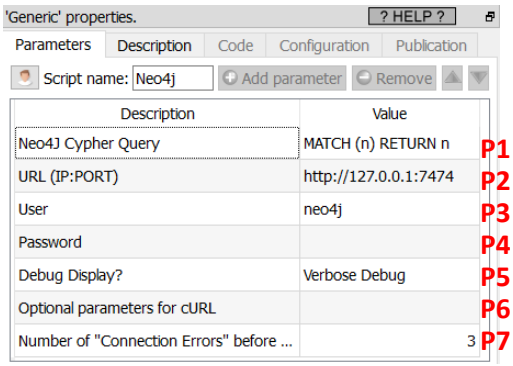
The documentation that explains the syntax of the JSON string that contains the different operations to perfom is here:
https://docs.mongodb.com/manual/reference/operator/update-field/

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.78. Download data from a Neo4J server

Icon:

Property window:



Short description:
Download data from a Neo4J server

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

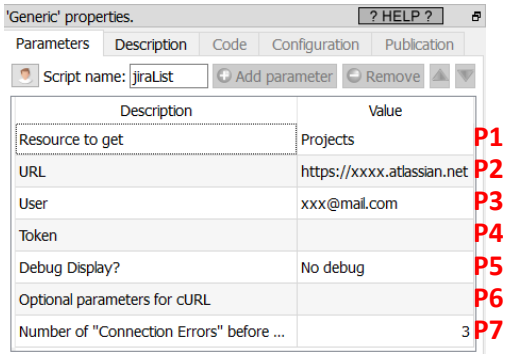
The connection to a Neo4J server is very simple & straightforward: just use your normal Neo4J login & password and you are “good to go”!

A word of caution: By default, the connection to a Neo4J server is unsecured (i.e. it’s using the “http” protocol instead of the “http**s**” protocol). This means that, by default, your login&password are directly visible to anybody looking at the TCP/IP packets passing through your network. Our advice is to configure your Neo4J server to use the more secure “http**s**” protocol. Anatella supports both protocols (“http” or “http**s**”).

5.23.79. List ressources on a Jira server

Icon:

Property window:



Short description:
List ressouces available in a Jira server

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

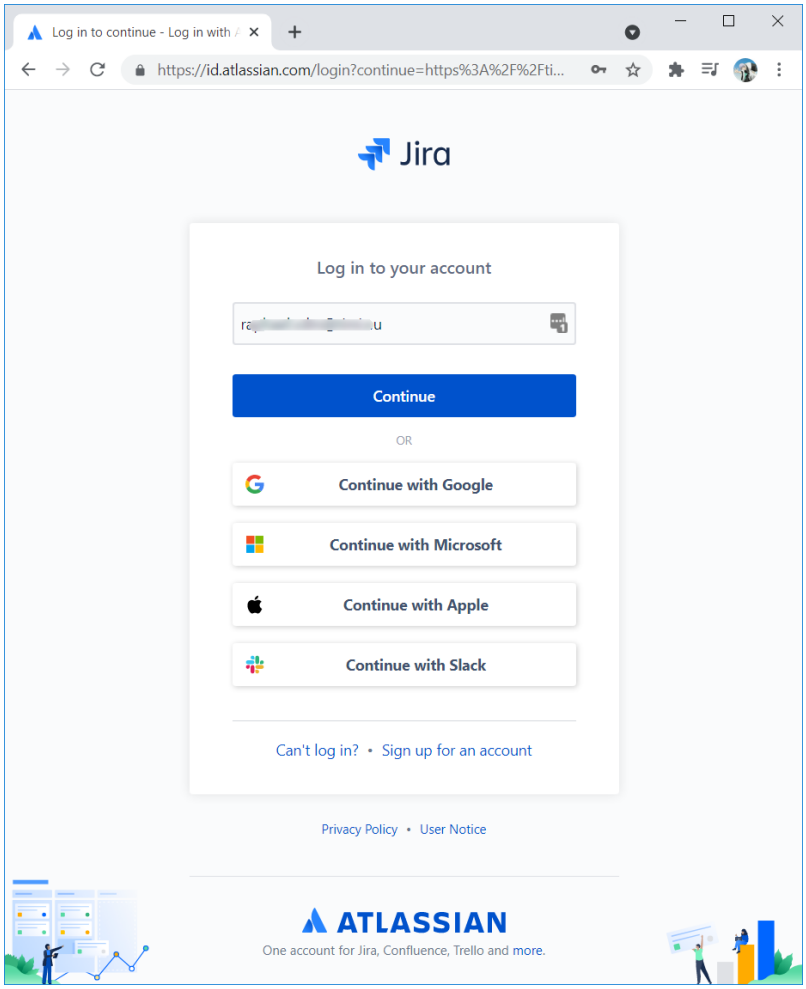
The parameter **P1** defines the resource downloaded from Jira: it can be: “Projects”, “Workflows” or “Dashboards”. All the parameters are self-explanatory with the exception of the parameter **P4**: it’s the “Jira token”: refer to the next section to know how to get your “Jira Token”.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

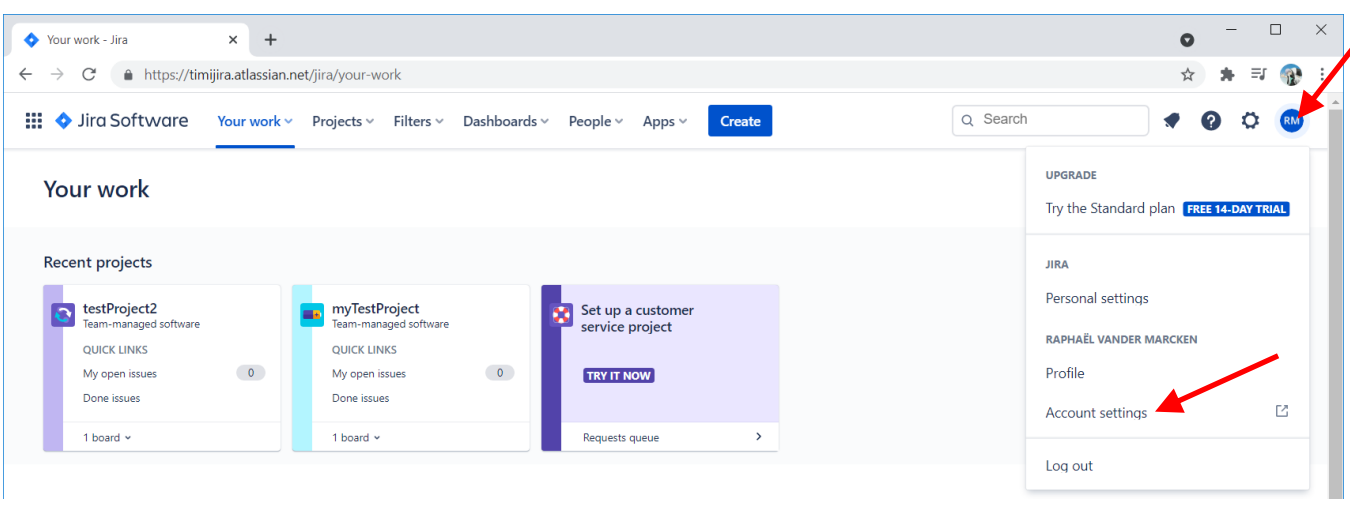
5.23.79.1. How to get your Jira Token?

Follow these steps to get your Jira Token:

1. Open the login page of your Jira server and log-in into the system. Typically, the login page of a Jira server looks like this:

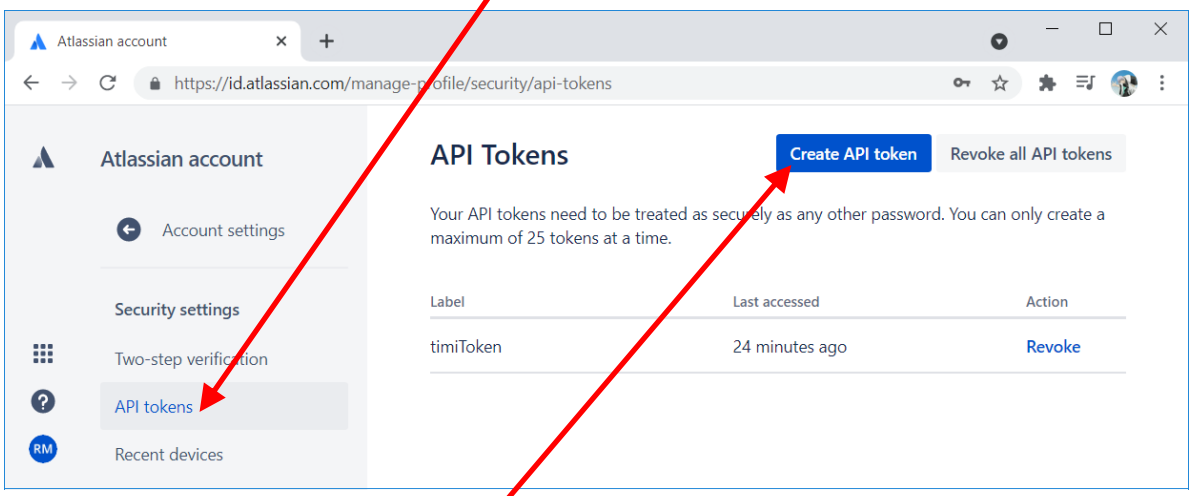


2. Click on the little blue circle with your initials in the top-right corner of the Browser and select inside the drop-down menu the “Account settings” option:

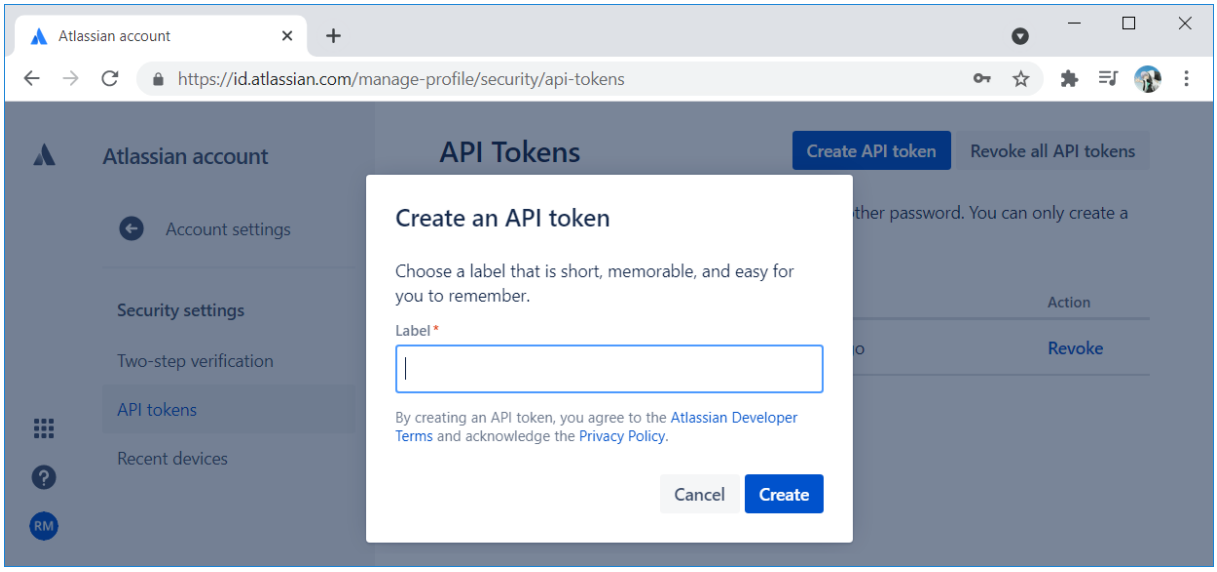


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

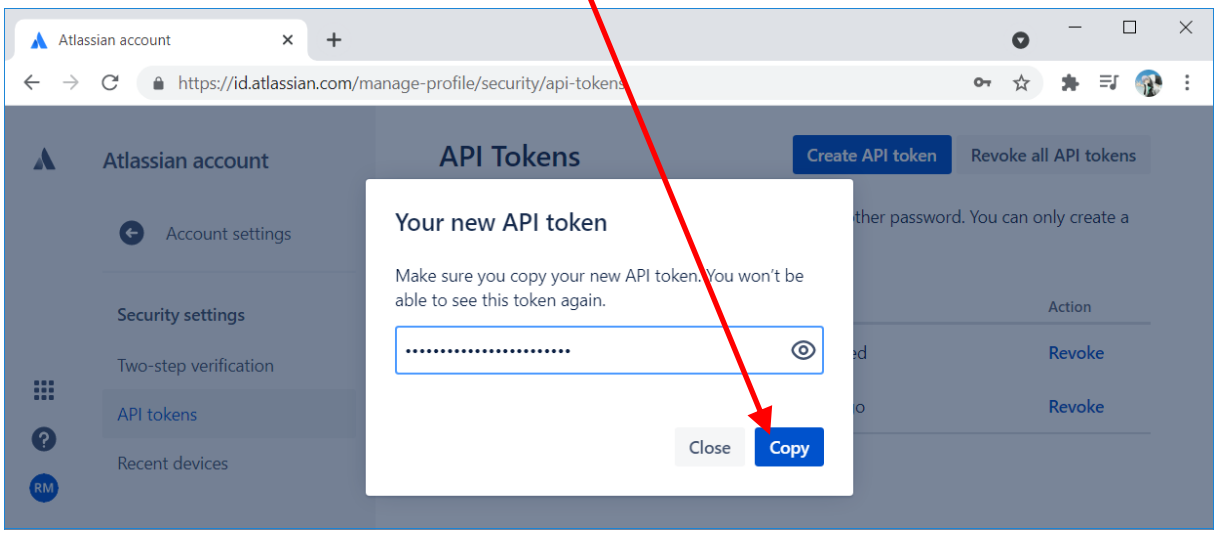
3. In the left column, click on “API Tokens”:



4. Click on the “Create API Token” button:
5. Enter any (meaningful) label for your token and click the “Create” button:



6. You get your new Jira API token: Click the “Copy” button and paste your token into Anatella.



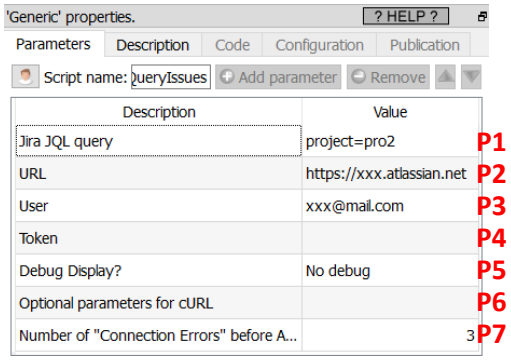
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Warning: There is no way of retrieving again the token from the Jira server after this step. This means that you need to store it in a safe place if you intend to use it again later on.

5.23.80. Retrieve the results of a JQL query running on a Jira server

Icon:

Property window:



Short description:
Download issue-related data from a Jira server using a JQL query.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

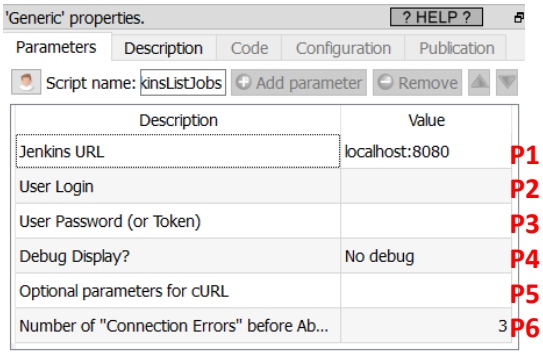
The parameter **P1** is the Jira JQL query executed on your Jira Server. For more information about the JQL syntax, please refer to this webpage:
https://www.atlassian.com/software/jira/guides/expand-jira/jql

All the parameters of this Action are self-explanatory with the exception of the parameter **P4**: it's the "Jira token": refer to the previous section 5.23.79.1. to known how to get your "Jira Token".

5.23.81. List jobs in Jenkins

Icon:

Property window:



Short description:
List all the planned jobs in a Jenkins server

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

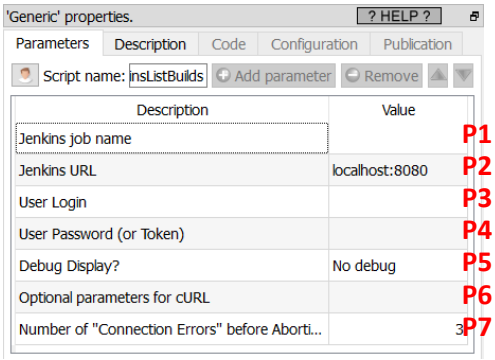
The other parameters are self-explanatory.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.82. Download logs from Jenkins

Icon:

Property window:



Short description:
Download the execution logs from a Jenkins server for a specific job

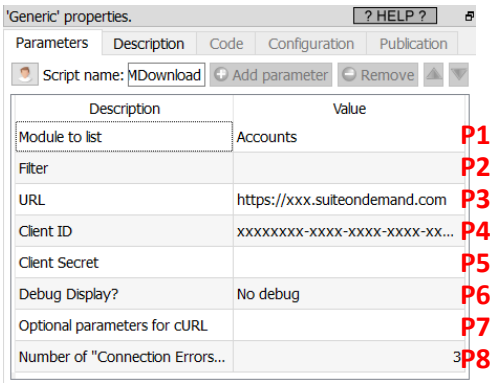
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

The other parameters are self-explanatory.

5.23.83. Download data from SuiteCRM

Icon:

Property window:



Short description:
Download data from SuiteCRM

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P7** for web-access through a PROXY server.

To use this Action, you'll need to get your credentials (i.e. the parameter **P4** and **P5**) from your SuiteCRM Website. Please see the next section 5.23.83.1. for more details on how to get these two parameters.

5.23.83.1. SuiteCRM First time setup

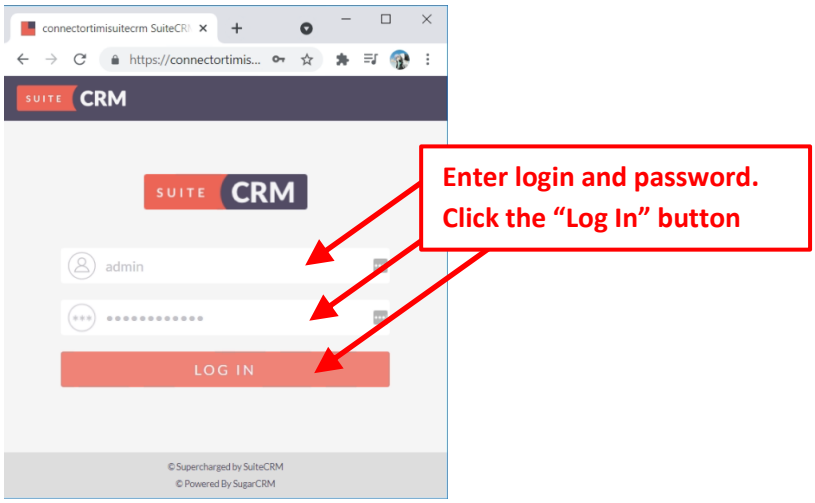
Before using the SuiteCRM Actions inside Anatella, you need to get these 2 parameters:

- the Client ID (Parameter **P4**)
- the Client Secret (Parameter **P5**)

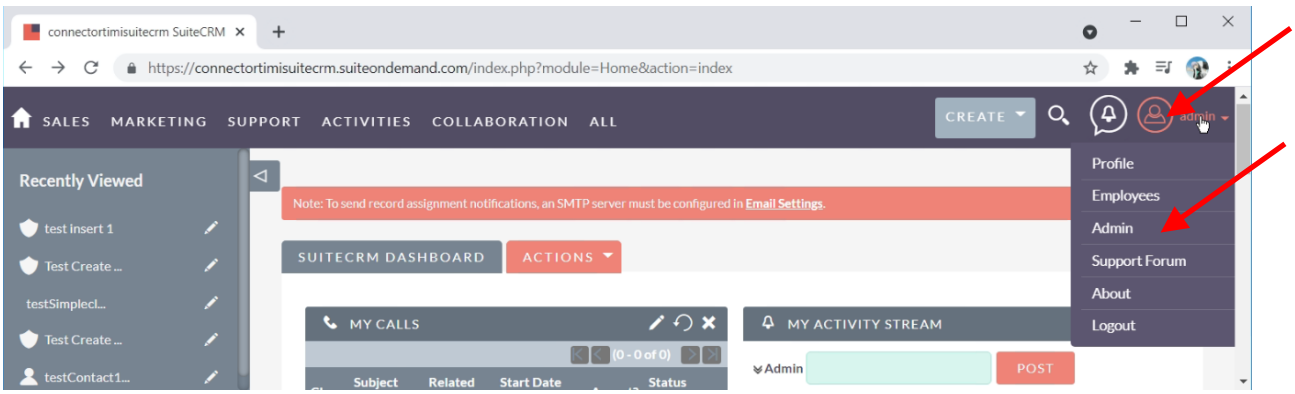
…from the SuiteCRM website. Here are the steps to get these 2 parameters:

1. Open your SuiteCRM website in a browser and "log-in" using your normal "Login" and "Password":

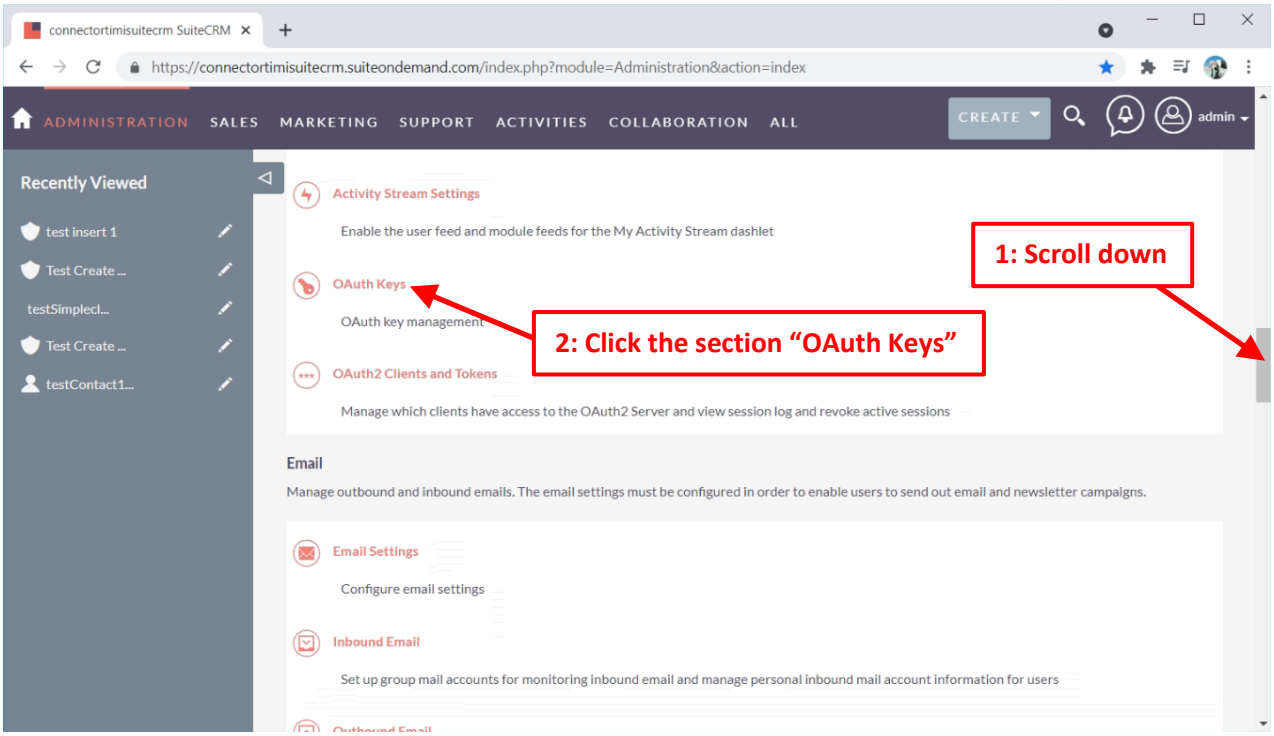
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



2. Open the Menu in the Top-Right corner, and select the "Admin" option:

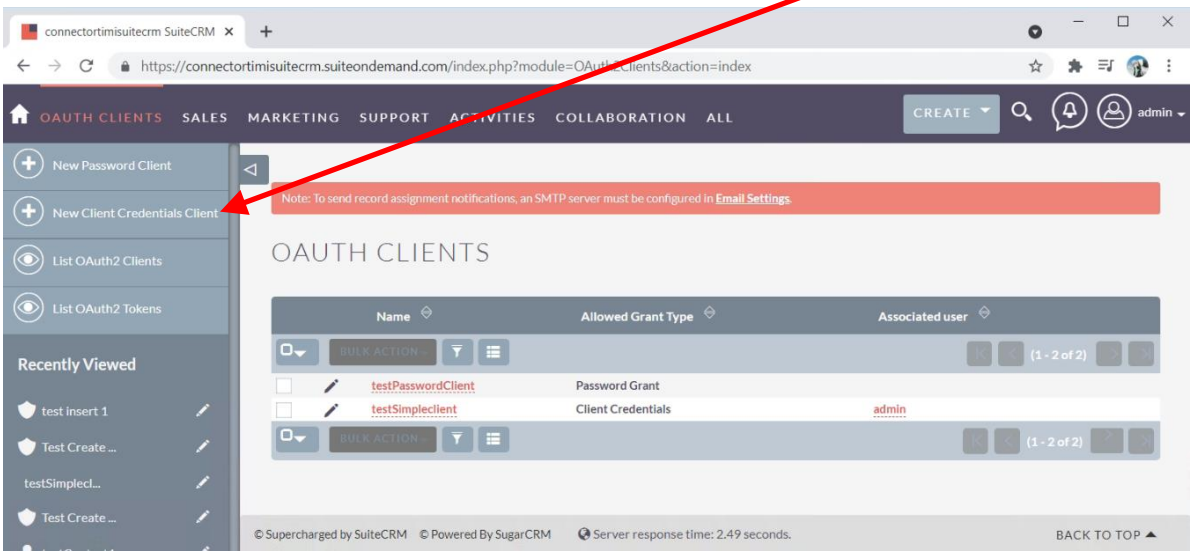


3. Inside the "admin" page, scroll-down to the section named "OAuth Keys" and click on it:

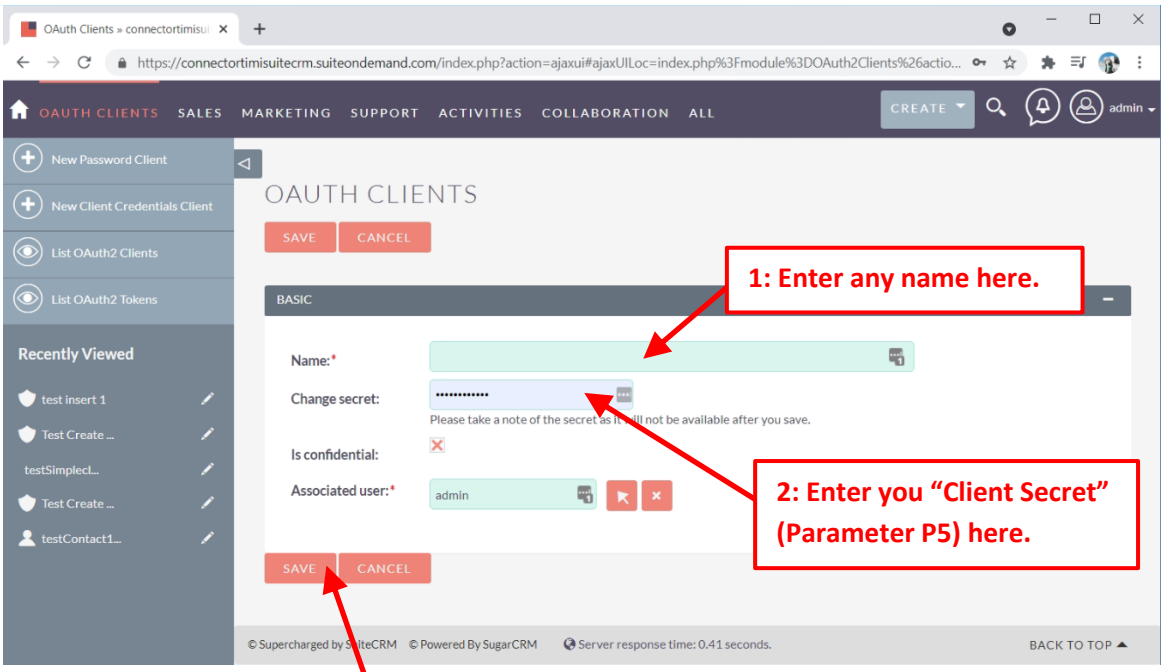


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

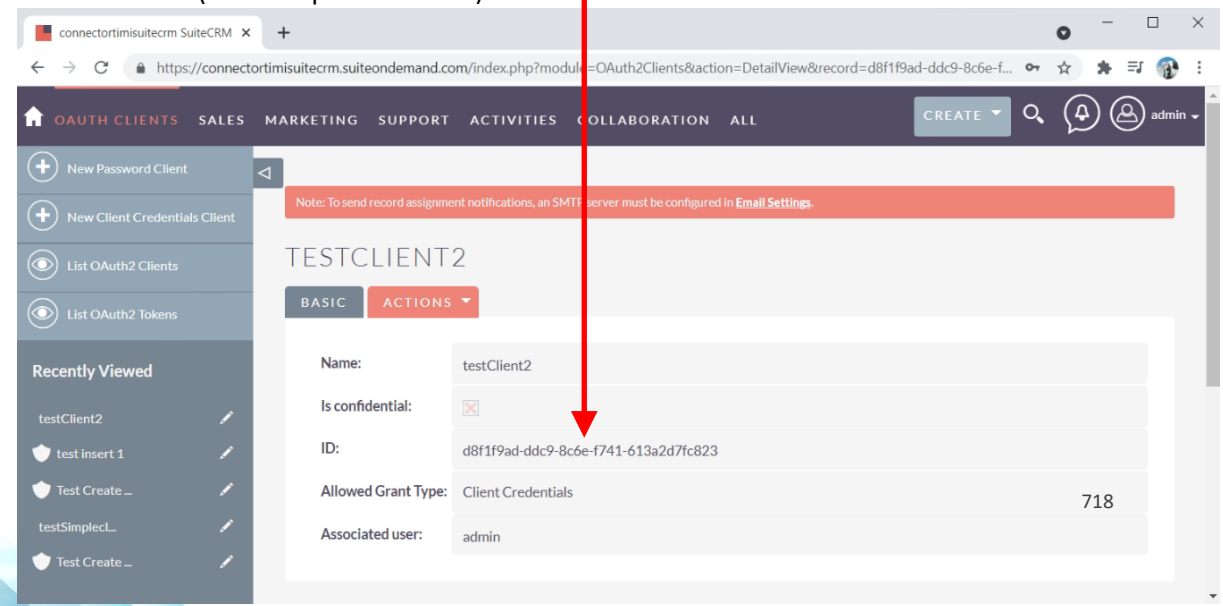
4. In the left panel, click on the “New Client Credentials Client”:



5. Inside the next form, you must enter a name (you can choose any name) and a “secret”. The “secret” is the parameter “Client Secret” (parameter **P5** from Anatella):

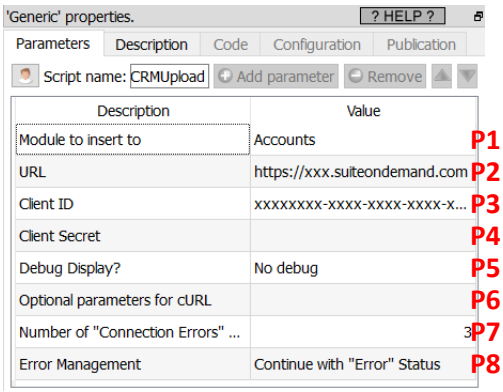


6. Click the SAVE button:
7. Your “Client ID” (Anatella parameter **P4**) is here:



5.23.84. Upload data to SuiteCRM

Icon:



Property window:

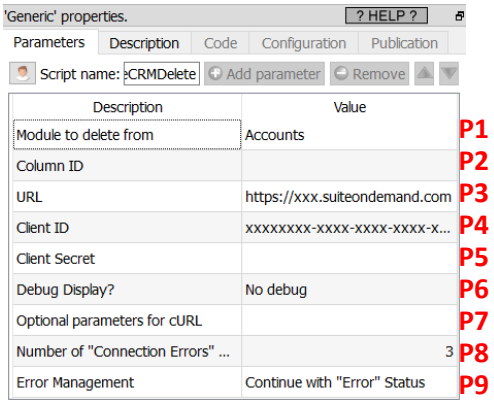
Short description:
Upload data to SuiteCRM

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

To use this Action, you'll need to get your credentials (i.e. the parameter **P3** and **P4**) from your SuiteCRM Website. Please see the previous section 5.23.83.1. for more details on how to get these two parameters.

5.23.85. Delete data on a SuiteCRM server

Icon:



Property window:

Short description:
Delete data on a SuiteCRM server.

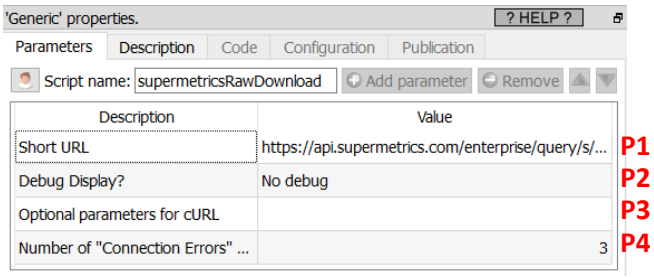
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

To use this Action, you'll need to get your credentials (i.e. the parameter **P3** and **P4**) from your SuiteCRM Website. Please see the previous section 5.23.83.1. for more details on how to get these two parameters.

5.23.86. Get Data from Supermetrics

Icon:

Property window:



Short description:
Get Data from Supermetrics.

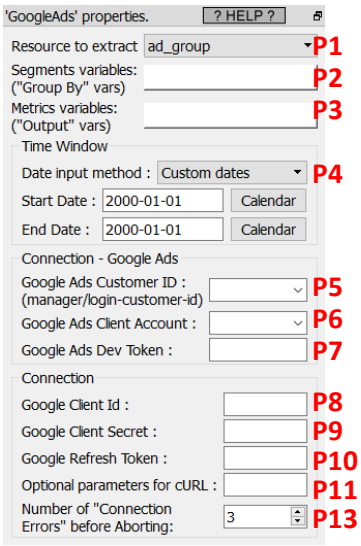
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P3** for web-access through a PROXY server.

5.23.87. Google Ads

Icon:

Property window:



Short description:
Get Data from Google Ads.

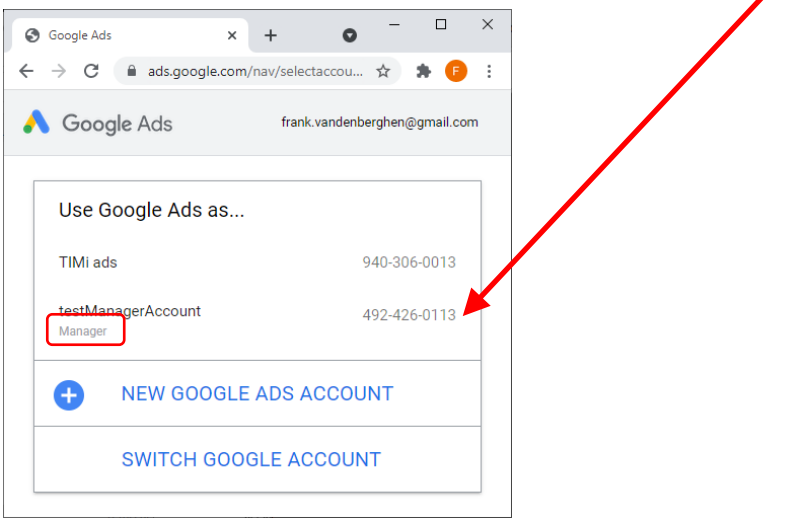
Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P11** for web-access through a PROXY server.

To be able to use this Action, you need to get these 3 parameters from Google: (1) Parameter **P8**: your "Client ID", (2) Parameter **P9**: your "Client Secret", (3) Parameter **P10**: your "Refresh Token". To get these 3 parameters, you must use the "Unlock Google Services" action detailed in section 5.23.11.

Here is the procedure to get the connection parameters **P5**, **P6**, **P7**:

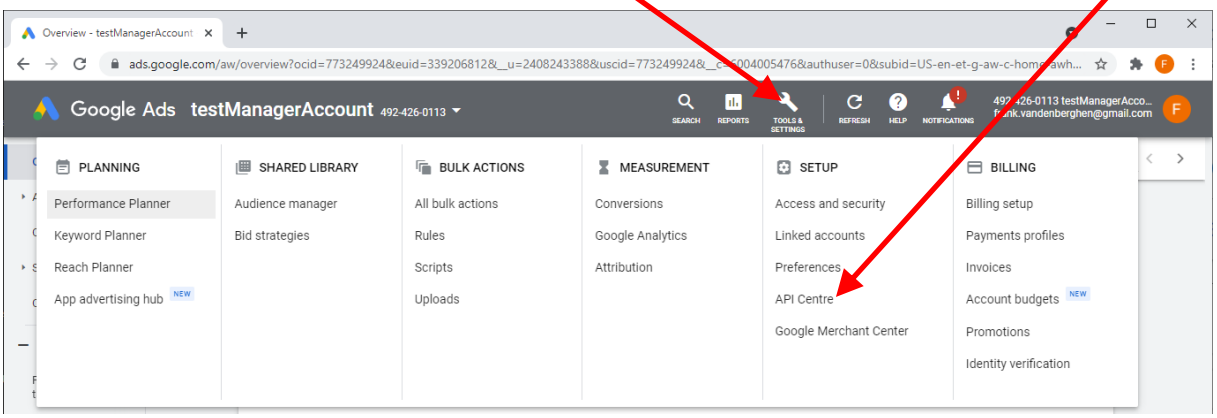
1. Open the URL https://ads.google.com/ and click on the "Sign in" button:

2. Sign-in as a **Manager** account: The parameter **P5** (manager login id) is here:

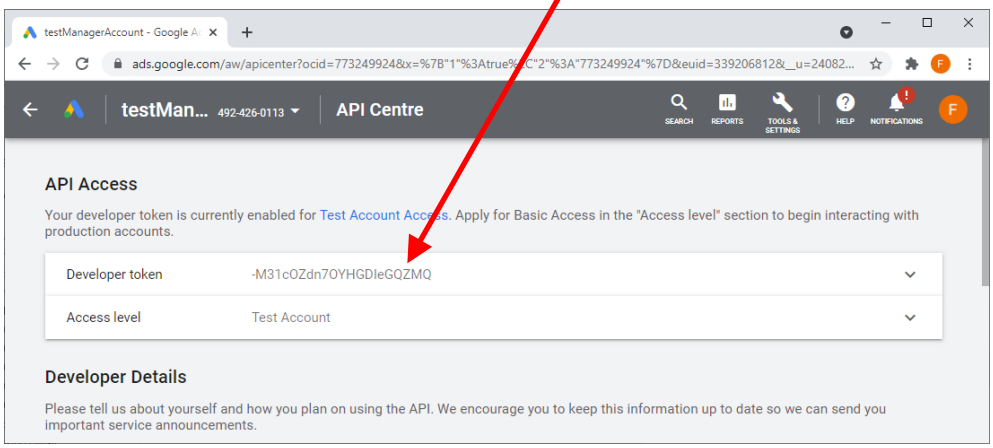


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

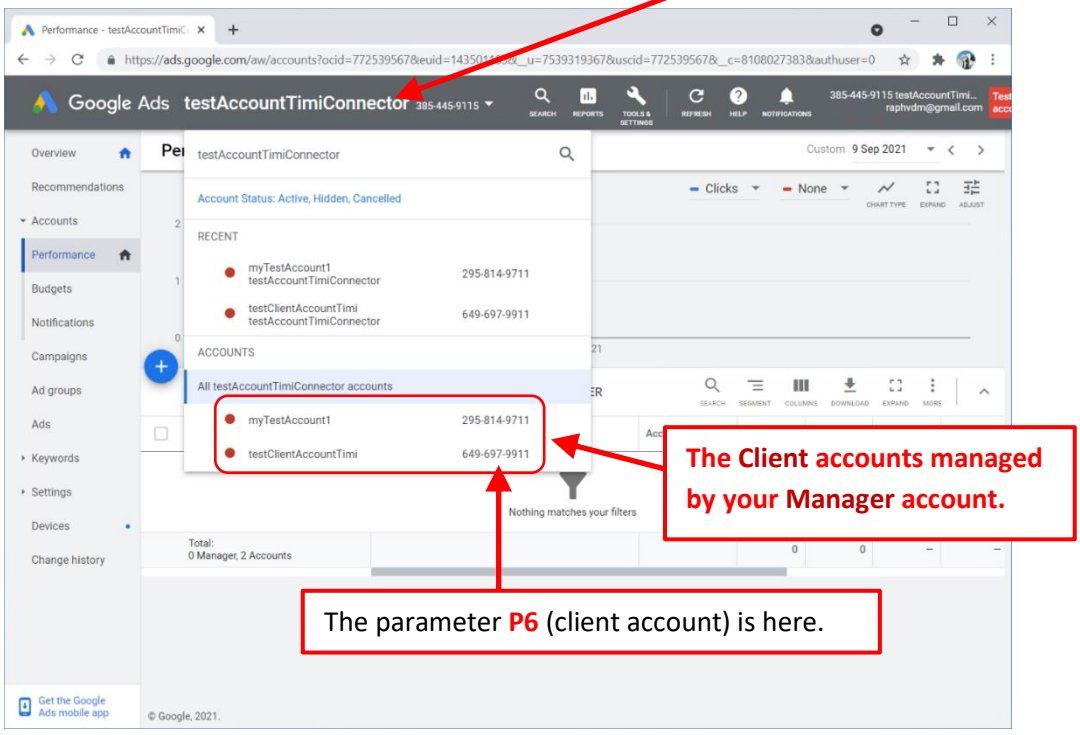
3. In the header, click on the “Tools&Settings” button: and then click on the “API center” menu:



4. The parameter **P7** (Developper token) is here:



With this parameter **P7** (Developper token) you will be able to access the data from all the “Google Ads **Client** Accounts” that are supervised by your “Google Ads **Manager** Account”. To see all the “Google Ads **Client** Accounts” that are managed by your “Google Ads **Manager** Account”, you can click on your name here, in the header:

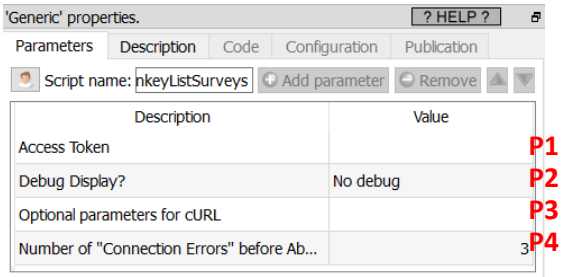


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.88. List Surveys from Survey Monkey

Icon:

Property window:



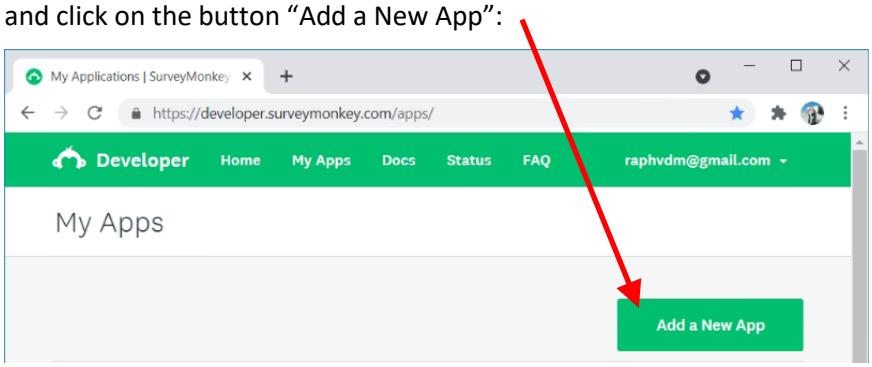
Short description:
List Surveys from Survey Monkey.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P3** for web-access through a PROXY server.

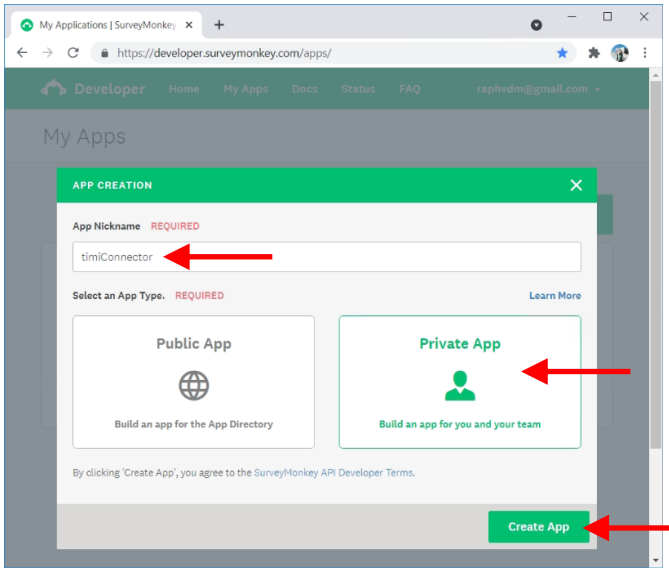
5.23.88.1. Survey Monkey First Time setup.

Before using SurveyMonkey (and to get your access token, the parameter **P1**), you need to follow the following procedure:

1. Log-in to Survey Monkey, open the URL : https://developper.surveymonkey.com/apps/ and click on the button “Add a New App”:

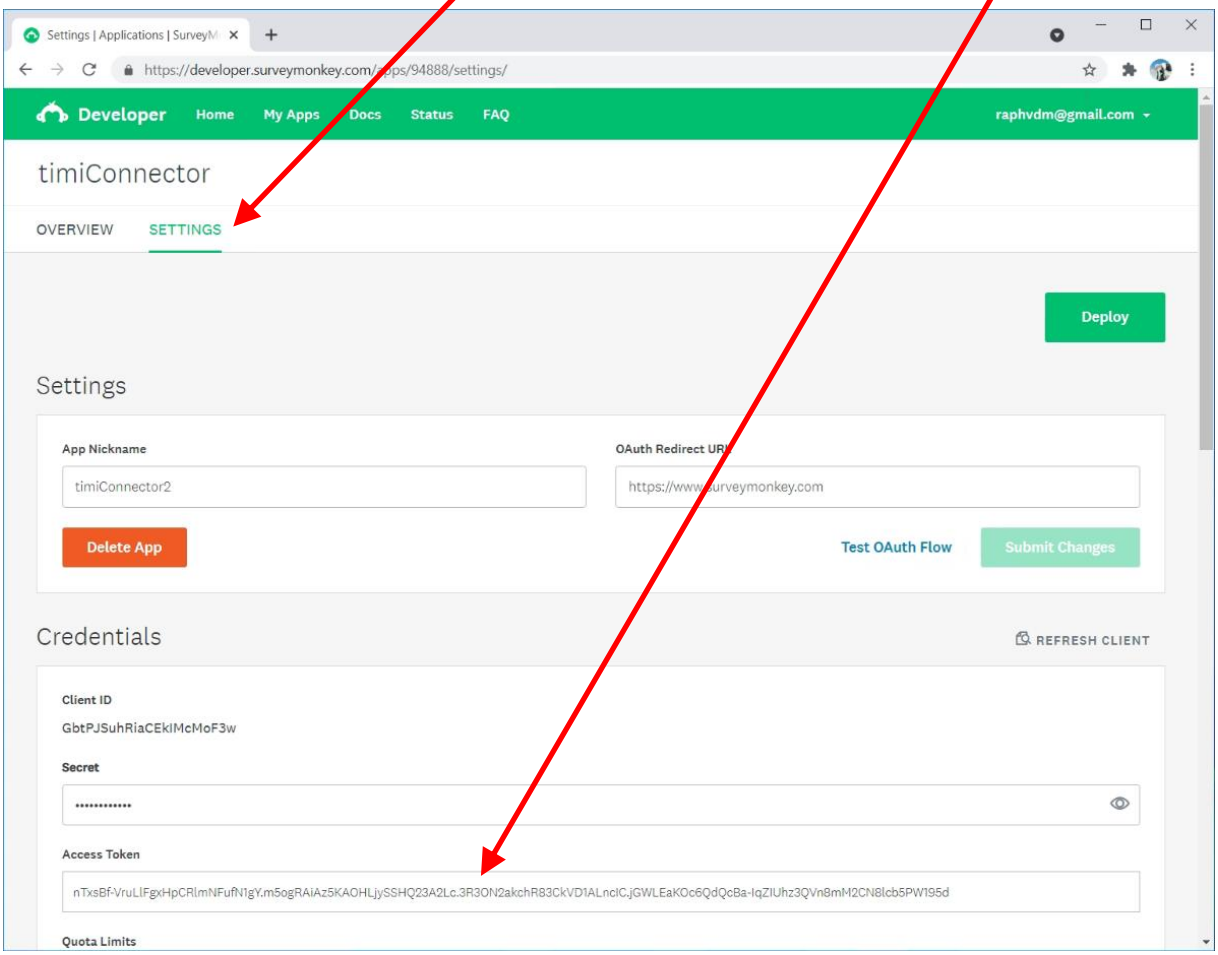


2. Give a name to your app (e.g. “timiConnector”), select “Private App” and click on the “Create App” button:

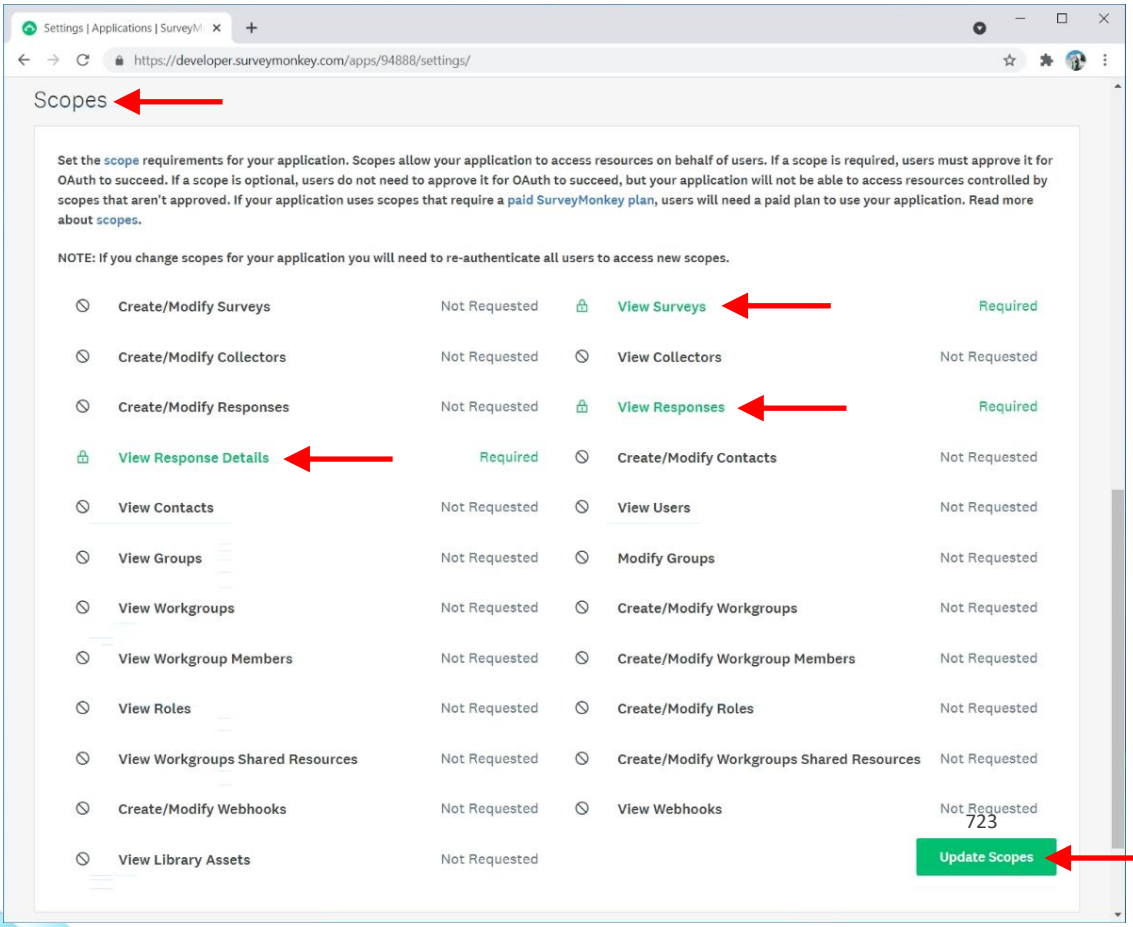


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Click on the “SETTINGS” menu: . The parameter **P1** (access token) is here:

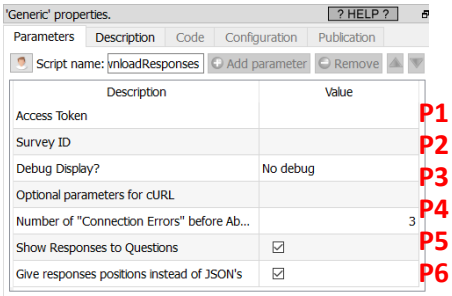


4. On the same webpage, scroll down to the section named “Scopes”, select these 3 scopes: “View Surveys”, “View Responses”, “View Response Details” and click on the “Update Scopes” button:



5.23.89. Download Responses from Survey Monkey

Icon:



Property window:

Short description:
Download Responses from Survey Monkey.

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P4** for web-access through a PROXY server.

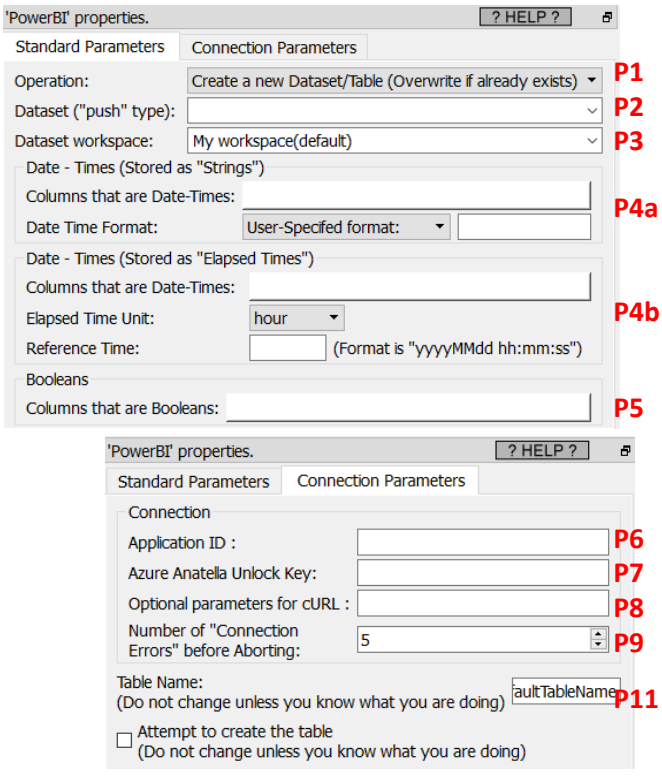
Before downloading reponse data from Survey Monkey, you need to follow the procedure given in the previous section (section 5.23.88.1.). The same section 5.23.88.1. explains how to get the parameter **P1** (access token). To get the parameter **P2** (Survey ID), you must execute the SurveyMonkeyListSurveys Action. The parameter **P2** is inside the column "id" in the output of the SurveyMonkeyListSurveys Action:

	title	nickname	language	folder_id	category	question_count	page_count	response_count	date_created	date_modifie	id
1	fullSurvey		fr	0		6	2	2	2021-10-01...	2021-10-01...	312468658
2	mon sondag...		fr	0		5	2	2	2021-09-24...	2021-09-24...	312157980
3	mon sondage		fr	0		2	1	2	2021-09-24...	2021-09-24...	312152088
4	Modèle d'éval...		fr	0	customer_...	8	5	0	2019-08-06...	2019-08-06...	185207330

5.23.90. PowerBI Upload

Icon:

Property window:



Short description:
Upload data To Microsoft Power BI

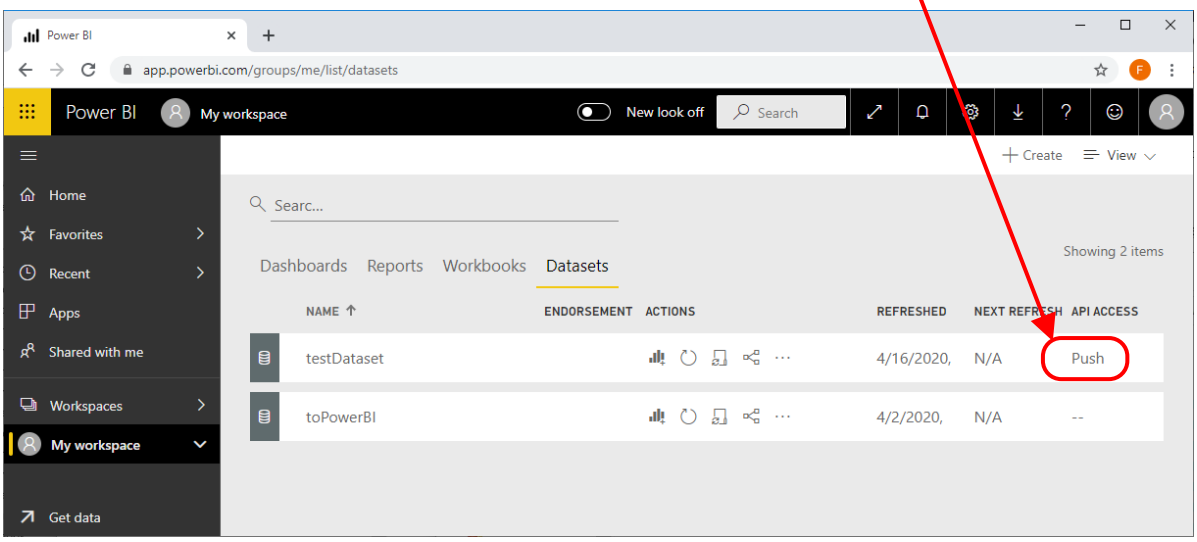
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P8** for web-access through a PROXY server.

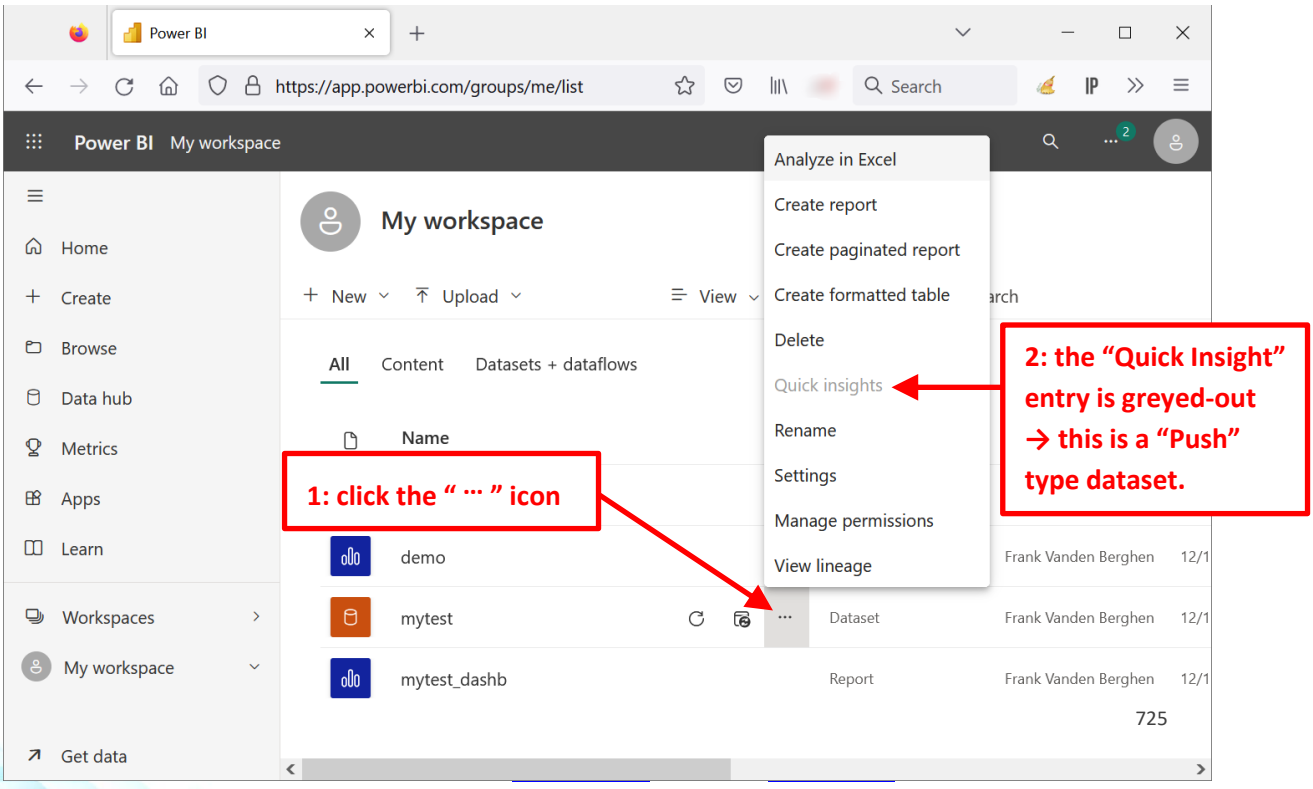
To use this Action, you'll need to get several parameters from the Azure Website (i.e. you need your "Application ID" -parameter **P6**- and your "Anatella Unlock Key" -parameter **P7**). Please see the section 5.23.27. for more details on how to get these parameters. Please don't forget the step 16 of this setup procedure because it's the most important one for PowerBI.

Once you have completed the "setup process" described in the next section 5.23.27, you can use the parameters **P1** to **P5** to upload data to your Microsoft Power BI system.

The PowerBI Action creates and updates datasets of the **"Push" type** on the PowerBI website (i.e. datasets with the property "addRowsAPIEnabled" set to TRUE). Back in 2020, you could easily see directly inside the powerBI web interface if a dataset was of the "Push" Type here:



In 2022, the PowerBI interface has changed and it's more difficult to see if a dataset is of the "Push" type or not but I *might* have found a trick: Click on the "…" icon next to the dataset name and a context-menu appears. Inside this context-menu, the "Quick Insights" menu entry is greyed-out for the "Push" type datasets: See illustration:



How to create a new "Push" Type dataset inside PowerBI? Open the PowerBI action inside Anatella. Enter the name of your new "Push" type dataset inside the parameter **P2** (Anatella will complain that it does not find this particular dataset: Just ignore this warning message), select "Create a new Dataset" for parameter **P1** and run the PowerBI Action. Refresh your browser. You new dataset should now be visible inside PowerBI.

On December 2022, inside Microsoft Power BI, the "Push" datasets have very strong limitations:

- 75 max columns
- 75 max tables (1 table per dataset; The table's name in each dataset is defined in parameter **P11**)
- 1,000,000 rows added per hour per dataset
- 200,000 max rows stored per table in FIFO dataset
- Other, less important, limitations:
  - 10,000 max rows per single POST rows request
  - 5 max pending POST rows requests per dataset
  - 120 POST rows requests per minute per dataset
  - If table has 250,000 or more rows, 120 POST rows requests per hour per dataset
  - 5,000,000 max rows stored per table in 'none retention policy' dataset
  - 4,000 characters per value for string column in POST rows operation

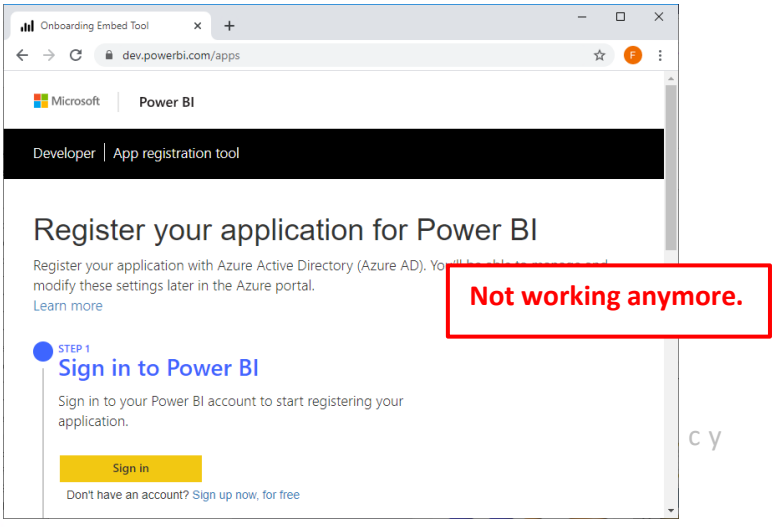
These limitations are really restrictive for many use cases: Please pay attention to the above numbers: PowerBI is only usefull if you have a *very small* data volume. In all other situations, we advise you to choose another BI solution.

The parameter **P5** lists all the columns of the "Boolean" type. This "Boolean" type is a special type from PowerBI that only accepts TRUE/FALSE/NULL values. When exporting columns from Anatella to the "Boolean" type of PowerBI, Anatella applies the following conversion rules:

Value in Anatella	Value in Power BI for the "Boolean" type
NULL or empty	NULL
Any non-zero number, y, yes, t, true, p	TRUE
0, n, no, f, false	FALSE

5.23.90.1. PowerBI first time setup

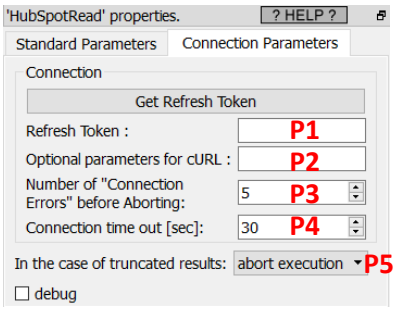
Before using Anatella to upload "Push" datasets to Microsoft PowerBI, you need to get the correct values for the parameters **P6** and **P7**. To get these 2 parameters you must follow the instructions given inside the section 5.23.27 (don't forget the step 16). This procedure takes a little bit of time but you only need to do it once to get access to ALL common Azure services (oneDrive, Email, SharePoint, PowerBI). Back in 2020, it was also possible to get the parameters **P6** and **P7** using a simplified Wizard accessible here: https://dev.powerbi.com/apps but this wizard is now broken (Thanks Microsoft!).

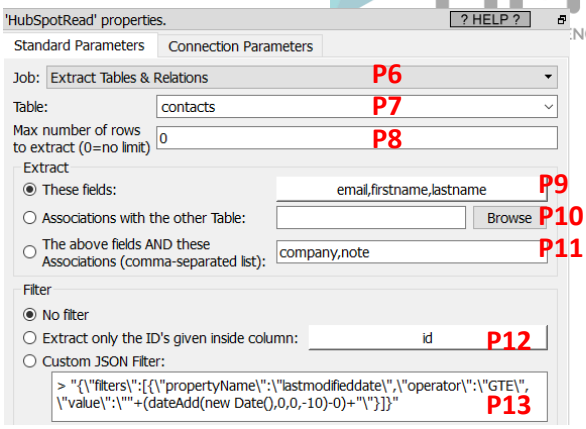


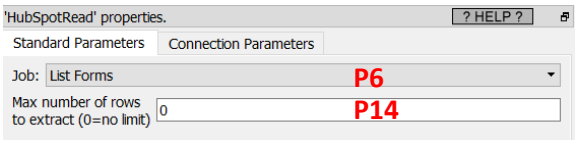
5.23.91. Download from Hubspot

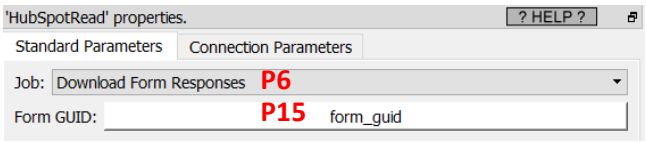
Icon:

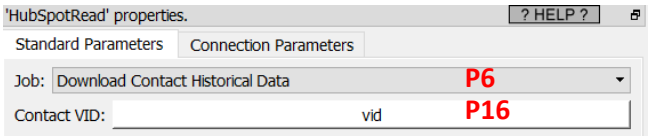
Property window:











Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup parameter **P9** for web-access through a PROXY server.

To get the parameter **P1** ("Hubspot Refresh Token"), just click the button "Get Refresh Token" inside Anatella.

All the parameters are pretty much self-explanatory with the exception of:

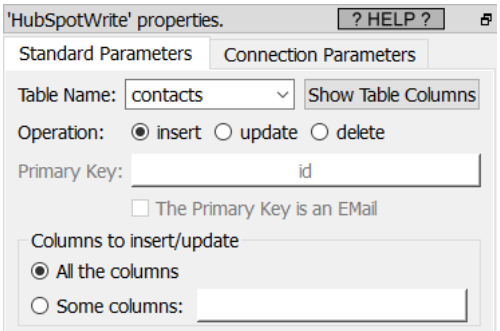
- The parameter **P13** ("Custom JSON Filter"). The default value of this parameter extracts from the "contacts" table all the rows that have been modified over the course of the last 10 days. The exact syntax of the filter expression is given here:
  https://developers.hubspot.com/docs/api/crm/search#filter-search-results
- The parameter **P5**: When using the "Custom JSON Filter" option the number of rows in output is limited to 10.000 (this is a limitation of HubSpot). If your "custom filter" returns more than 10.000 rows, the ouput data will thus be truncated. Using the parameter **P5** you can decide what Anatella must do when such truncation happens.

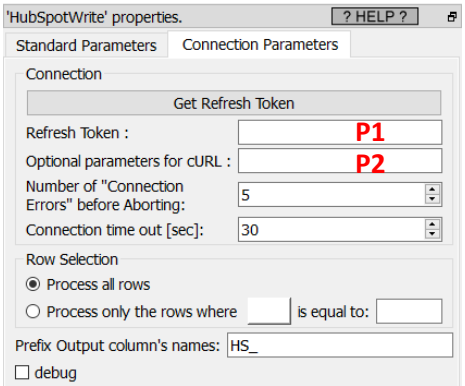
The operating mode **P11** is faster than **P10** and, also, it returns more associations!?? How is that so?

5.23.92. Upload data to Hubspot

Icon:

Property window:





Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P2** for web-access through a PROXY server.

To get the parameter **P1** ("Hubspot Refresh Token"), just click the button "Get Refresh Token" inside Anatella.

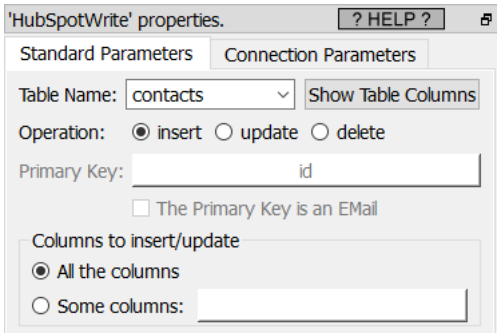
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The input column names must match the columns names (properties's names) inside Hubspot. To help you find the right column's names, you can see the exact Hubspot columns names by clicking the "Show Table Columns" button.

5.23.93. DropBox unlock

Icon:

Property window:

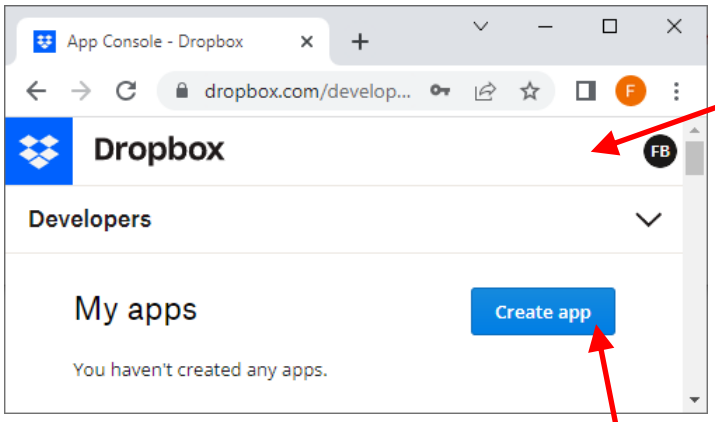


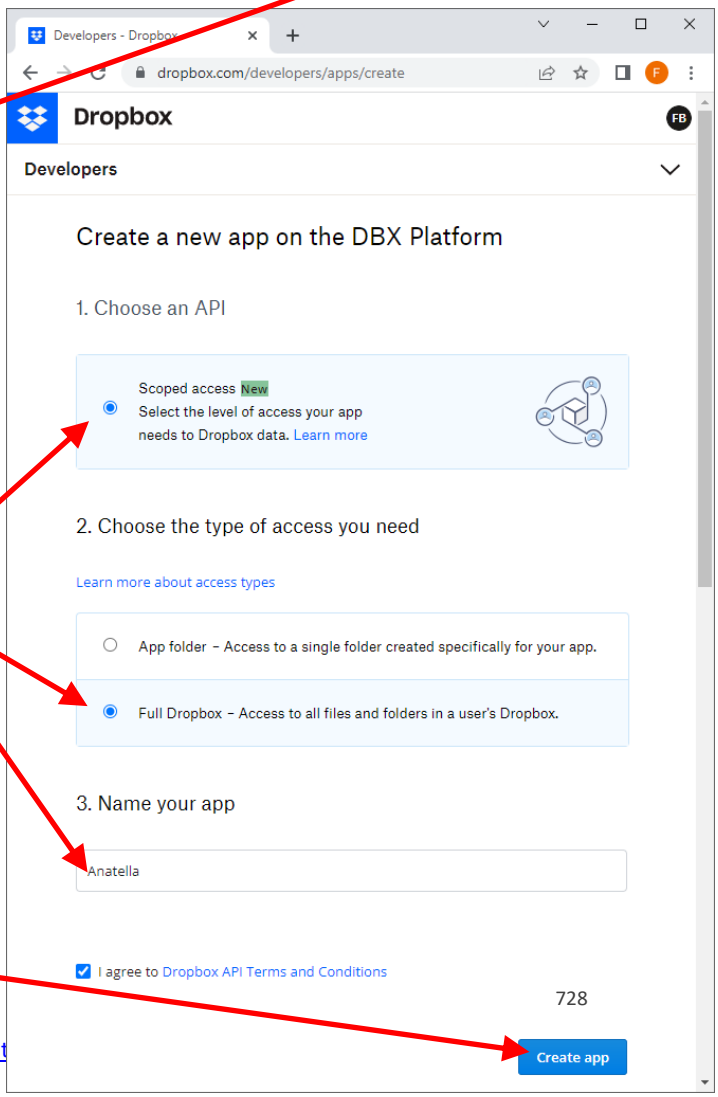
Short description:
Unlock connection to Dropbox

Long Description:
To connect to Dropbox, you need 3 parameters: Dropbox App key, Dropbox App Secret, Dropbox Refresh Token. The procedure to get these 3 parameters is:

1. First, a word of caution: You will need your "App Key", "App Secret" and "Refresh Token" to use (nearly) all the Dropbox actions inside Anatella. There are no ways to copy/paste back these 3 informations from one Anatella action to another (this is, of course, "by design"). So, you should keep yourself these 3 informations in a secure place, if you intend to re-use them later.

2. Open in your web browser "the Dropbox App Console": Go to the URL:
   https://www.dropbox.com/developers/apps
   ..and sign-in into your dropbox account. After the sign-in, you should see:

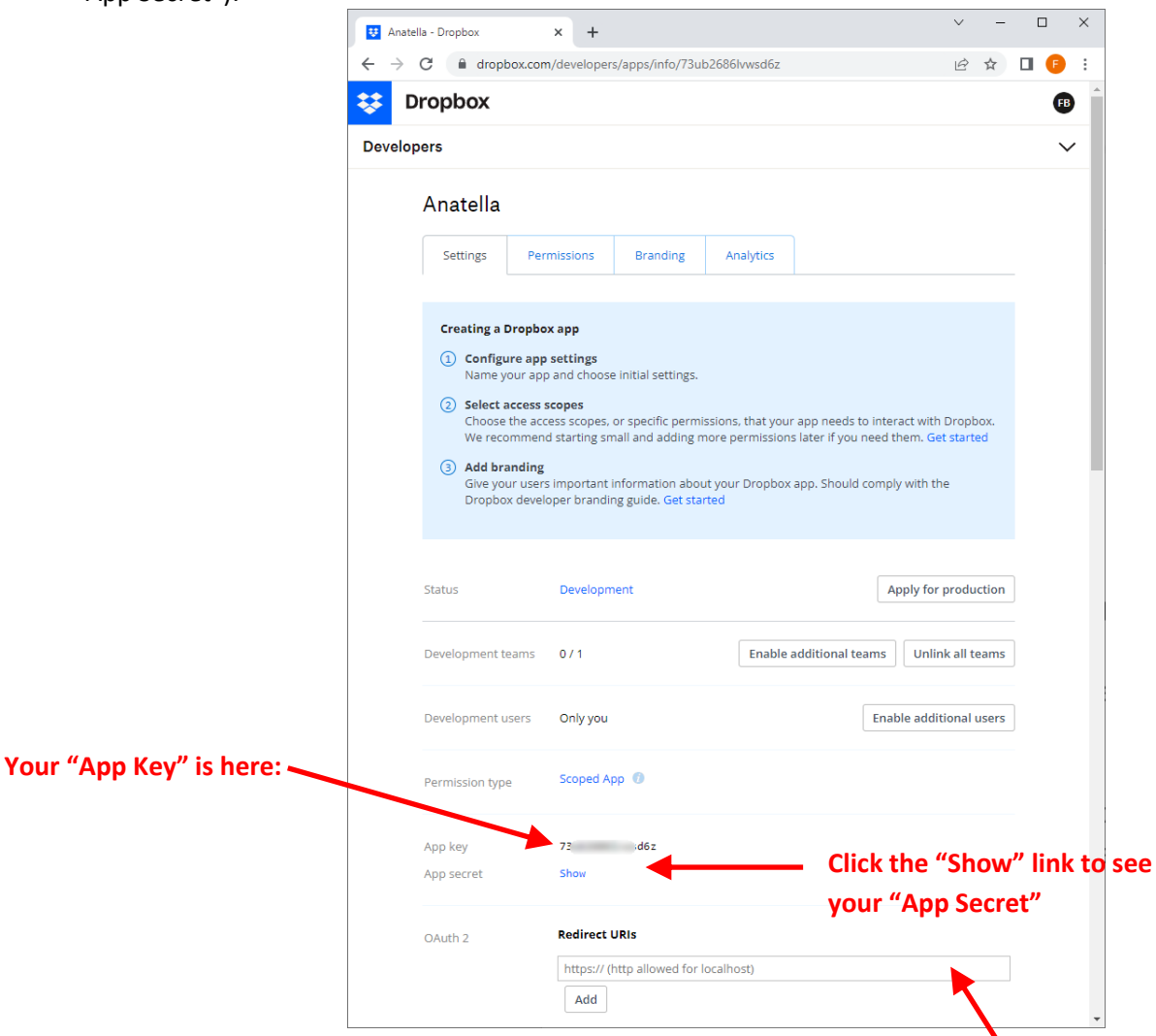




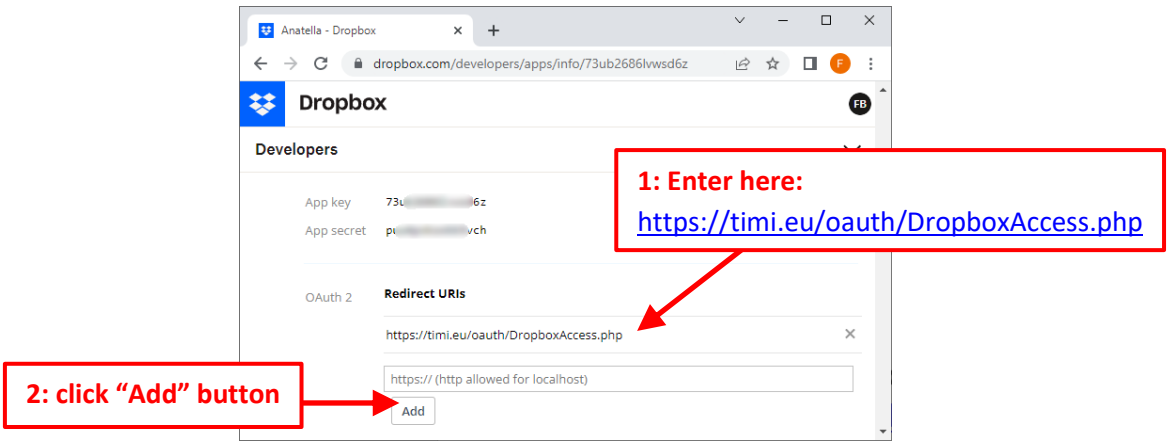
3. Create a new Dropbox app:
   3.1. Click the button "Create app":
   3.2. Click on "Scoped access":
   3.3. Click on "Full Dropbox":
   3.4. Write an application name (you can use whatever name you want):
   3.5. Click on the "Create app" button

©2010 TIMi SPRL. E-mail: sales@t

4. You now have access to the "**App Key**" and "**App Secret**" (click the "show" button to see your "App Secret"):

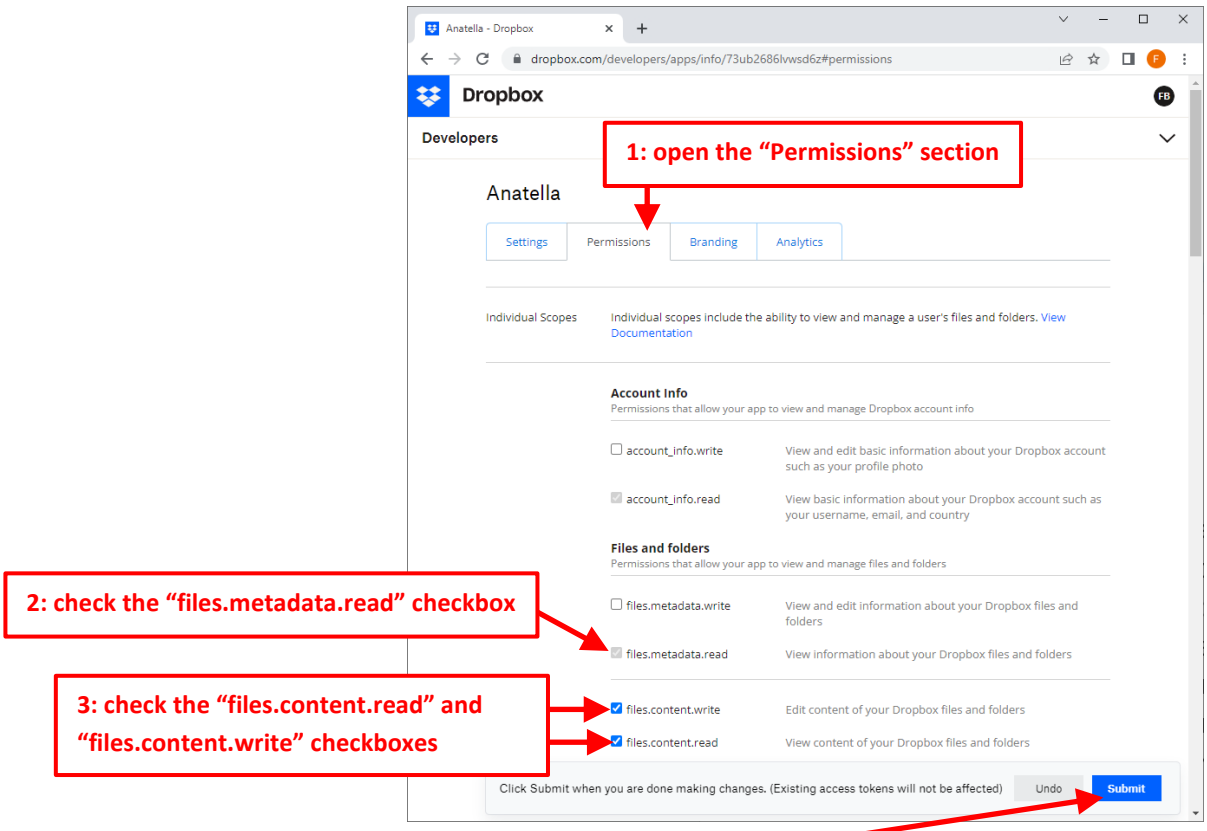


5. This step is important! Inside the OAuth2 section, add a new "Redirect URI" here:
   Enter: https://timi.eu/oauth/DropboxAccess.php
   ..and click the "Add" button. You should now have something like this:



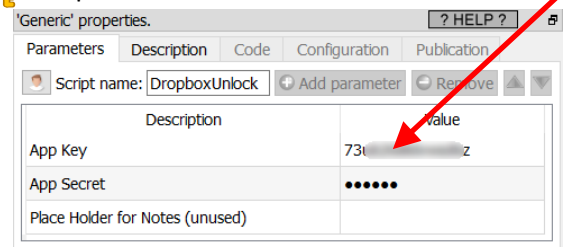
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. For security reasons, you may want to limit the scopes of your Anatella Dropbox App to the minimum. You still need to need to activate the following 3 scopes for the Anatella Dropbox actions to work properly : files.metadata.read, files.content.read, files.content.write.
This is done inside the “Permission” Section here:



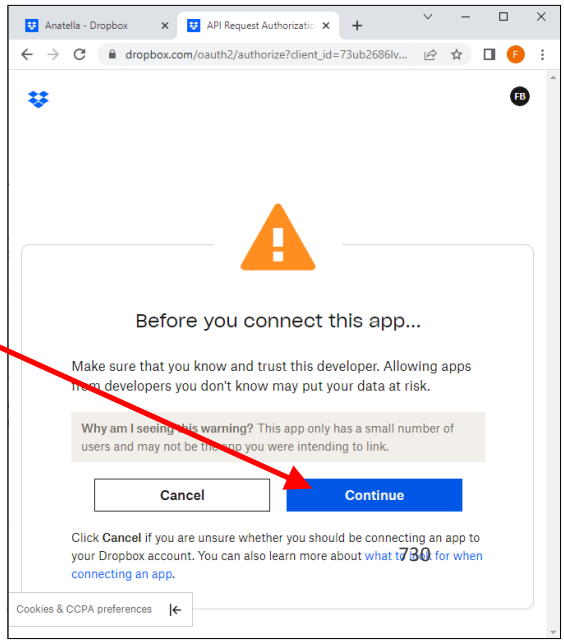
7. Click on the "Submit" button.
Your Dropbox App is now finished.

8. Copy/paste your your “App Key” and “App Secret” (obtained from step 4.) inside the DropBox Unlock action inside Anatella here:



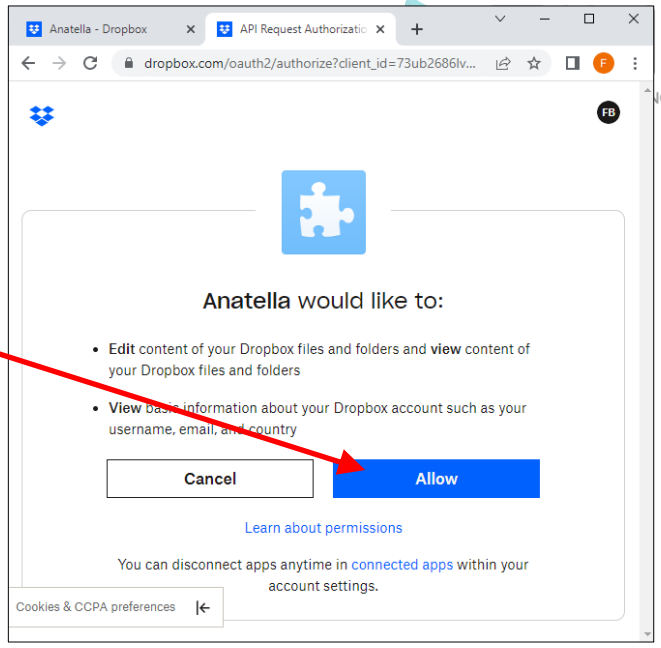
9. Run (i.e. click the output pin) the DropBox Unlock action. A web browser opens.

10. Since your (totally new) application isn’t verified, you get a warning message. This is perfectly normal and expected. Just click on the “Continue” button:

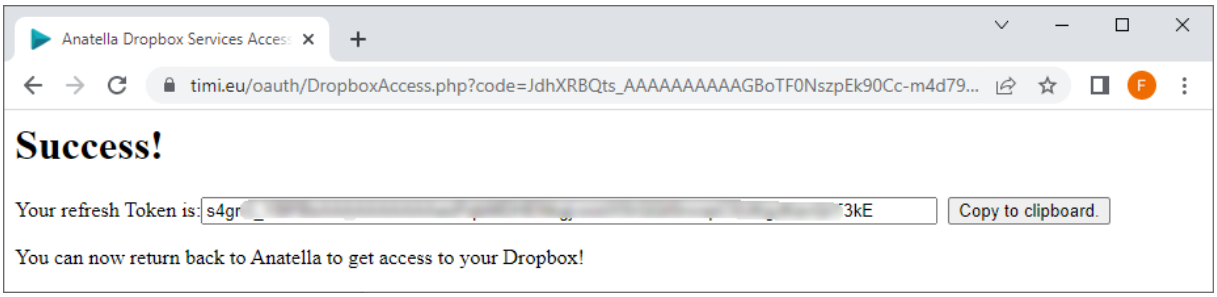


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us

11. Select the account that owns the “Dropbox” to which you want to connect to (you might need to login on Dropbox again) and confirm that you want access to your Dropbox files from Anatella: Click the "Allow" button:



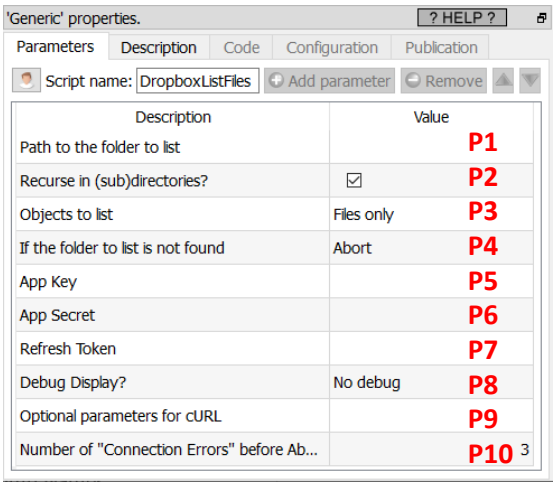
12. Finally, you receive your “Refresh Token”:



You now have your complete Dropbox credentials:

- an “AppKey” (obtained at step 4.)
- an “AppSecret” (obtained at step 4.)
- a “Refresh Token” (obtained at this step 12)

5.23.94. Dropbox List Files



Icon:

Property window:

Short description:
List the files in your Dropbox

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P9** for web-access through a PROXY server.

To use this Action, you’ll need to get 3 parameters from the Dropbox Website (i.e. you need your Dropbox App key -parameter **P5**-, Dropbox App Secret -parameter **P6**- and your Dropbox Refresh Token -parameter **P7**). Please see the section 5.23.93. for more details on how to get these parameters.

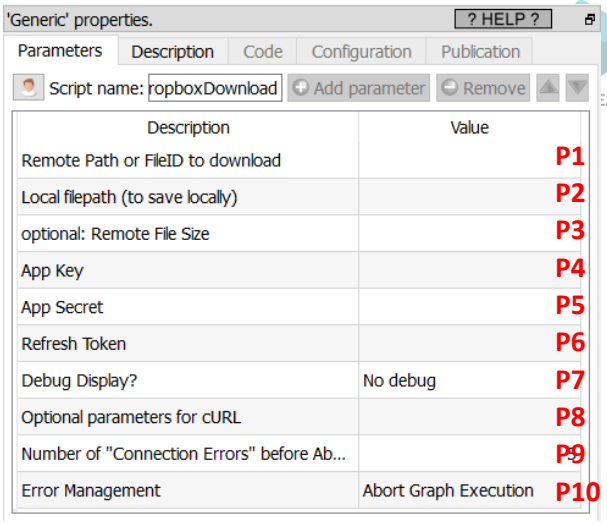
Once you have completed the “setup process” described in the section 5.23.93., you can use the parameters **P1** to **P4** to list the files in your Dropbox.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.95. Dropbox Download Files

Icon:

Property window:



Short description:
Download files from Dropbox

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

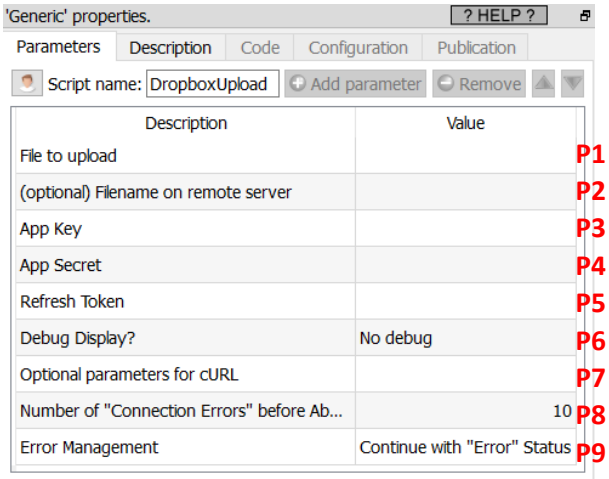
To use this Action, you'll need to get 3 parameters from the Dropbox Website (i.e. you need your Dropbox App key -parameter **P4**-, Dropbox App Secret -parameter **P5**- and your Dropbox Refresh Token -parameter **P6**). Please see the section 5.23.93. for more details on how to get these parameters.

Once you have completed the "setup process" described in the section 5.23.93., you can use the parameters **P1** and **P3** to download the required files from your Dropbox.

5.23.96. Dropbox Upload Files

Icon:

Property window:



Short description:
Upload files to Dropbox

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P6** for web-access through a PROXY server.

To use this Action, you'll need to get 3 parameters from the Dropbox Website (i.e. you need your Dropbox App key -parameter **P3**-, Dropbox App Secret -parameter **P4**- and your Dropbox Refresh Token -parameter **P5**). Please see the section 5.23.93. for more details on how to get these parameters.

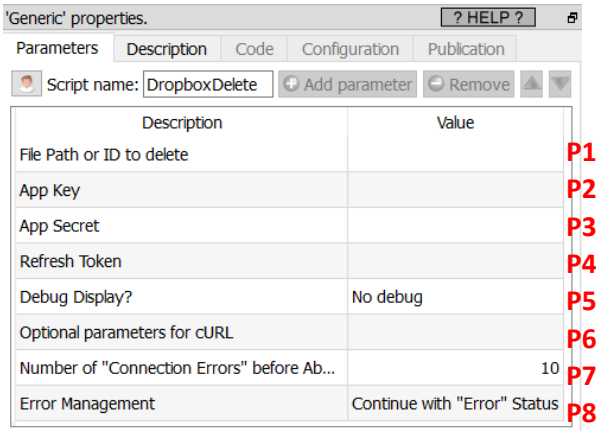
Once you have completed the "setup process" described in the section 5.23.93., you can use the parameters **P1** and **P2** to upload the required files to your Dropbox.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.23.97. Dropbox Delete Files

Icon:

Property window:



Short description:
Delete files from your Dropbox

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P5** for web-access through a PROXY server.

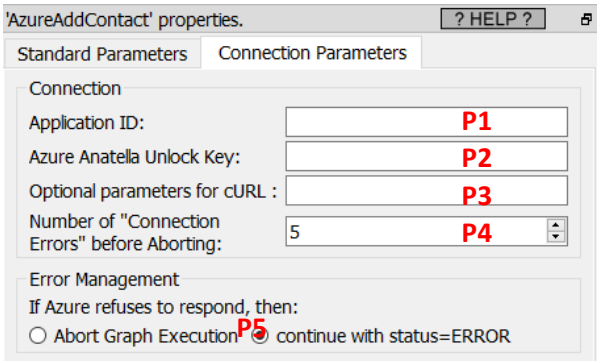
To use this Action, you'll need to get 3 parameters from the Dropbox Website (i.e. you need your Dropbox App key -parameter **P2**-, Dropbox App Secret -parameter **P3**- and your Dropbox Refresh Token -parameter **P4**). Please see the section 5.23.93. for more details on how to get these parameters.

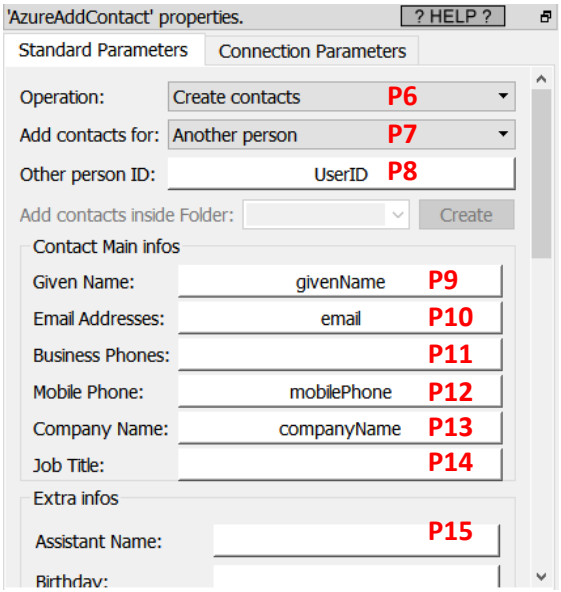
Once you have completed the "setup process" described in the section 5.23.93., you can use the parameter **P1** to delete the required files inside from your Dropbox.

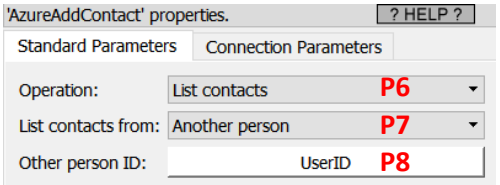
5.23.98. Azure Manage Contacts

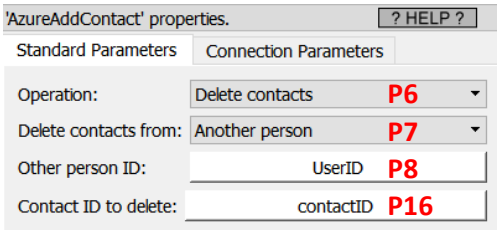
Icon:

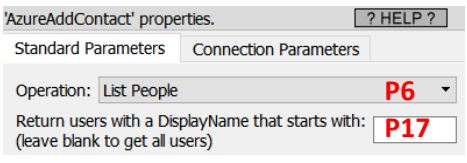
Property window:









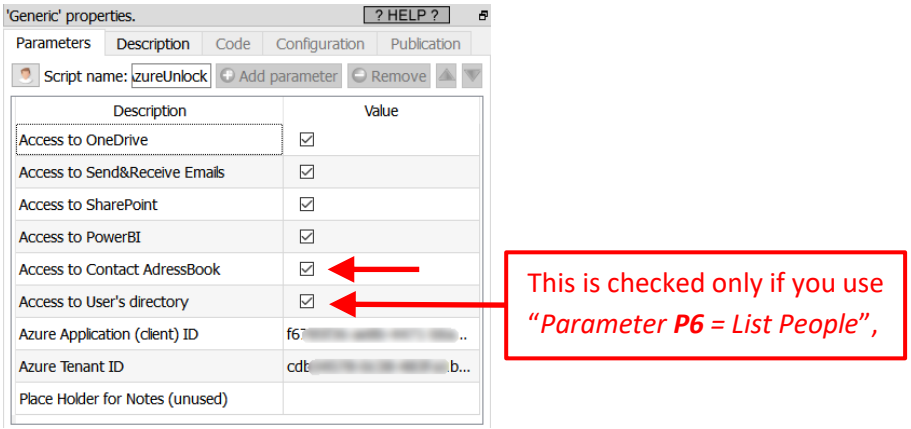


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
This Action also works when accessing the web through a PROXY server: Please consult the section 5.1.9.2. for more details on how to setup the parameter **P3** for web-access through a PROXY server.

To use this Action, you'll first need to get 2 parameters from the Azure Website: i.e. You need to obtain your "Application ID" (parameter **P1**) and your "Anatella Unlock Key" (parameter **P2**) from the Azure Website. Please see the section 5.23.27. for the exact details on how get these 2 parameters. In particular, you should follow **the step 18** from the procedure given inside section 5.23.27. Also, if you want to use the special operating mode "*Parameter **P6** = List People*", then you must also follow **the step 19** from the procedure given inside section 5.23.27.

These are the two checkboxes from the AzureUnlock Action that must be checked in order to use the AzureManageContacts Action:



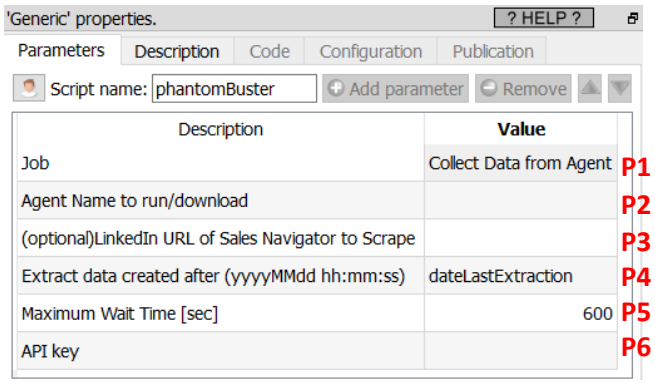
Once you have completed the "setup process" described in the section 5.23.27., you can use the parameters **P5** to **P17** to manage your contacts data on Azure.

The parameter **P7** can either be "My self" or "Another person". By default, the parameter **P7** is "My Self". When the parameter **P7** is "Another person", you need to provide, using the parameter **P8**, a column with the UserID of the "other person". To extract these UserID's from azure, you can set the Parameter **P6** to "List People" and run the Action. However, when the Parameter **P6** is "List People", you will need high-level administrative rights to setup the Azure access, so this might not always be possible, due to security reasons.

5.23.99. PhantomBuster Extraction

Icon:

Property window:



Short description:
Runs a PhantomBuster job

Long Description:
Self-explanatory

The different values for the parameter **P1** are:

- List All Agents
- Run Agent

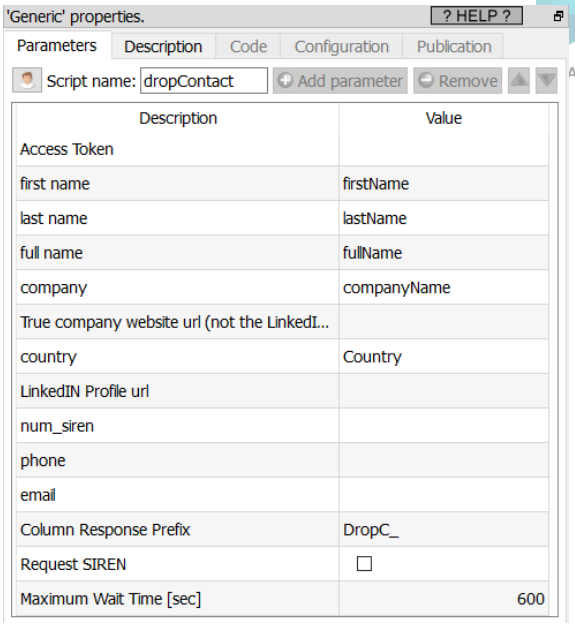
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Run Agent+Wait
- Run Agent+Wait+Collect Data
- Collect Data from Agent

5.23.100. DropContact Extraction

Icon:

Property window:



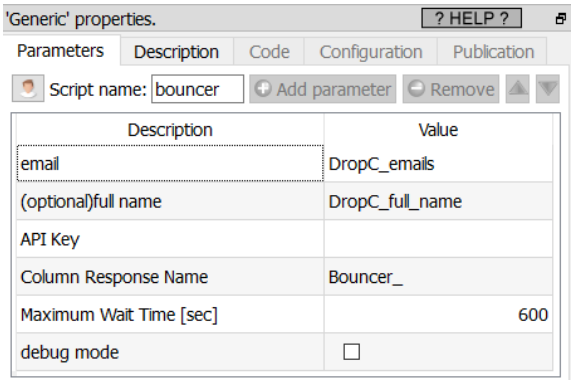
Short description:
Runs a DropContact Extraction

Long Description: Self-Explanatory

5.23.101. Bouncer Validation

Icon:

Property window:



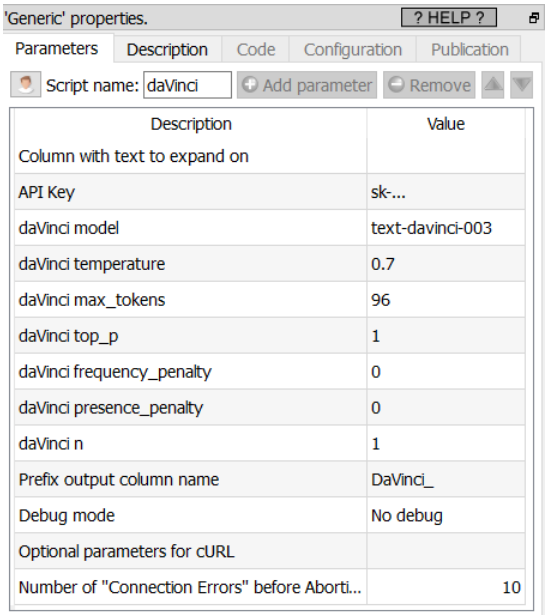
Short description:
Runs a Bouncer Validation

Long Description: Self-Explanatory

5.23.102. ChatGPT query

Icon:

Property window:



Short description:
Runs a ChatGPT query

Long Description:
Self-Explanatory

More details on the different parameters of this action are here:
https://platform.openai.com/docs/api-reference/completions/create#completions/create-model

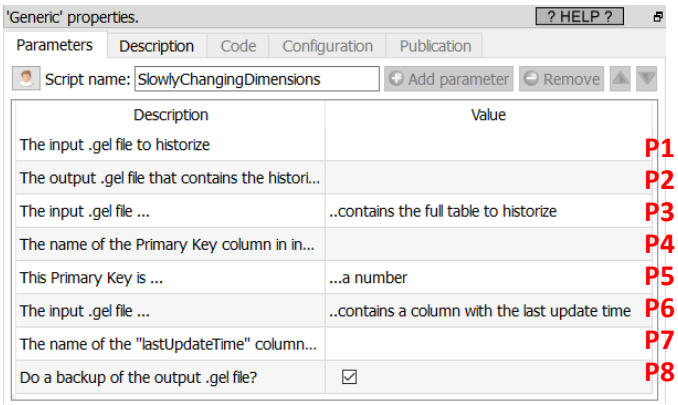
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.24. TA - Data Engineering

5.24.1. Historized Slowly Changing Dimensions

Icon:

Property window:



Short description:
Create a historized view of a SCD table

Long Description:
When engaging in predictive analytic activities, we must be able to "see" any table as it was at a specific point in time in the past (i.e. in technical terms, we must see a table at the "observation date"). Unfortunately, most operational systems do not keep an history of their tables and "going back in time" is almost never possible… …unless you use the Anatella "Slowly Changing Dimensions" Action!

This Action creates an output table that keeps tracks of all the changes made to the specified (database) table. This process is typically referred as "*Keeping a historical view of a Slowly Changing Dimension*": you'll find more details on this process here:

https://en.wikipedia.org/wiki/Slowly_changing_dimension

Typically, you run this action every day (or week) and it will look for all the changes inside an input table and logs them all inside one output table.

According to wikipedia, there exists 6 different techniques to handle slowly changing dimensions:

SCD Type	Description
Type - 0	Retaining Original: you just get one dump
Type - 1	Overwriting: you overwrite the dimension with the last version
Type - 2	Adding new rows (you keep history as rows)
Type - 3	Adding columns (you keep history as extra columns)
Type - 4	Adding new records on new table (History Table)
Type - 6	Hybrid approach (Type 1 + Type 2 + Type 3) ← **used here**

Here we will be using a 'Type - 6" implementation including :

- A surrogate (or technical) key as unique id: This is a composite key composed of the columns "SCD_Key" and "SCD_Seq".
- Active line indicator: "SCD_current_flag"
- Start date and End date: [SCD_from_date, SCD_to_date[
- Seq number of history: "SCD_Seq"
- A very basic deactive flag (only when the parameter **P3** is "*contains the full table*"): "SCD_active"

The parameter **P4** is the name of a unique key (e.g. customer key) inside the input table. This key must be unique in the input file.

Ideally, the input file should also contain a column that contains the date of the last update of a row (the date format is yyyyMMdd). The name of this "lastUpdateDate" column is given in the parameter

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

P7. If you don't have such a column, no worries: This is a very common situation: Just set the parameter **P6** to "*…does not contains a "lastUpdateTime" column (use the current date instead)*".

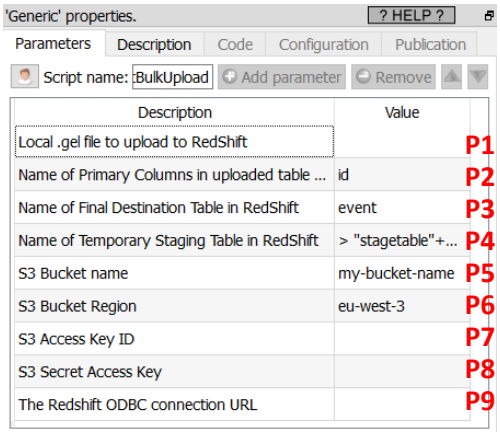
The output table contains these additional columns (in addition to all the columns from the input table):

SCD_Key	Logical key of the slowly changing dimension. Equal to the key (e.g. customer key) in the input file
SCD_seq	The sequence of statuses… first status = 1… next update = 2 …
SCD_from_date	Start moment of the validity for the record (YYYYMMDD)
SCD_to_Date	End date of the validity (YYYYMMDD)
SCD_current_flag	1 if the record is the current active record 0 if the record is a historic record
SCD_active	This is a technical active flag only relevant when the parameter **P3** is "*contains the full table*"): A = active D = deleted If the primary key is in the input file, then the status is set to A. If the primary key is missing in the input file the status will be 'D'.

5.24.2. RedShift Bulk Upload through a S3 Bucket

Icon:

Property window:



Short description:
Upload a table to RedShift

Long Description:
With the standard redshift ODBC driver, it's not possible to insert rows into RedShift. If you want to insert rows into a table inside Redshift, you need to use the current Action. This action proceeds in 5 steps:

1. Using the user-provided .gel file (parameter **P1**), it creates on the local hard drive a temporary csv file with all the data to copy into RedShift.
2. It copies the temporary csv file from the local hard drive into a S3 bucket.
3. It copies the temporary csv file from the S3 bucket to a Staging Table (parameter **P4**) into RedShift.
4. It copies the content of the Staging Table to the Final Table (parameter **P3**) into RedShift. Before the copy starts, to avoid to get any Primary Keys "in double" inside the Final Table, it removes from the Final table all the rows that have the same Primary Keys as the Primary Keys inside the Staging table. The name(s) of the Primary key column(s) is(are) given in parameter **P2**.
5. Cleaning: It deletes the staging table, the csv file inside the S3 bucket and inside the local hard drive.

To be able to connect to your S3 bucket storage, you need to get from Amazon these 4 parameters:

1. Parameter **P5**: your "Bucket Name"

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Parameter **P6**: your “Region”
3. Parameter **P7**: your “Access Key ID”
4. Parameter **P8**: your “Secret Access Key”

Please refer to the section 5.23.6. for the procedure to get these 4 parameters.

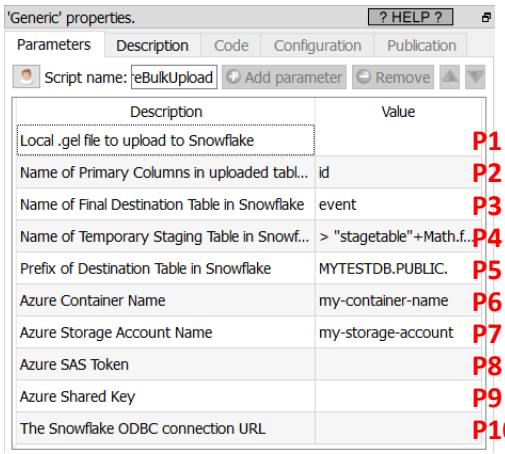
To be able to connect to your RedShift database (to run the SQL command to import the data from the S3 bucket), you need to:

1. Download & install the ODBC drivers for RedShift: see section 5.1.6.11. for more details about this step.
2. Get the ODBC connection URL to connect to your Redshift database: Enter this URL as the Anatella parameter **P9**. The procedure to get this ODBC connection URL is given here: https://docs.aws.amazon.com/redshift/latest/mgmt/configure-odbc-connection.html#obtain-odbc-url
   Typically, this ODBC connection URL looks like this:

Driver={Amazon Redshift (x64)}; Server=redshift-cluster-1.chhydacago64.eu-west-3.redshift.amazonaws.com; Database=dev; UID=<my_user>; PWD=<my_password>

5.24.3. SnowFlake Bulk Upload through an Azure Storage

Icon:



Property window:

Short description:
Upload a table to SnowFlake

Long Description:
Inserting rows into Snowflake using the standard ODBC connector is horribly slow.

To get more speed, you can use the current Action. This action proceeds in 5 steps:

1. Using the user-provided .gel file (parameter **P1**), it creates on the local hard drive a temporary csv file with all the data to copy into SnowFlake.
2. It copies the temporary csv file from the local hard drive into an Azure Storage.
3. It copies the temporary csv file from the Azure Storage to a Staging Table (parameter **P4**) into SnowFlake.
4. It copies the content of the Staging Table to the Final Table (parameter **P3**) into SnowFlake. Before the copy starts, to avoid to get any Primary Keys “in double” inside the Final Table, it removes from the Final table all the rows that have the same Primary Keys as the Primary Keys inside the Staging table. The name(s) of the Primary key column(s) is(are) given in parameter **P2**.
5. Cleaning: It deletes the staging table, the csv file inside the Azure Storage and inside the local hard drive.

To be able to connect to your Azure storage, you need to get from Azure these 4 parameters:

1. Paramater **P6**: your “Azure Container Name”
2. Paramater **P7**: your “Azure Storage Account NAme”

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Paramater **P8**: your “Azure SAS Token”
4. Paramater **P9**: your “Secret Sahred Key”

Please refer to the section 5.23.30.1. for the procedure to get these 4 parameters.

To be able to connect to your SnowFlake database (to run the SQL command to import the data from the S3 bucket), you need to:

1. Download & install the ODBC drivers for RedShift. You can download these drivers from us here: http://download.timi.eu/ODBC/ODBC_drivers_SnowFlake/
2. Get the ODBC connection URL to connect to your Snowflake database: Enter this URL as the Anatella parameter **P10**. The instructions to get this ODBC connection URL is given here: https://docs.snowflake.com/en/user-guide/odbc-parameters.html#configuration-parameters
   Typically, this ODBC connection URL looks like this:

Driver={SnowflakeDSIIDriver}; Server=dy47396.eu-west-1.snowflakecomputing.com; Database=MYTESTDB; UID=<my_user>; PWD=<my_password>

5.24.4. SnowFlake Bulk Upload through a S3 Bucket

Icon:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: 3BulkUpload | Add parameter | Remove

Description	Value	
Local .gel file to upload to Snowflake		P1
Name of Primary Columns in uploaded table (c...	id	P2
Name of Final Destination Table in Snowflake	event	P3
Name of Temporary Staging Table in Snowflake	> "stagetable"+M...	P4
Prefix of Destination Table in Snowflake	MYTESTDB.PUBLIC.	P5
S3 Bucket name	my-bucket-name	P6
S3 Bucket Region	eu-west-3	P7
S3 Access Key ID		P8
S3 Secret Access Key		P9
The Snowflake ODBC connection URL		P10

Property window:

Short description:
Upload a table to SnowFlake

Long Description:
Inserting rows into Snowflake using the standard ODBC connector is horribly slow.

To get more speed, you can use the current Action. This action proceeds in 5 steps:

1. Using the user-provided .gel file (parameter **P1**), it creates on the local hard drive a temporary csv file with all the data to copy into SnowFlake.
2. It copies the temporary csv file from the local hard drive into a S3 bucket.
3. It copies the temporary csv file from the S3 bucket to a Staging Table (parameter **P4**) into SnowFlake.
4. It copies the content of the Staging Table to the Final Table (parameter **P3**) into SnowFlake. Before the copy starts, to avoid to get any Primary Keys “in double” inside the Final Table, it removes from the Final table all the rows that have the same Primary Keys as the Primary Keys inside the Staging table. The name(s) of the Primary key column(s) is(are) given in parameter **P2**.
5. Cleaning: It deletes the staging table, the csv file inside the S3 bucket and inside the local hard drive.

To be able to connect to your S3 bucket storage, you need to get from Amazon these 4 parameters:

1. Paramater **P6**: your “Bucket Name”
2. Paramater **P7**: your “Region”
3. Paramater **P8**: your “Access Key ID”

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. Paramater **P9**: your "Secret Access Key"

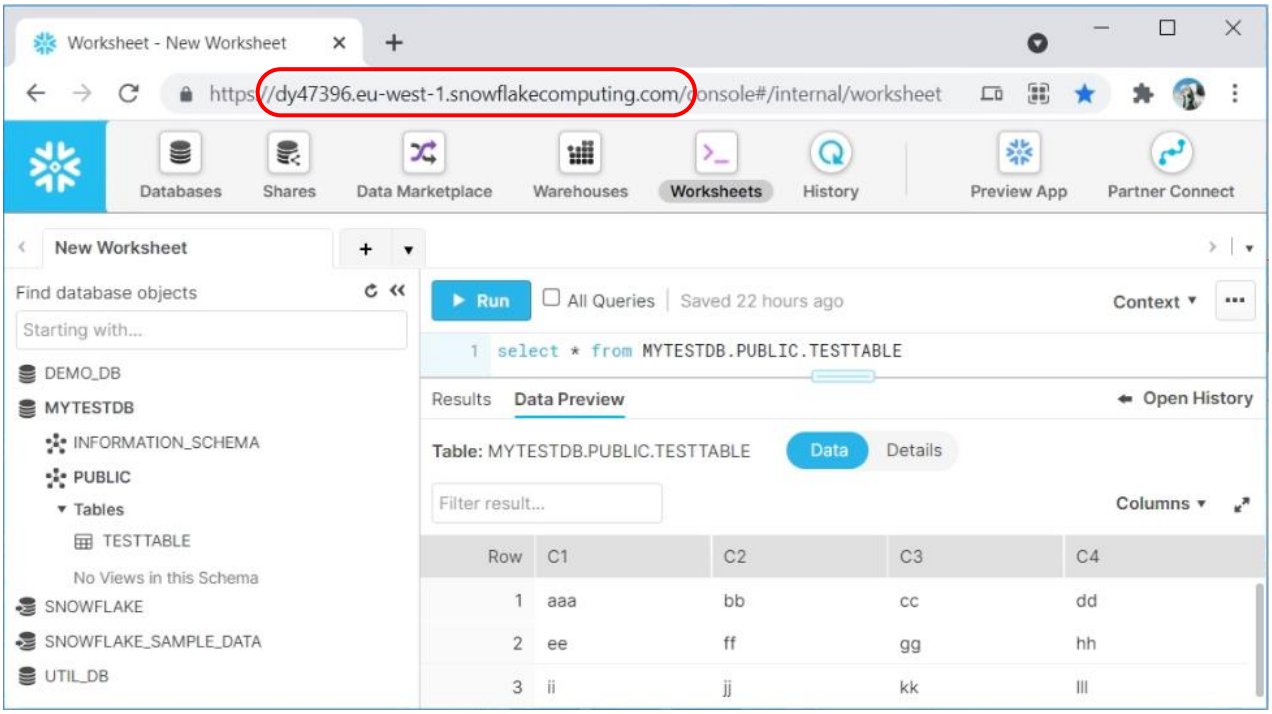
Please refer to the section 5.23.6. for the procedure to get these 4 parameters.

To be able to connect to your SnowFlake database (to run the SQL command to import the data from the S3 bucket), you need to:

1. Download & install the ODBC drivers for RedShift. You can download these drivers from us here: http://download.timi.eu/ODBC/ODBC_drivers_SnowFlake/
2. Get the ODBC connection URL to connect to your Snowflake database: Enter this URL as the Anatella parameter **P10**. The instructions to get this ODBC connection URL is given here: https://docs.snowflake.com/en/user-guide/odbc-parameters.html#configuration-parameters
   Typically, this ODBC connection URL looks like this:

Driver={SnowflakeDSIIDriver}; Server=dy47396.eu-west-1.snowflakecomputing.com; Database=MYTESTDB; UID=<my_user>; PWD=<my_password>

   To get the value of the "server" parameter inside the above ODBC connections string, you just need to look at the URL used to query your SnowFlake instance:

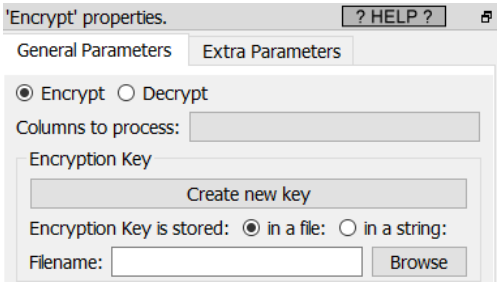


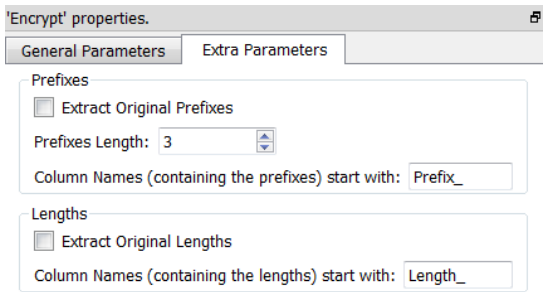
5.25. TA – Other (TA=Transformations Actions)

5.25.1. Encrypt (High-Speed action)

Icon:

Property window:





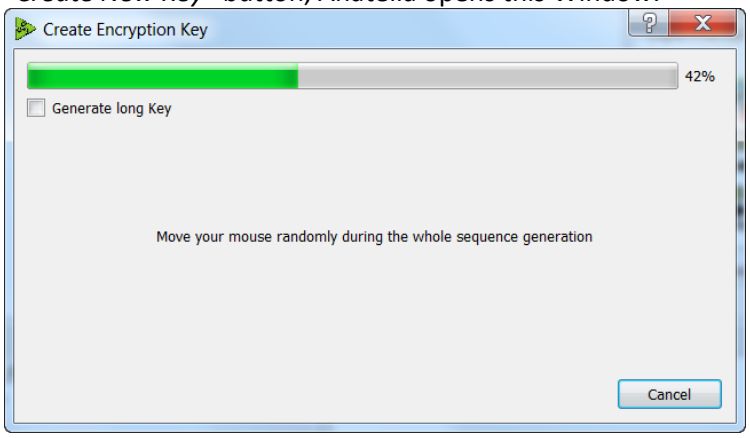
Short description: Encrypt/Decrypt some fields.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Encryption and Decryption are based on a symmetric key (i.e. there is one unique key that allow encryption and decryption). The key is save inside a "Key File". You need a "Key File" to encrypt or decrypt your data.

To select a "Key File" (using the "*Browse*" button) or to create a new "Key File" (using the "*Create New Key File*" button), you need to switch to "Expert-user-mode". To switch to expert-user-mode: Click the button in the main toolbar of the application! Once you are in "expert-user-mode", the "*Browse*" button and the "*Create New Key*" button are enabled.

When you click the "*Create New Key*" button, Anatella opens this Window:



By moving randomly your mouse inside this window, you generate random numbers that are used to create a 100% random key. This encryption key is saved inside a "Key File" or inside a string (this last option allows to use a "Global Parameter" to specify the key when decrypting the data).

NOTE:
Never lose your "Key File". If you lose your key file, you'll never be able to decrypt your data later.

NOTE:
Never send your "Key File" to third parties.

NOTE:
The encryption algorithm that is used is DES (for the short keys) and 3DES (for the long keys). It's a well-studied encryption algorithm that does not seem to have any weakness.

The encryption algorithm used inside Anatella is symmetric. This guarantees that there will never be any "collisions". For example: Let's assume that you are encrypting many MSISDN (i.e. many phone numbers): because there are no collisions, the number of distinct MSISDN before and after encryption is the same. There will never be 2 different un-encrypted MSISDN that are "mapped" to the same encrypted MSISDN (i.e. there are no collisions, never).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Since there are no collisions, you can safely use the encrypt Action to anonymize your datasets.
In particular, when anonymizing datasets containing MSISDN numbers, you'll lose, after encryption, some precious information about the MSISDN. The lost information is:

- Is it a "short" phone number? (e.g. like the voice-mail number)
- Is it an international call?

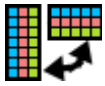
These pieces of information are *__very__* important when analyzing communication-graphs using SNA (Social Network Analysis) algorithms. You can use:

- The "*Extract Original Prefixes*" option to keep un-encrypted the first few digits of the MSISDN (This allows to detect international calls).
- The "*Extract Original Lengths*" option to save the length of the un-encrypted MSISDN (This allows to detect "short" phone number like the voice-mail number).

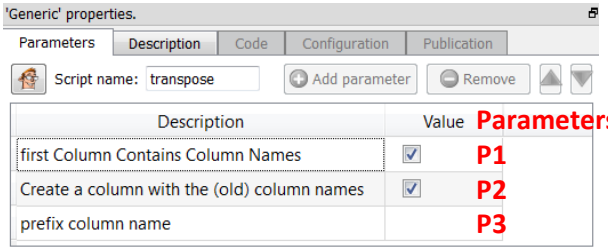
Anonymizing a dataset using a non-symetric encoding (such as MD5) can lead to some "collisions". Non-symetric encodings (such as MD5) are thus **bad and dangerous alternatives** when anonymizing some dataset.

Let's take an example. Let's assume that you are anonymizing 2 million different MSISDN using a 5-characters-MD5-code. A 5-characters-MD5-code can only have, at maximum, 1 million different values (=16^5). This means that you will have a *__catastrophic__* number of collisions that will make your anonymized dataset completely useless (Actually, even if you use, on the same population, a 6-character-MD5-code, there are 99% chance that you'll also have so many collisions that your anonymized dataset is also useless).

5.25.2. Transpose

Icon: 

Property window:

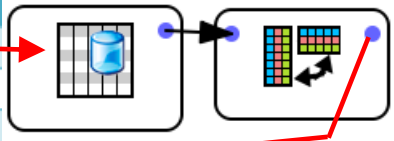


Short description:
Transpose the whole input table.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:
Here is an example:

names	value	count
a	10	1
b	12	2
c	14	3

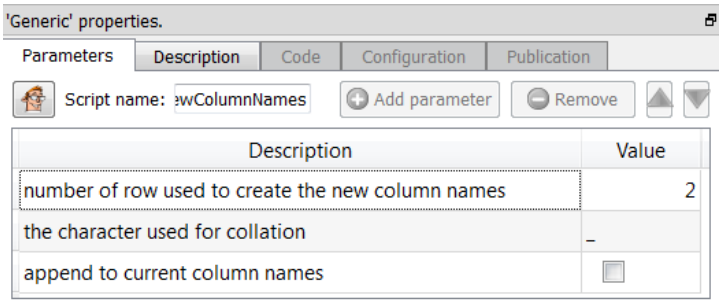


Output Table		Parameter P1: First Column Contains (New) Column Names	
		True (checked)	False (unchecked)
Parameter P2: Create a column with the (old) column names	True (checked)	names | a | b | c value | 10 | 12 | 14 count | 1 | 2 | 3	OldColumnNames | C1 | C2 | C3 names | a | b | c value | 10 | 12 | 14 count | 1 | 2 | 3
	False (unchecked)	a | b | c 10 | 12 | 14 1 | 2 | 3	C1 | C2 | C3 a | b | c 10 | 12 | 14 1 | 2 | 3

This Action only works on small input tables because it requires loading into the RAM memory the whole table.

5.25.3. Extract New Column Names

Icon:



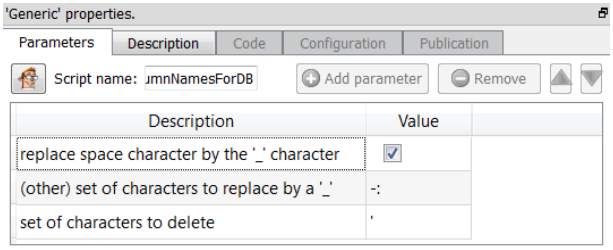
Property window:

Short description:
Extract new column names using the content of the first few rows of the dataset.

Long Description:
This box will concatenate the text from the N first rows to create the column's names. Make sure to unselect the option "Append to current column names" to avoid having C1, C2, … Cx in your names.

5.25.4. Column Names for DB

Icon:



Property window:

Long Description:
Process the column names so that they can be used inside a database.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.25.5. File List from Observation Date

Icon:

Property window:

'Generic' properties. ? HELP ?

Parameters | Description | Code | Configuration | Publication

Script name: mObsDate | Add parameter | Remove

Description	Value	Parameters:
Observation date is:	The Date from today	P1
Fixed Observation Date	> ObservationDate	P2
Date Format	yyyyMMdd	P3
Filename Prefix	data_	P4
Filename Suffix	.gel_anatella	P5
Offset relative to the Observation Date	-10	P6
Number of output rows (maximum)	10	P7
Date Unit / Date Increment	Day	P8
Test Files Existance	Skip Files that already exist	P9
Column Name	myFile	P10

Short description:
Generate a File List.

Long Description:

This Action is typically used in collaboration with a ReadGel Action, a MergeSortInput Action, a multipleDownAndUpLoad Action or a loopAnatellaGraph Action (see an example of usage in section 5.20.6).

The objective of the fileListFromObsDate Action computes a list of .gel_anatella filenames (or .txt filenames, or .xlsx filenames, or .json filenames, etc.). The maximum length of the list (i.e. the maximum number of files) is specified using the parameter **P7**. The size from the list can be smaller than the given parameter **P7** because of the parameter **P9** that might remove from the list some specific files. The output of the Action is a table with one column named "File" (the column's name can be changed using the parameter **P10**).
The filenames inside this list have a specific structure that is composed of 3 different parts:

- Part 1: a constant filename **prefix** (see parameter **P4**)
- Part 2: a date in the "yyyyMMdd" format, by default (this can be changed using the parameter **P3**).
- Part 3: a constant filename **suffix** (see parameter **P5**)

To compute the "date" part from the filenames we'll use 4 parameters: **P1, P2, P6, P8**.

The "date part" starts at a specific time and increases, on each row, by the increment given in parameter **P8**. The starting time is computed in this way:

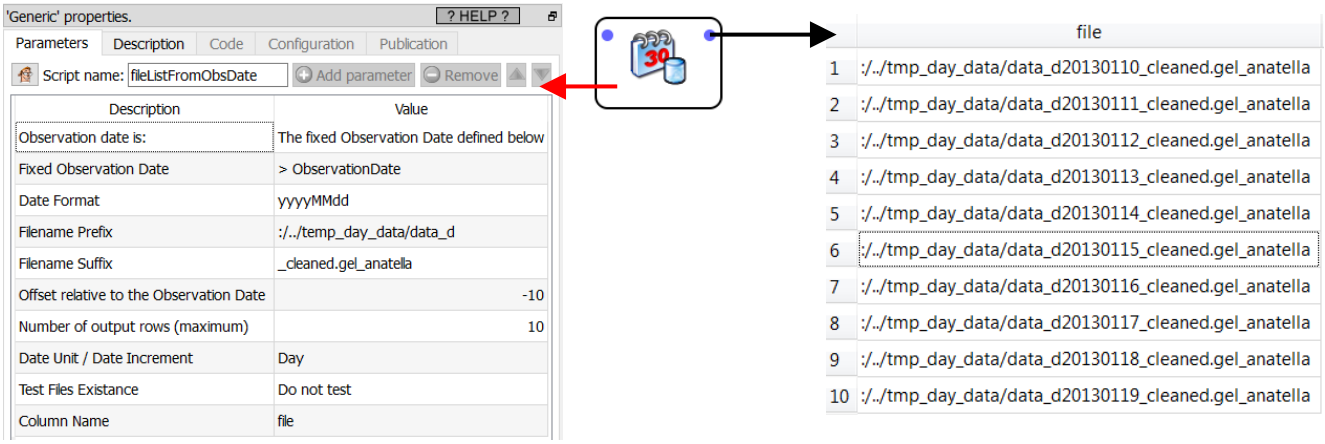
- The parameter **P1** is equal to "*The date from Today*":
  The starting time is the **date from today** plus the offset given in parameter **P6**. The unit of the parameter **P6** is given in parameter **P8**.

- The parameter **P1** is equal to "*The fixed Observation Date defined below*":
  The starting time is the **date given as parameter P2** plus the offset given in parameter **P6**. The unit of the parameter **P6** is given in parameter **P8**.

  By default, the parameter **P2** is initialized to have to value of the Graph-Global-Parameter named "ObservationDate": See section 4.7.1. and section 9.4.2. about "Graph-Global-Parameters". By default, the Graph-Global-Parameter "ObservationDate" is a date expressed in the "yyyyMMdd" format (see section 5.1.3. about "date formats"). The Parameter **P3** allows you to use another date-format.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

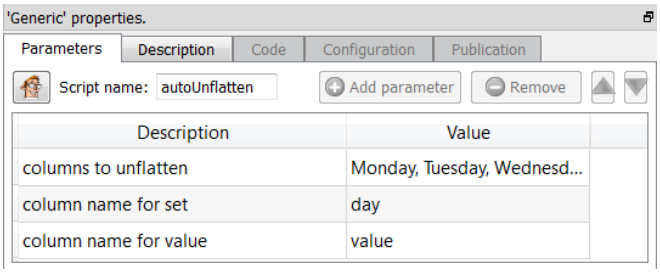
For example, if we used the above settings and we set “ObservationDate”= 20130120, we obtain as output a file list with 10 files and the “last” file is just before the “ObservationDate”:



5.25.6. Auto unflatten

Icon:

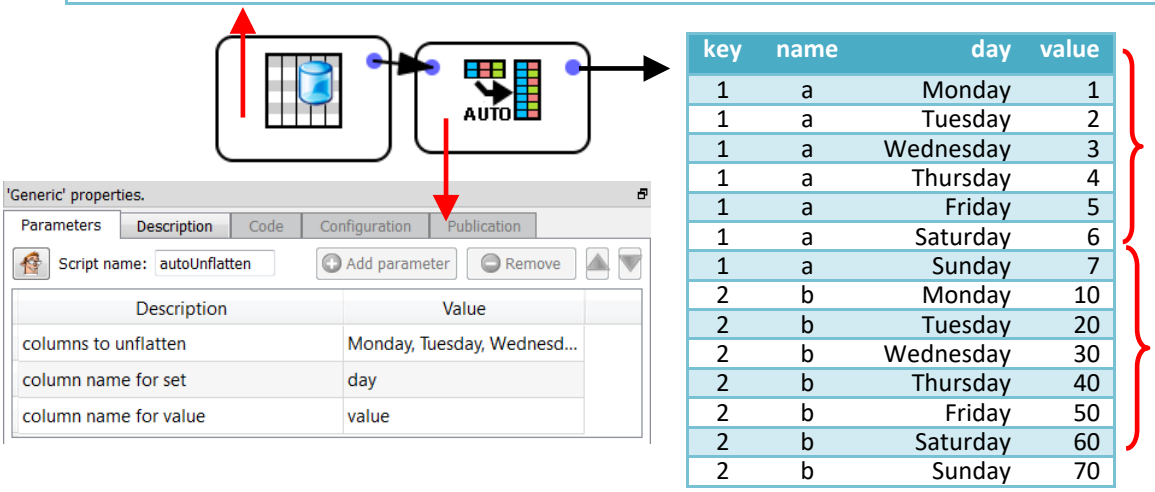
Property window:



Short description:
Execute an “unflatten” operation where each “set” is composed of one column.

Long Description:
Here is an example:

key	name	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	a	1	2	3	4	5	6	7
2	b	10	20	30	40	50	60	70



key	name	day	value
1	a	Monday	1
1	a	Tuesday	2
1	a	Wednesday	3
1	a	Thursday	4
1	a	Friday	5
1	a	Saturday	6
1	a	Sunday	7
2	b	Monday	10
2	b	Tuesday	20
2	b	Wednesday	30
2	b	Thursday	40
2	b	Friday	50
2	b	Saturday	60
2	b	Sunday	70

The

AutoUnflatten Action is slower than the C++ UnFlatten Action (because it’s a Javascript-based Action). Thus, if possible, use instead the C++ UnFlatten Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.26. User-Deprecated Transformations

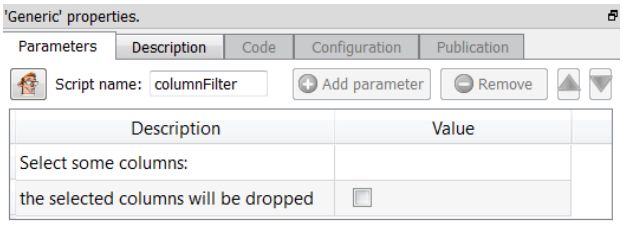
Typically, the end-user of Anatella (I mean: You!) will develop many tailor-made JavaScript/R/Python actions. Most of these new Actions will evolve over time. Some of them will become extremely useful and used on a daily basis. Other Actions will become obsolete and will never be used again. Anatella offers you a complete versioning system that allows you to manage easily all your JavaScript Actions: see section 9.8. for more information at this subject.

The obsolete actions do still have some interest because they can still be used as a “starting point” when developing new JavaScript actions. You can’t thus delete these obsolete Actions because they could still be useful but you don’t want to mix them with the other “good” Actions that you are using every day. The solution is to “mark” the obsolete Actions as “deprecated”. A deprecated action won’t be mixed with the other “good” actions because they have their own panel inside the Anatella Interface, thus there is no risk of erroneously using them.

5.26.1. Column Filter

Icon:

Property window:



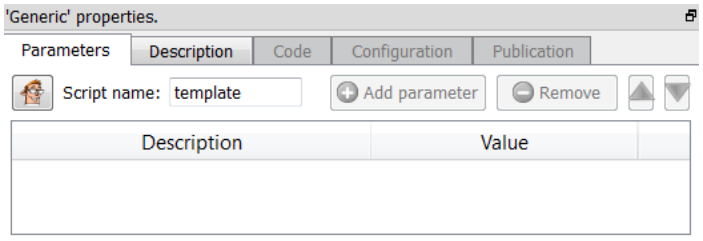
Short description:
Remove some columns

Long Description:
This operator removes some columns of the input table. You can also use this Action to re-order the columns of your table: see section 5.5.7 for a detailed explanation.

5.26.2. Template

Icon:

Property window:

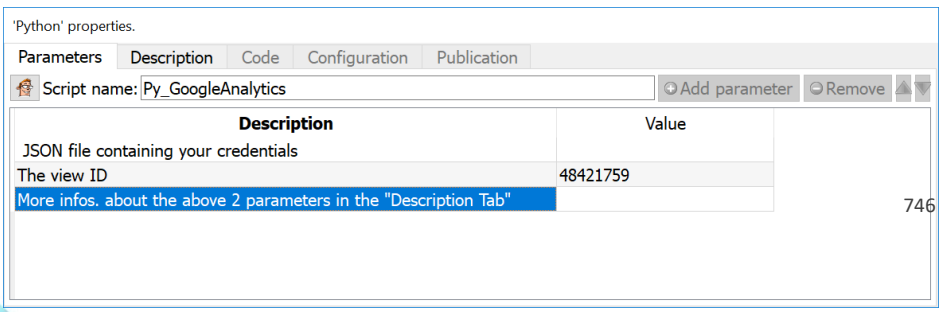


Short description:
Template

Long Description:
This Action simply "pass by" the rows from the input pin to the output pin without doing any operation. This Action is simply a "Basic Template" that you can "tweak" to create your own, complex Actions.

5.26.3. PyGoogle Analytics (Python action)

Icon:



Property window:
Short description:
Connect to Google Analytics

Long Description:

WARNING: This action is deprecated: You should rather use instead the Google Analytics Action from the section 5.22.23.

Before executing this Action, please install the "Google API" inside Python by executing inside the windows shell:

pip.exe install --upgrade google-api-python-client
The "pip.exe" executable is usually inside the directory "<root_of_python_dir>/scripts".

To create your JSON file (with your credentials), the procedure is:

1. You need to first use the setup tool (https://console.developers.google.com/start/api?id=analyticsreporting.googleapis.com&credential=client_key), which guides you through creating a project in the Google API Console, enabling the API, and creating credentials.
2. Open the Credentials page (https://console.developers.google.com/apis/credentials): Click Create credentials and select "Service Account Key": select "JSON format" to store your key and download the JSON file.
3. Open you JSON file inside a text file editor and search for the field: "client_email": This should be something like: "xxx@yyy.iam.gserviceaccount.com",
4. Use the email address obtained on point 3 (just here above) to add a user (https://support.google.com/analytics/answer/1009702) to the Google analytics account you want to access via the API. For this Action only Read & Analyze (https://support.google.com/analytics/answer/2884495) permissions are needed for this user.

The above 4-steps procedure (to get your JSON file with your credentials) is also documented here: https://developers.google.com/analytics/devguides/reporting/core/v4/quickstart/installed-py

It's quite easy to change the Python code to:

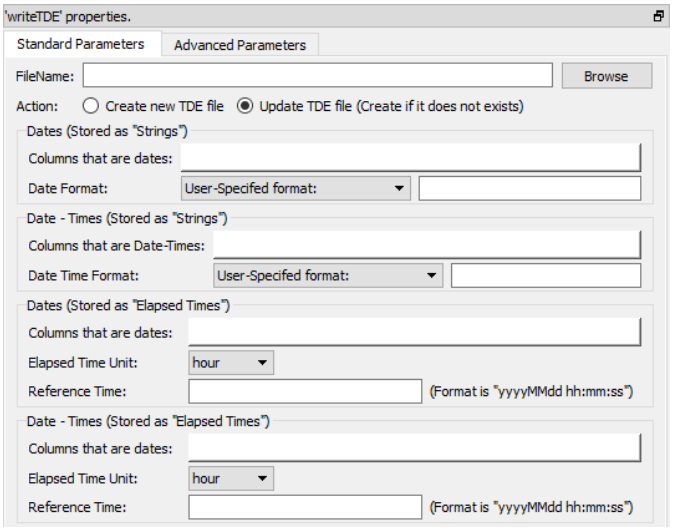
- change the analyzed period of time (change "dateRanges"): By default, we are looking at the last 30 days.
- extract new DIMENSIONS and new METRICS out of Google Analytics:
  All the available DIMENSIONS and METRICS are visible here: https://www.googleapis.com/analytics/v3/metadata/ga/columns?pp=1
  (see the comments in the code to know where to change the code)

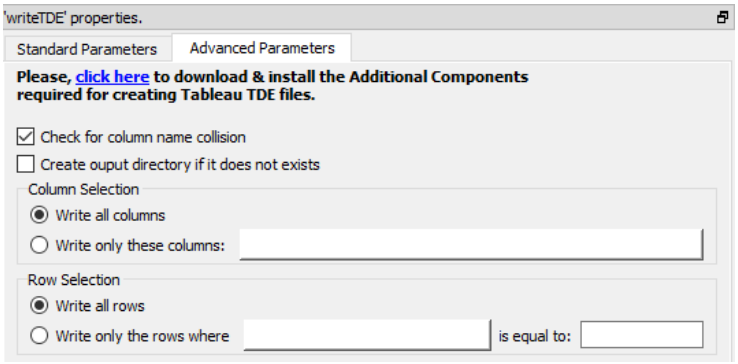
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.26.4. Create old Tableau Hyper File (.hyper) or Tableau Data Extract (.tde) file

Icon:

Property window:





Short description:
Create/Update Tableau Data Extract (TDE) or old Hyper files.

Long Description:
Self-explanatory.

WARNING: This action is deprecated: You should rather use instead the writeHyper Action from the section 5.27.16.

Once you have created (or updated) your .hyper file (or your .tde file), you can easily copy (i.e. "publish") your .hyper file to your remote tableau server using the

"Tableau Publish" action (see section 5.22.5. for more details on this action).

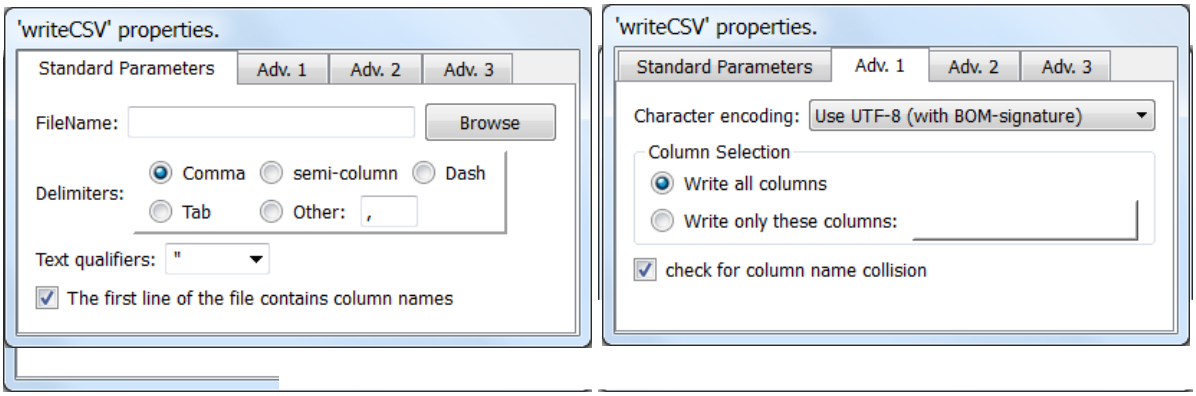
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.27. Output Actions

5.27.1. CSV file writer

Icon:

Property window:



Short description:
Writes to a Text/CSV file

Long Description:
Please refer to section 5.1.1 to have more information on how to specify the filename of the Text/CSV file (i.e. You can use relative path and Javascript to specify your filename).

When writing to a Text/CSV file, Anatella converts the internal Unicode Character representation (UTF16) to, either:

- Standard UTF-8
- The default character encoding used inside your Windows OS.

Anatella must convert (i.e. "cast") all the columns of the floating point data-type (see section 5.1.2 about data type) to the String data-type before writing them inside a Text/CSV file. The parameter that defines the notation used for converting numbers to string is the "*Float to String Cast*" parameter. It can either be: %g, %.16g, %5.3f, %f. Please refer to section 5.5.1.3 to know more about this subject.

The Text/CSV format is simple but not 100% standardized. Here are some examples that clarifies the usage of the "Text Qualifiers" into the Text/CSV files generated with Anatella (in these examples the delimiter is the comma and the "Text Qualifier" is the double-quote):

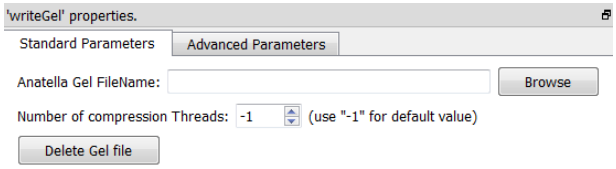
Content of a column inside Anatella	Content of the Text/CSV file
a"b	a"b
a,b	"a,b"
a,"b	"a,""b"
"ab	"""ab"

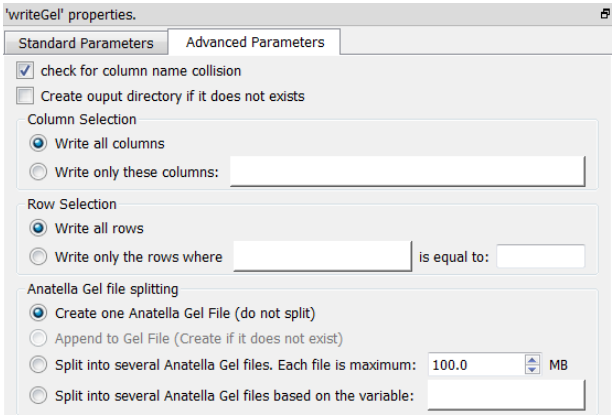
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.27.2. Anatella "Gel file" writer

Icon:

Property window:





Short description:

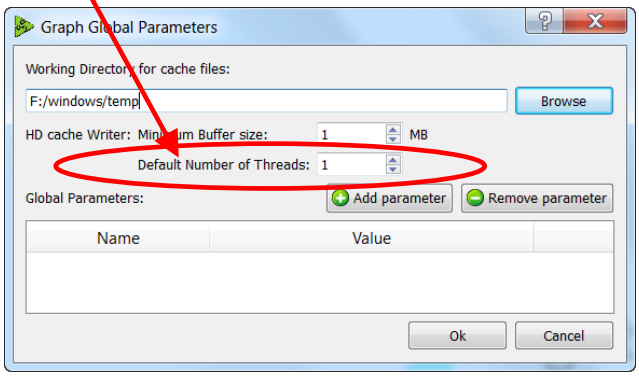
Writes a table to a ".anatella_gel" file

Long Description:

Please refer to section 5.1.1 to have more information on how to specify the filename of the .gel_anatella file (i.e. You can use relative path and Javascript to specify your filename).

The .gel_anatella files contain data in a proprietary compressed format. The maximum de-compression speed (when reading a .gel_anatella file) is around 600 MB/sec (uncompressed) (600 MB/Sec of uncompressed data is roughly equal to 170 MB/sec of compressed data because the compression ratio is usually between 1:3 and 1:4). The typical compression speed (on a Intel Core I7 processor) is around 60MB/sec (uncompressed data) (when using one thread). This (relatively) slow compression speed can sometime become the main "bottleneck" inside your data-transformation graph. To avoid such situation, you can decide to use several threads/CPU's for data compression. For example, when using 3 CPU's, you get around 3x60=180 MB/sec of compression speed.

You can set the number of CPU's used for compression using the "Number of compression Threads" parameter. When this parameter is "-1", Anatella uses the default value given inside the "Graph Global Parameters" window:



The Anatella automatic "HD cache" system also creates "Gel Files" when you click on the output pins of the different Actions. These "Gel Files" are also useful when you want to exchange data with other Anatella processes: See section 5.3.3. about this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella has a special behaviour when you click the output pin of the **WriteGel Action**. The output pin of the WriteGel Action is slightly different than the output pins of ALL the other Actions and this can sometime be a little bit disturbing, if you don't expect that.

These are the three main differences:

Difference 1. When clicking the output pin of the WriteGel Action and as long as the Anatella graph is running, you don't get any data preview. Once the data transformation is stopped, you get the data-preview as usual.

Difference 2. When you "split" the input table in several different Anatella files (using the "*Anatella Gel File Splitting*" option), you won't get any data preview when clicking the output pin of the WriteGel Action (Instead, if you still want a data-preview, simply click the output pin of the previous or the next Action).

Difference 3. When you activate the "column selection" option (to write inside your .gel_anatella file a (sub)selection of some specific columns of the input table), the data-preview that you obtain when clicking the output pin of the WriteGel Action only includes the selected column (and not ALL the columns). Nevertheless, the WHOLE table (including ALL the columns) is still propagated to the "next" Action.

About Blocking (and Non-blocking) I/O Algorithm

A typical algorithm that writes some rows inside a file on the Hard Drive follows these 4 steps:

1. Receive from the input pin of the GelFileWriter Action one row to write on the hard drive.
2. Copy the content of the row inside a large RAM buffer.
3. When the large RAM buffer is "full":
   - Compress the RAM buffer to get one data-block, containing compressed data.
   - Compute the CRC code used to validate the data-block when reading the ".gel_anatella".
   - Write the data-block to the Hard Drive.
4. If there are no more rows to write on the input pin one of the GelFileWriter Action, stop here. Otherwise, go back to step 1.

In terms of speed, the above algorithm is not very efficient because, it's a "blocking" I/O algorithm: i.e. When the large RAM buffer is "full", the Action connected to the input pin of the GelFileWriter Action cannot send any more rows (i.e. because there is no space left in the RAM buffer to store them!). Instead, this transformation

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

must block until the content of the large RAM buffer is "flushed" on the hard drive: More precisely, it must:

- Block until the compression of the data-block is completed.
- Block until the computation of the CRC code used to validate of data-block is completed.
- Block until the write of the data-block in the Hard Drive is completed.

…and then (and only then) the data-transformation unblocks and it can process the next rows.

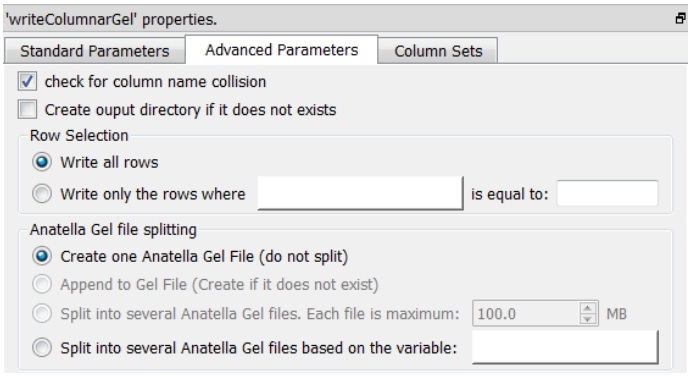
Anatella always use a (better) non-blocking I/O algorithm to create all the ".gel_anatella" file. The GelFileWriter Action always uses one (or more) dedicated compression thread(s) when writing the .gel_anatella files to the hard drive. This means that all the tasks from the step 3 herabove will be performed continuously using "one (or several) background thread(s)", so that the data-transformation-process never blocks (i.e. it will never block if enough dedicated compression threads are used).

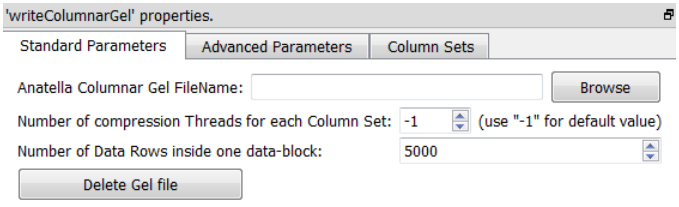
When reading or writing Gel Files (row-based ".gel_anatella" or columnar ".cgel_anatella", all I/O's inside Anatella are non-blocking.

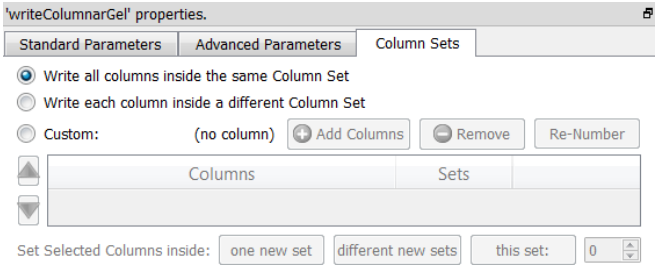
5.27.3. Anatella "Columnar Gel" file writer (column based)

Icon:

Property window:







Short description:

Write a table to a columnar ".cgel_anatella" file

Long Description:

Write a table to a columnar ".cgel_anatella" file (and to the associated "column set" data files "*.NNN.cs_anatella"). Please refer to section 5.1.1 to have more information on how to specify the filename of the .cgel_anatella file (i.e. You can use relative path and Javascript to specify your filename).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

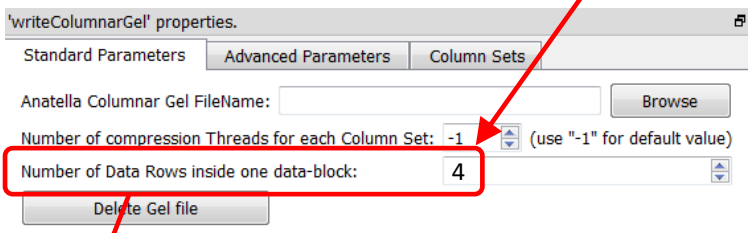
The writeColumnarGel Action has all the nice functionalities from the simpler WriteGel Action: In particular:

- Inside the writeColumnarGel Action, all the I/O routines are also non-blocking: See the previous section (i.e. the comments at the end of the previous section 5.27.2.) about "*blocking and non-blocking I/O algorithms*".
- If the compression algorithm used to create the ".cgel_anatella" becomes the bottleneck inside the data-transformation-graph, you can decide to use more threads (and thus more computing power) to compress faster the data, to remove any bottleneck.

The Columnar file format is optimized to reduce to the minimum the quantity of bytes extracted from the Hard Drive when reading a subset of the columns and a subset of the rows from the columnar file (i.e. this is the main objective of the "columnar" format!).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is an example: Let's assume that we want to save the following table: on the hard drive using the writeColumnarGel Action with the parameter "*Number of Rows inside one data-block*"=4:



One data-block is 4 rows of data

Input: a Table:

	Name	Revenue	Acquisition Date
Data-block 1	Frank	10	2001
	Sabrina	20	2000
	Daniel	30	2002
	David	15	2001
Data-block 2	Simmon	40	2005
	Bob	35	2003
	Morane	20	2008
	Tony	50	2009
Data-block 3	Stark	55	2014
	Fred	15	2010
	Freeman	5	2011

Output: "Column set" Data File:

Frank	0
Sabrina	1
Daniel	2
David	3
10	4
20	5
30	6
15	7
2000	8
2001	9
2001	10
2002	11
Simmon	12
Bob	13
Morane	14
Tony	15
40	16
35	17
20	18
50	19
2003	20
2005	21
2008	22
2009	23
Stark	24
Fred	25
Freeman	26
55	27
15	28
5	29
2014	30
2010	31
2011	32

Output: Main ".cgel_anatella" file:

Range on Name	Range on Revenue	Range on Acquisition Date	Offset Information
[Daniel ; Sabrina]	[10 ; 30]	[2000 ; 2002]	0;4;8;12
[Bob ; Tony]	[20 ; 50]	[2003 ; 2009]	12;16;20;24
[Fred ; Stark]	[5 ; 55]	[2010; 2014]	24;27;30;33

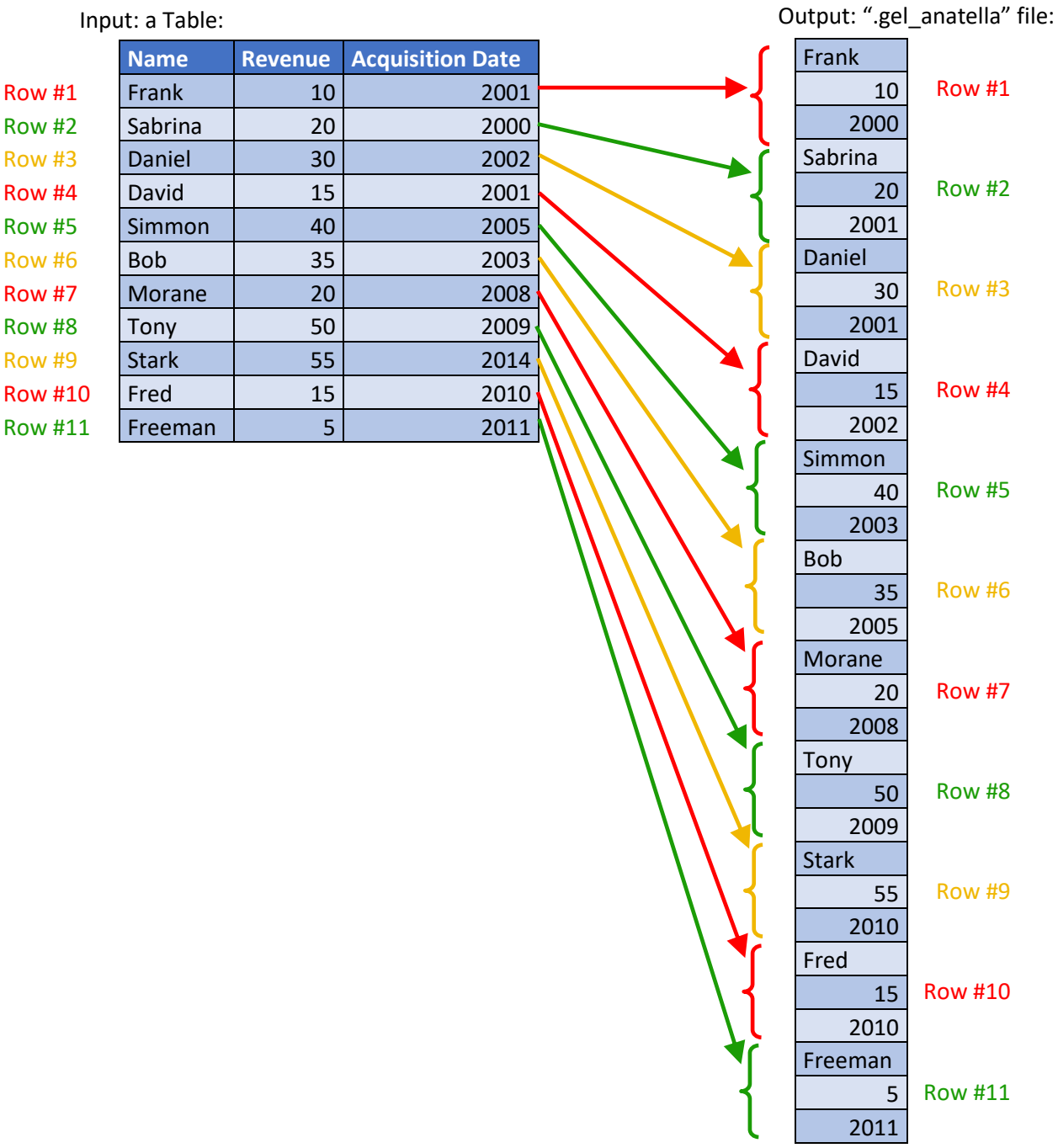
As output, on the hard drive, we actually get 2 files:

- The first file contains some meta-data about the table. This is the "main" file: It has the extention ".cgel_anatella". The content of this file is illustrated here:
- The second file really contains the "payload": i.e. The content of all the columns from the table. This file is a "column set file": it has the extention ".cs_anatella". The content of this file is illustrated here:

Offset inside the "Column Set" data file:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Just as a comparison, the row-based ".gel_anatella" file format is illustrated this way:



Let's now assume that we want to read the above table using the columnar files illustrated inside the above drawing (i.e. using the ".cgel_anatella" file and the ".cs_anatella" file). More precisely, we want to:

- **Read the column "Name"**

  Looking inside the main ".cgel_anatella" file (more precisely, looking at the field "Offset Information"), we see that the content of the "name" column is stored inside the "column set" file at some precise locations (i.e. at the offsets ((0 to 4), (12 to 15), (24 to 27)) inside the "column set" file). Thus, we'll only extract out of the Hard Drive, from the "column set" file, these precise locations (in reality, in addition to extracting data from the hard drive, we also

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

need to decompress the data, because all the columnar files are compressed) and we get our result: The complete "name" column.

Please note that we did NOT read the complete "column set" file to get the "name" column (i.e. we skipped extracting the data at the offsets ((4 to 12) and (15 to 24)). This is where the real "speed gain" comes from, when using the columnar file: You don't have to read the whole file to get one column.

Unfortunately, the story does not ends here. Although you specifically asked to the Microsoft Windows Operating System to skip reading the bytes located at the offsets ((4 to 12) and (15 to 24)), the Hard Drive is still reading these bytes!!? How is that possible? This odd behavior originates from an "optimization" performed at the OS-level.

Let's step back a little and ask ourselves the following question:

> "*Let's assume that we just read 2GB out of a 10GB file.*
> *What are now the chances that we'll read one more megabyte?*".

These chances are pretty high: I would say that there is 99% chance that we'll continue reading the file. Based on this simple principle, Microsoft decided to modify the "*internal routine that reads the files from the hard drive*" to read "in advance" the next bytes of the file (although you didn't specifically asked for these bytes *yet*). In this way, when your program finally asks for these bytes, there are already available in RAM, ready to be used. This is a nice simple "trick" to increase disk access speed.

Actually, this "trick" to increase the hard drive efficiency is used by both Microsoft Windows and by some Manufacturers of Hard Drive (inside their Low-Level Hardware Drivers). No all manufacturers use it (because it's not trivial to code because it implies some intricate knowledge of the NTFS disk structure), but still…

Because of this "trick", it's actually the same time to read the whole data file from start to finish than to read the file "skipping" some bytes from time-to-time: *This "trick" means that the columnar-speed-gain is basically lost!* (…at least under Windows: Linux does not have this very nice trick, up to now).

To really benefit from the columnar-speed-gain, we need to place the content of each column inside different ("Column Set") data file, like this:

"Column set" File 0: (contains the "Name" column)

Frank
Sabrina
Daniel
David
Simmon
Bob
Morane
Tony
Stark
Fred
Freeman

"Column set" File 1: (contains the "Revenue" column)

10
20
30
15
40
35
20
50
55
15
5

"Column set" File 2: (contains the "Acquisition Date" column)

2000
2001
2001
2002
2003
2005
2008
2009
2010
2010
2011

This second solution is faster (and thus better). It has 4 files (instead of 2):

- There is always the main ".cgel_anatella" file, just as before.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- There are three "Column set" data files (one file for each different column).

Now, when we want to read the "Name" column, we'll only extract from the Hard Drive the bytes from the "Column set File 0" (and also the bytes from the main ".cgel_anatella" file, but this is negligible). When using this second solution, the "trick" performed by the Microsoft Windows Operating Systems plays in our advantage and really increase the reading speed (because it reads "in advance" the bytes that we really need). Everything now *seems* ok, but the story is not yet finished.

- **Read the "Name" column and the "Revenue" column.**

Let's assume again that each column is stored inside its own "column set" file. Thus, we again have:

"Column set" File 0: (contains the "Name" column)

Frank
Sabrina
Daniel
David
Simmon
Bob
Morane
Tony
Stark
Fred
Freeman

"Column set" File 1: (contains the "Revenue" column)

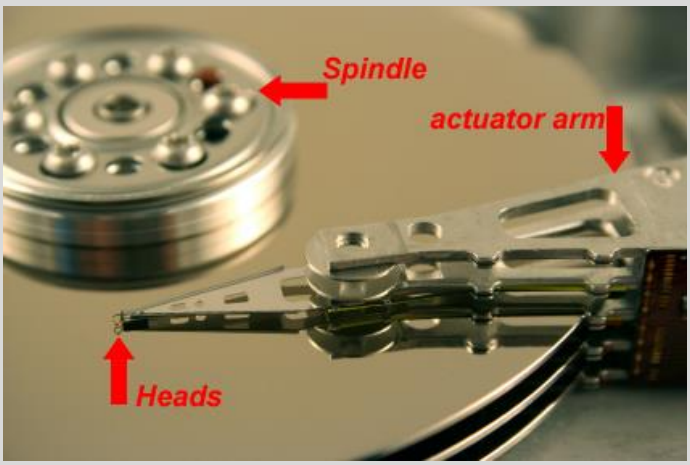
10
20
30
15
40
35
20
50
55
15
5

"Column set" File 2: (contains the "Acquisition Date" column)

2000
2001
2001
2002
2003
2005
2008
2009
2010
2010
2011

To extract from the Hard Drive the content of the "Name" column and of the "Revenue" column, we must read 2 different files. This means that the "*Heads of the Hard Drive*" will have to physically "jump" from one file to the other and this is typically a very slow operation that might producing, at the end, a slower overall reading speed.

Here is a photograpy of one of the heads (There are usually many heads inside a hard drive) that read the data out of the hard drive disk (platters):

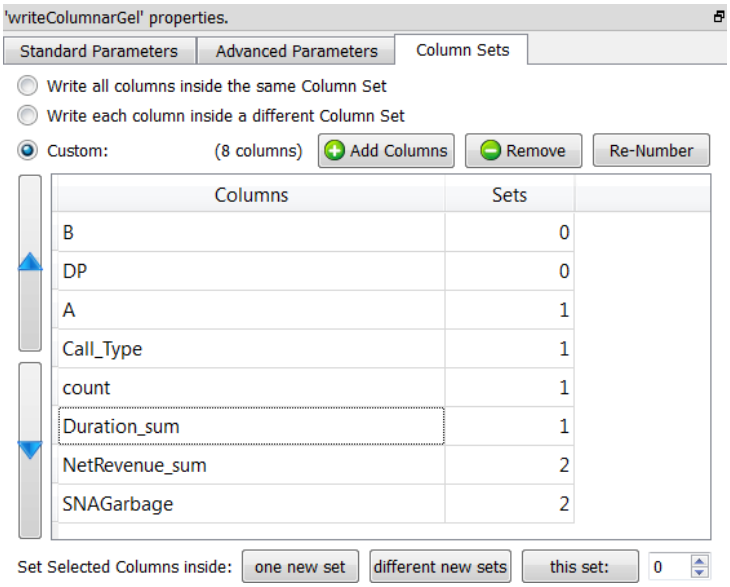


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Thus, we can't have too many different "column sets" data file opened at the same time (because "jumping" from one file to the other is actually slowing us down). Thus, we need to find a good compromise:

- Too many data files (e.g. Each different column is inside a different "column set" data file) slows us down (because the "hard drive heads" are "jumping around").
- Too few data files (e.g. All the columns are inside one "column set" data file) is bad because we'll involuntary extract from the hard drive many bytes that we don't need.

You can easily test different "compromises" using the Anatella interface (to find the one that leads to the best performances). This interface allows you to quickly & easily, in a few mouse-clicks, select which columns are placed inside which "column set" data file. For example:



… The above screenshot will produce 3 "column set" data files. The first "column set" data file (with index 0) contains only the columns named "B" and "DP".

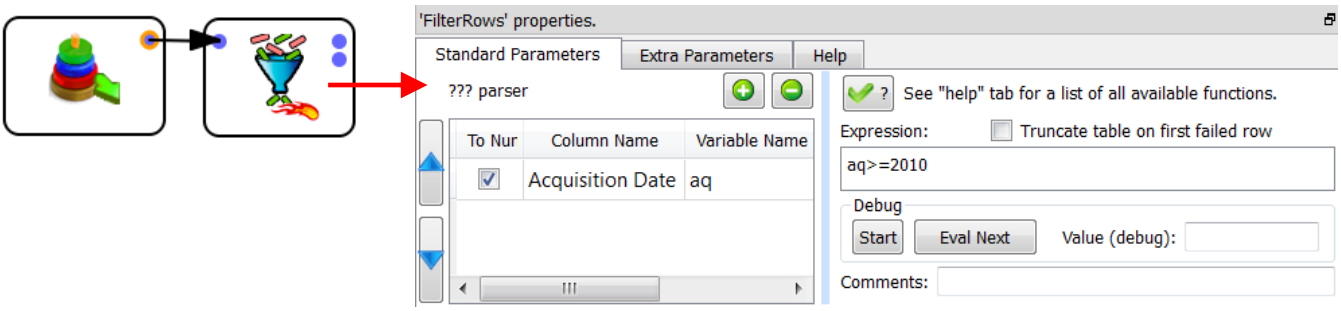
As a general "rule-of-thumb", you can follow these advices:

- Group together inside the same "column set" the columns that are used very often together inside your aggregations&data-transformations.
- Put aside in a separate "column set" the columns that seldom used (to avoid losing time reading them involuntarily because of the Microsoft Windows "Read Ahead Trick"). For example, the column named "comments" is typically not very often used inside large queries/data-transformations and should be "set aside" inside its own column set.
- If you are using SSD drives, you are lucky: SSD drives do not have any "hard drive heads" to move and so there is practically no penality when reading simultaneously from many different files. When using SSD drives, the best option is nearly always: "*Write each column is inside a different "column set" data file*". This is easy! ☺
- If you are using "Old", classical Magnetic Drives, try to reduce to the minimum the number of "column sets" data files.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

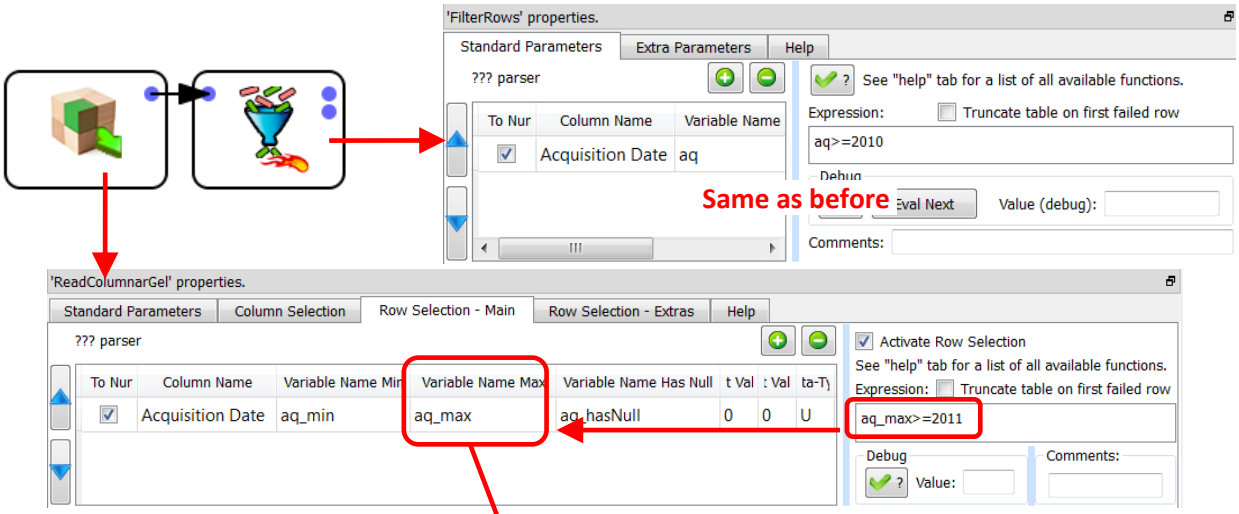
- **Read the rows for which "Acquisition Date>=2011"**

To accomplish that without the old ".gel_anatella" row-base gel file, you'll typically do the following:



This means that you'll read the whole table out of the hard drive and, therafter, "filter out" the rows that do not verify your condition (i.e. you'll remove the rows that have "Acquisition Date"<2011).

With the new Columnar Gel File, we'll have:



The Row Filter (more precisely the "Data-Block Filter") embedded inside the ColumnarGelFile Reader is using the ranges of each data-block (stored inside the ".cgel_anatella") to filter-out the non-required data-bock. To remind you, in the above example, we had:

Range on Name	Range on Revenue	Range on Acquisition Date	Offset Information	
[Daniel ; Sabrina]	[10 ; 30]	[2000 ; 2002]	0;4;8;12	data-block 1
[Bob ; Tony]	[20 ; 50]	[2003 ; 2009]	12;16;20;24	data-block 2
[Fred ; Stark]	[5 ; 55]	[2010 ; 2014]	24;27;30;33	data-block 3

The only data-block that passes successfully the test is the "**data-block 3**" (because, for the other data blocks, we have "*2002>=2011 is false*" and "*2009>=2011 is false*"): This is the only data-block that is actually extracted from the Hard Drive.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The content of the "**data-block 3**" is:

Data-block 3

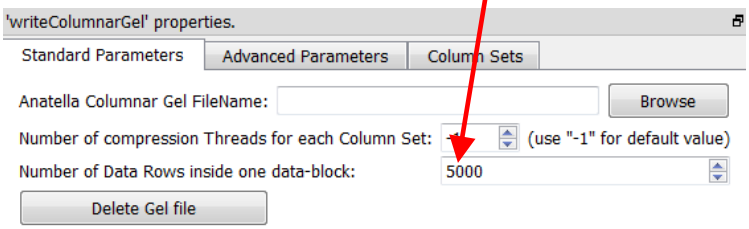
Name	Revenue	Acquisition Date
Stark	55	2014
Fred	15	2010
Freeman	5	2011

The FilterRows Action following the ColumnarGelFile Reader is still required to remove the row containing "Fred". We obtain as final output:

Name	Revenue	Acquisition Date
Stark	55	2014
Freeman	5	2011

The "Data-Block Filter" embedded inside the ColumnarGelFile Reader allows you to avoid extracting from the Hard Drive all the non-required data-blocks, to reduce to the minimum the I/O performed by Anatella, still increasing speed.

And finally, a small note about the parameter "*Number of Rows inside one data-block*":



The optimal value of this parameter is again a compromise:

- A large value means a large RAM memory consumption (both at the time of saving the file on the Hard Drive and at the time of reading the file from the Hard Drive).

  (The large RAM consumption originates from the fact that the writeColumnarGel Action must use large data-blocks in RAM memory before "flushing" them column-by-column on the hard drive.)

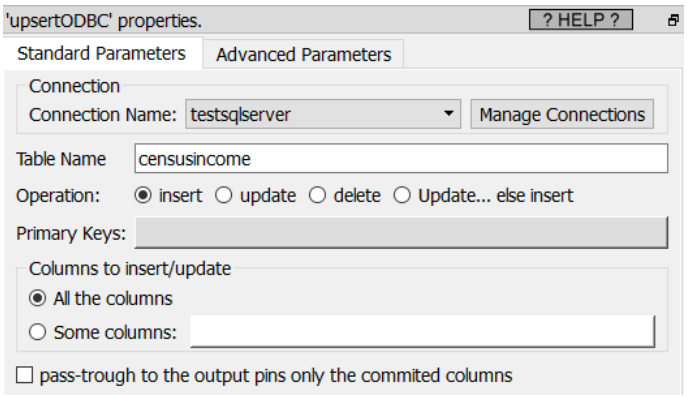
- A small value means that the "Data-Block Filter" embedded inside the ColumnarGelFile Reader won't be very efficient: One test of the data-block-filter will only allow to "skip" the extraction of a small number of rows (and thus a large number of tests will be required to find the exact rows to extract).

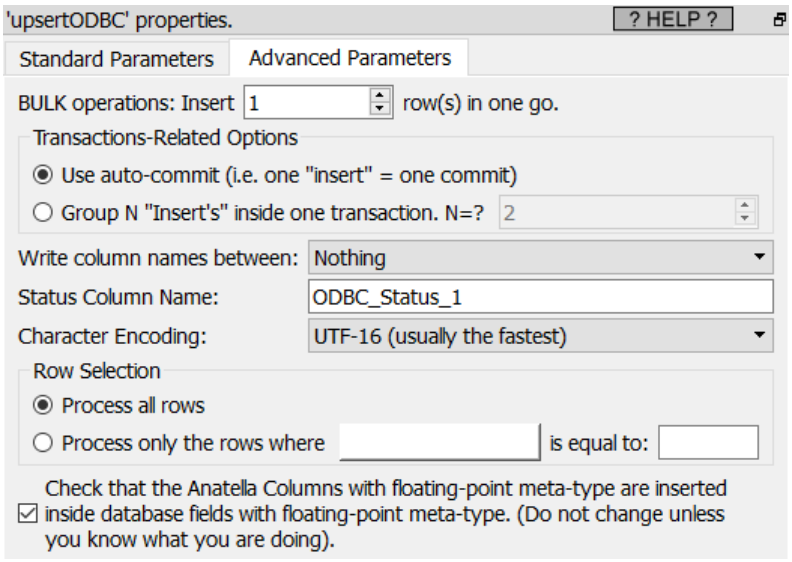
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.27.4. Generic ODBC Writer

Icon:

Property window:





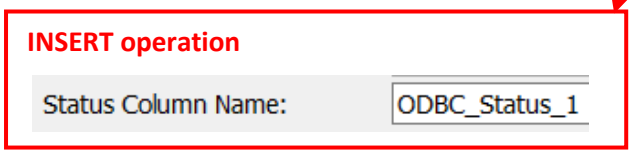
Short description:
Insert/update/Delete some rows inside a relational database using the ODBC driver.

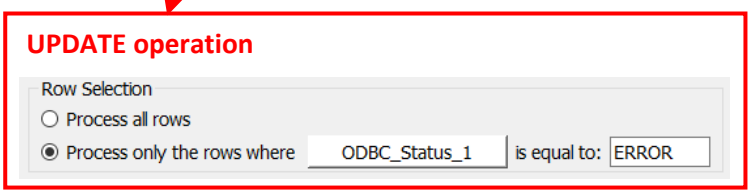
Long Description:
This action Insert, Update or Delete some rows inside a relational database using the ODBC driver.

Before executing the

upsertODBC Action, you must first define an ODBC connection: See the section 5.1.6. to know how to create an ODBC connection from Anatella to your database.

Here is how to do a classical "Upsert" (Insert otherwise Update) operation in an efficient way:





This Anatella-Graph first attempts to insert some new rows into a table inside a relational database. Some of these rows already exist inside our relational database. These rows will trigger an error at each attempt of insertion. These "insertion" errors are signaled inside the output column "ODBC_Status_1": i.e. the column "ODBC_Status_1" contains "ERROR" for all failed insertion. Thereafter, there is a second "ODBC writer" that takes as input all the rows with an insertion error and performs an "update" on them.

The "Insert", "Update" and "Delete" operations are sent "in batch" to the database driver. The size of these batches (the number of "Insert", "Update" and "Delete" operations in a batch) is the parameter is named "*BULK operations*". Large-batch-size usually means higher performances but, unfortunately,

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

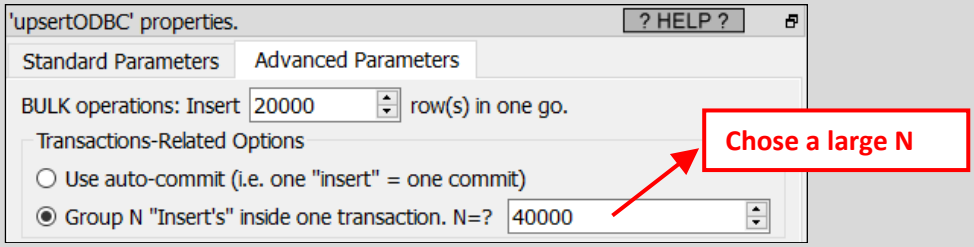
not all the databases are working properly with large-batch-size. The safest option when working with low-grade database drivers is to set the "*BULK operations*" parameter to 1.

On most databases, using the "auto-commit" option is the fastest option because it means that the database does not have to manage all the ACID properties related to the processing of TRANSACTIONS. This is not always true: When the "target" table (i.e. the table to modify) has an index structure, it might be faster to use TRANSACTIONS: See the next section about this subject.

Oddly enough, the most efficient way to INSERT rows inside a "DB2" database is to use large TRANSACTIONS.

Indeed, DB2 is forcing a SYNCHRONIZE operation (i.e. an operation where the content of the database in RAM is "flushed out" to the hard drive, to prevent any data loss in case of electrical power cut) at the end of each and every TRANSACTION. It means that, if you are using "auto-commit", DB2 peforms a SYNCHRONIZE operation for every row that is INSERT'ed inside the database: This dramatically slows down everything.

Since all SYNCHRONIZE operations are extremely slow, it means that, to get the highest speed when using DB2, the best option is to reduce to the minimum the number of TRANSACTIONS. Thus, the fastest option for DB2 is usually:



5.27.4.1. Inserting Rows inside a Table with an INDEX

All databases are using an INDEX structure that allows them to quickly find a particular cell in a large table containing billions of cells: I suggest you to read the section 10.10.1. about this subject if you didn't do it yet.

One very annoying side-effect of using an INDEX structure is that the "insert" type of operations are very slow. Indeed, each time you insert a new row inside a table, you must update the INDEX structure to reflect the presence of this new row: This is *very* slow! There are several "tricks" that you can use to avoid losing time (because always updating this INDEX structure after each insertion of a new row consume a large amount of time): i.e. To add many rows inside an already existing Table T at high speed, you can use different solutions:

- "Fast INSERT" Solution 1
  The solution 1 is composed of 3 steps:
  - Delete all INDEX structures built on top of table T (i.e. run many "DROP INDEX" SQL commands)
  - Insert your new rows
  - Re-create all the INDEX structures built on top of table T (i.e. run many "CREATE INDEX" SQL commands)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Most of the time, this first solution is impractical and it's almost never used.

Indeed, when using the solution 1, there are no "INDEX" on your table anymore during a brief amount of time (i.e. during the time required to insert all your rows): ..And this is **very dangerous** because all the processes that are attempting to run a query on the Table T will most certainly fail (because these processes will "time out" because they will run very slowly because there are no "INDEX" currently available). Furthermore, the time required to re-create all the INDEXes is usually so large that it completely nullifies the speed gain obtained by inserting the row at a slightly higher speed.

- "Fast INSERT" Solution 2

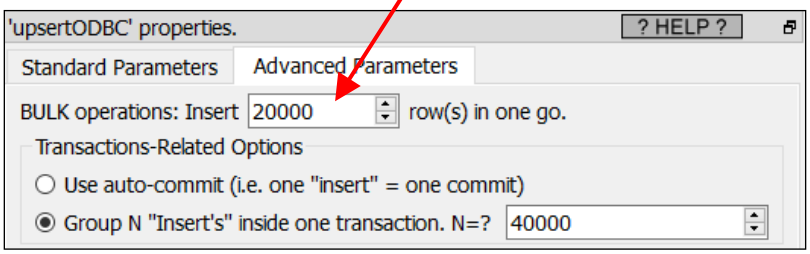
  The solution 2 is also composed of 3 steps:
  - Insert all your new rows inside an intermediary table that has no INDEX
  - Run a "INSERT INTO <Table T> SELECT * FROM <Intermediary Table>" SQL command. If the database engine is correctly coded, it should see that it only need to update one time the INDEX structure, once all the rows from the intermediary table have be added to the final Table T.
  - Empty the intermediary table (i.e. run a "TRUNCATE TABLE" SQL command)

  When you directly insert news rows one-by-one inside the final Table T (in a simple, straightforward way), the database engine must update the INDEX structure thousands of times (i.e. there is one update for each inserted row). Using the "solution 2" allows to reduce the number of updates of the INDEX-structure to only one, leading to a large gain in time.

  Unfortunately, the solutions 2 also involves copying a large amount of data (from the intermediary table to the final Table T). This large "copy" operation might consume so much time that, at the end, the final gain in time might be null.

- "Fast INSERT" Solution 3

  When Anatella communicates with a database, it sends to the database a large buffer that contains many INSERT expressions to be executed by the database engine. The quantity of INSERT operations that are sent inside the same buffer to the database engine is defined using the parameter named "BULK Operation" here:



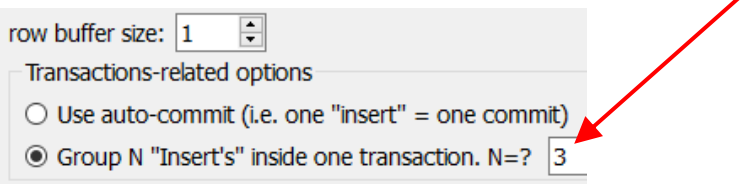
  Most (Java) ETL engines are not able to use any BULK Operations: i.e. They are sening to the database engine only one only "INSERT" SQL command at-a-time. Doing so, leads again to a very low insertion speed (because, for each inserted row, there is an update of the INDEX-structure). More precisely, if there are N rows to "insert", the ETL engine will send N "text buffer" to the database engine, (each "text buffer" containing one "INSERT" SQL command) and the database engine might then run N "INDEX-structure-update".

  A clever database engine will see that this "buffer" contains many INSERT statements (several thousands) and only update the INDEX-structure once the complete set of INSERT statements has been completely processed. Thus, instead of running an "INDEX-structure-update" for each inserted row, we run an "INDEX-structure-update" every 10 thousand rows (i.e. at the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

end of each buffer). This leads to large time savings (but, unfortunately, not all databases are "clever").

- "Fast INSERT" Solution 4
  Anatella is also able to include inside the same transaction many different "INSERT" statements. For example, if you want to "group together" 3 "INSERT" statements together inside the same transaction, you can use the following parameters insde Anatella:



A clever database engine will see that each transaction contains many INSERT statements (sometime several thousands) and only update the INDEX-structure once the complete set of INSERT statements has been completely processed (i.e. when the transaction ends with the "COMMIT" statement). Thus, instead of running an "INDEX-structure-update" for each inserted row, we run an "INDEX-structure-update" every 10 thousand rows (i.e. at the end of each transaction). This usually leads to large time savings.

Unfortunately, running any kind of transactions inside a database always incurs a large overhead (because of the mechanims used to guarantee the ACID properties of the transactions, that are very time-consuming). This processing overhead dramatically slows down the "INSERT" speed. So, the time gained by reducing the number of "INDEX-structure-update" might anyway finally be lost again because of the overhead of the transaction mechanism.

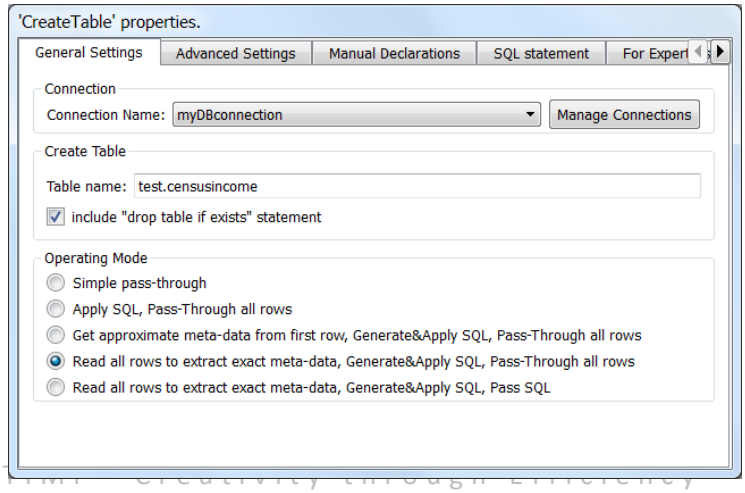
Which "Fast INSERT" solution (1-4) is best for which database?

Oracle: Solution 4 (or 3) is usually the best
MS-SQLServer: Solution 3 (or 4) is usually the best
Teradata: Solution 2 is usually the best (if you use it at the same time as the special TeradataWriter Action)
DB2: Solution 4 is usually the best (the DB2 database does not exactly support transactions but the "commit" keyword is usefull to improve the "INSERT" speed for still other reasons)

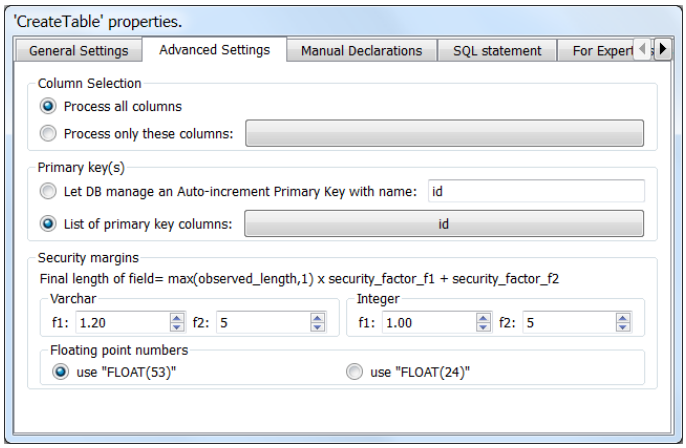
5.27.5. Create Table

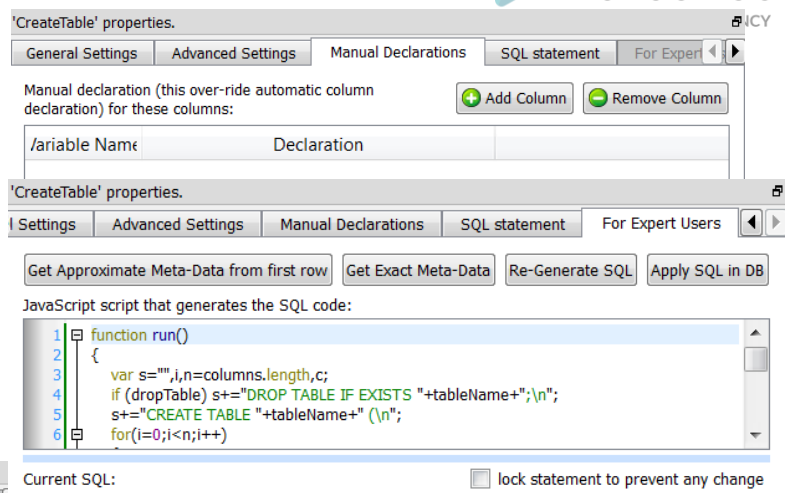
Icon:

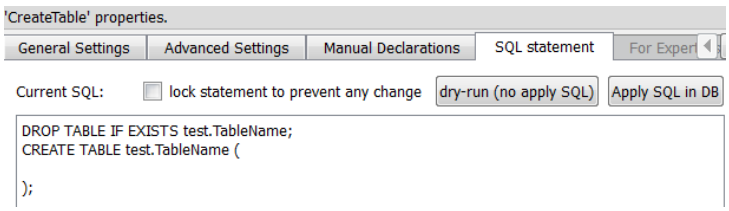
Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu







Short description:
Generate a CREATE TABLE statement to run inside a SQL database.

Long Description:
To generate the CREATE TABLE statement, we need to know:

- The type of each column: VARCHAR (used for Strings), FLOAT (used for Floating-Point numbers), DEC (used for integer numbers)
- The length of each field. For example: the maximum number of characters inside a column of the String type.

Before executing the CreateTable Action, you must first define an ODBC connection: See the section 5.1.6. to know how to create an ODBC connection from Anatella to your database.

The "Operating mode" parameter

There are basically 4 strategies to obtain these 2 informations (i.e. the Type & length of each field).
The "Operating mode" parameter allows you to choose which strategy Anatella will use:

1. "Apply SQL, Pass-Through all rows" option: This does not create any new CREATE TABLE statement. It re-uses an "old" one (that was most certainly "written by hand").

2. "Get approximate meta-data from the first row, Generate&Apply SQL, Pass-Through all rows" option: This reads only the first row of the input table. At this point, we have an approximate estimation of the type and length of each field.
   For example, it can happen that a field contains only integer numbers but is stored inside Anatella using the Unknown/String data-type. In such situation, the CreateTable Action will NOT detect the correct type to use inside the SQL statement (i.e. it will use the wrong "VARCHAR" type instead of the correct "DEC" type).

3. "Read all rows to extract exact meta-data, Generate&Apply SQL, Pass-Through all rows" option: This reads the whole input table: We then have the exact type and length of each field.
   Let's go back to the above/previous example: In this "exact mode", the CreateTable Action will correctly use the DEC type inside the SQL statement.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This option is very time consuming because, it stores inside a temporary HD Cache the whole input table (while analysing it), then create & apply the SQL statement, and then re-reads the temporary HD Cache to Pass-through all the rows.

4. "Read all rows to extract exact meta-data, Generate&Apply SQL, Pass SQL" option: This option returns the same SQL statement than the previous option but it's not so time consuming because it does NOT need to create any HD Cache.

The Security Factors Parameters

The CreateTable Action computes the (approximate or exact) length of each field by observing the data coming from the **current** input table. For example: Let's assume that our input table has a field named "ADDRESS" whose maximum length is 15. It's quite dangerous to declare "ADDRESS" as a "VARCHAR(15)" because it can happen that, on another input table, we have a longer "ADDRESS" field than 15 characters. When we try to INSERT into our SQL table an "ADDRESS" longer than 15 characters, we will have (at best) some truncation errors. To avoid truncation errors, we need to apply some "security margins": i.e. We'll define the "ADDRESS" field slightly larger than 15.

The formulae used to compute the length of a field is:

Final_Length_of_Field = max(Max_Observed_Length, 1) x security_factor_f1 + security_factor_f2

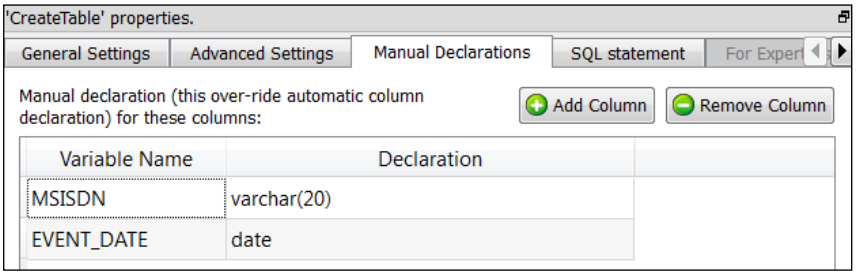
The **security_factor_f1** and **security_factor_f2** are two user-parameters of the CreateTable Action. We have, by default:

	security_factor_f1	security_factor_f2
For VARCHAR type (i.e. for strings)	1.2	10
For DEC type (i.e. for integer	1	5

The Manual Declarations Parameters

By default, any columns containing integer numbers will be declared using the "DEC()" type.
By default, any columns containing strings will be declared using the "VARCHAR()" type.

You can override the default behavior of The CreateTable Action using the Manual Declarations Parameters. For example:



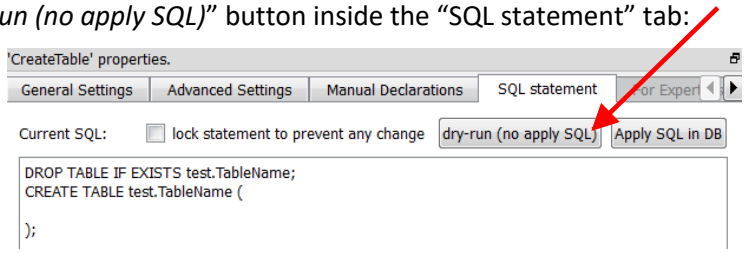
This will declare:

- the "MSISDN" column as a "VARCHAR(20)" column (disregarding the fact that it only contains integer numbers and should therefore normally declared as "DEC(20)").
- the "EVENT_DATE" column as a "DATE" column (disregarding the fact that it only contains strings and should therefore normally declared as "VARCHAR(10)").

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Generating a "Draft" SQL statement, editing it and running it.

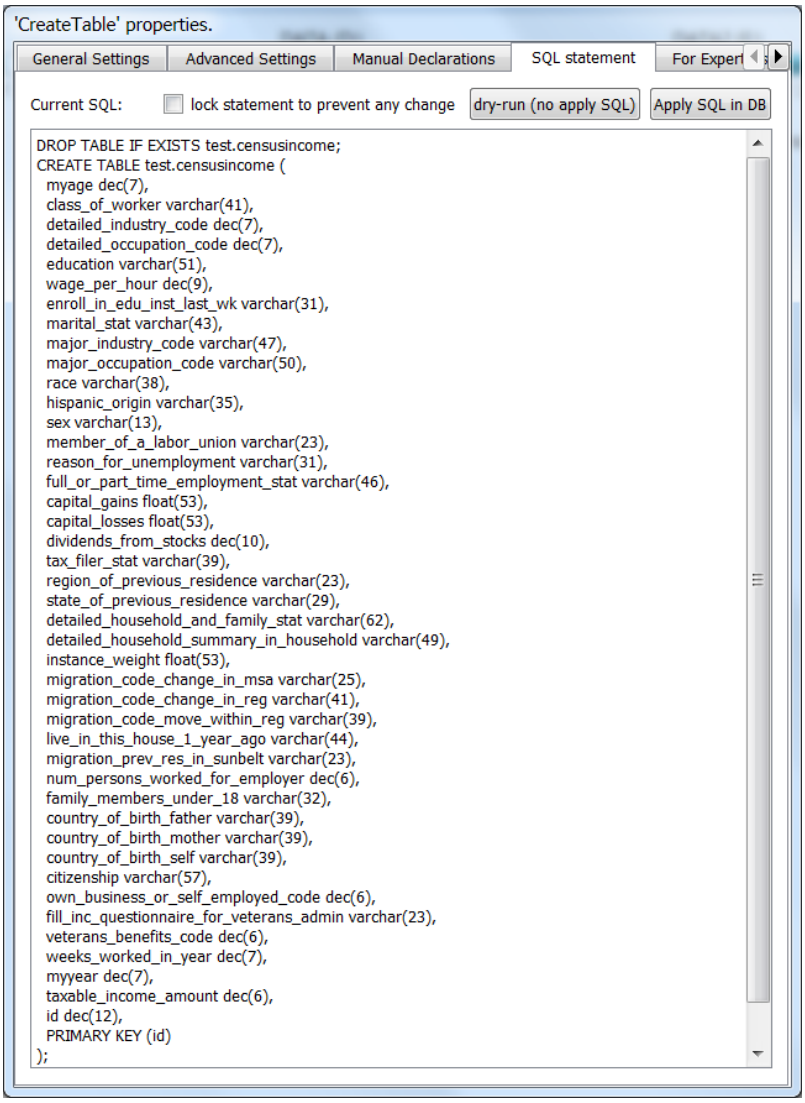
To generate a "Draft" SQL statement, select as "Operating Mode" either:
"*Get approximate meta-data from the first row, Generate&Apply SQL, Pass-Through all rows*"
or "*Read all rows to extract exact meta-data, Generate&Apply SQL, Pass-Through all rows*"
and click the "*dry-run (no apply SQL)*" button inside the "SQL statement" tab:



If you selected as "Operating Mode" the option "*Read all rows*", Anatella might "freeze" for a long time because it needs to "*Read all rows*" of the input table and this can take a large amount of time.

Tip: Use the Sampling Action to prevent Anatella from "freezing" for a long time.

Once the analysis of the input table is finished, Anatella will propose a "draft" SQL statement: For example:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

At this point, you can check the auto-generated SQL statement and change:

- The Manual Declarations Parameters.
- The Security Factors Parameters.
- The "*include 'drop table if exists' statement*" Parameter.

You will notice that the SQL statement is automatically & instantaneously updated when you change the above parameters (That's nice! ☺).

It can happen that you need to "manually" edit the SQL statement (because the above parameters are not giving you exactly the required results). You can directly click inside the SQL statement textbox and start editing the CREATE TABLE statement. You will notice that, as soon as you "manually" change something inside the SQL statement, the "*Lock statement to prevent any change*" parameter turns ON (i.e. it's checked). When the "*Lock statement*" parameter is ON, you cannot change anymore the SQL statement using any "high-level" parameters (such as the "Manual Declarations Parameters", the "Security Factors" Parameters, etc.). This is to avoid to inadvertently lose the "manual" editing by changing some security factor, for example.

Once you are happy with your CREATE TABLE statement, you can directly run it (and test it) inside your database by clicking the "*Apply SQL in DB*" button. Possible errors in your SQL statement are directly reported to you and you can directly correct them.

Expert User Mode

The "CREATE TABLE" statement generated with the CreateTable Action is 100% standard and should work with most databases.

There are however some cases when you want to make some "adjustments" to the SQL statement. A first easy solution is to manually edit the SQL statement. Another solution is to change the (Javascript) code that generates the CREATE TABLE statement. You can edit this code inside the "*For Expert Users*" tab of the CreateTable Action. To open this tab, you must be in "Expert-User-Mode". To switch to expert-user-mode: Click the button in the main toolbar of the application.

The Javascript code that generates the CREATE TABLE statement is using the following pre-defined variables:

- tableName: String variable: Self-Explanatory.
- columns: Array of Object: Each object represents one column. An object contains the following fields:
  - name: String variable: Self-Explanatory.
  - type: Char variable: the meta-type of the column.
  - declaration: String variable: the Manual Declaration (This is empty if there is no manual declaration for this variable).
- dropTable: Boolean variable: true if the "drop table" parameter is ON.
- useFloat53: Boolean variable: true if the "Float(53)" parameter is ON.
- integerF1, integerF2: Number variables: to define the security factor for the columns containing integer numbers (i.e. "DEC" type).
- varCharF1, varCharF2: Number variables: to define the security factor for the columns containing strings (i.e. "VARCHAR" type).
- autoPK: String variable: the name of the auto-increment primary key column.
- listPK: Array of Strings: the names of the primary key columns.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

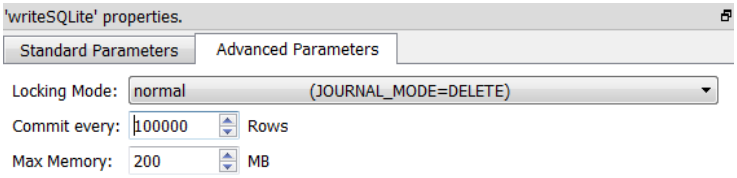
To run the Javascript code to generate a new SQL statement, click the "*Re-Generate SQL*" button.

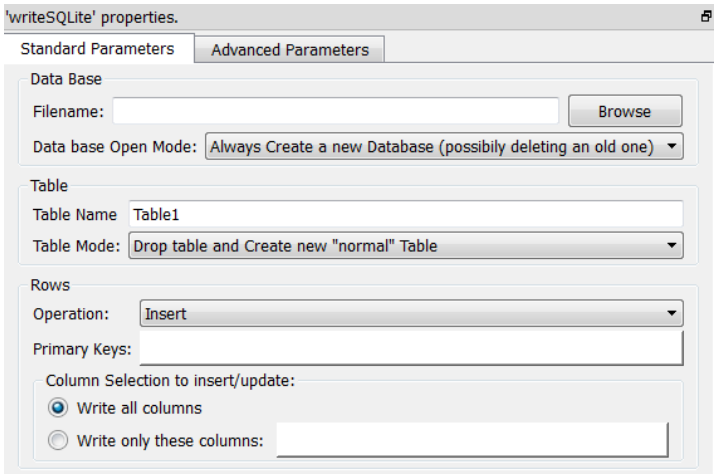
Once you are happy with your CREATE TABLE statement, you can directly run it (and test it) inside your database by clicking the "*Apply SQL in DB*" button. Possible errors in your SQL statement are directly reported to you and you can directly correct them.

5.27.6. SQLite Writer

Icon:

Property window:





Short description:
Create/Update SQLite databases.
Long Description:
Self-explanatory

It's only guaranteed to be able to **update** .sqlite files that were created by the same SQLite engine than the one included inside Anatella. On the other hand, Anatella can **read** any .sqlite file (whatever their version number).

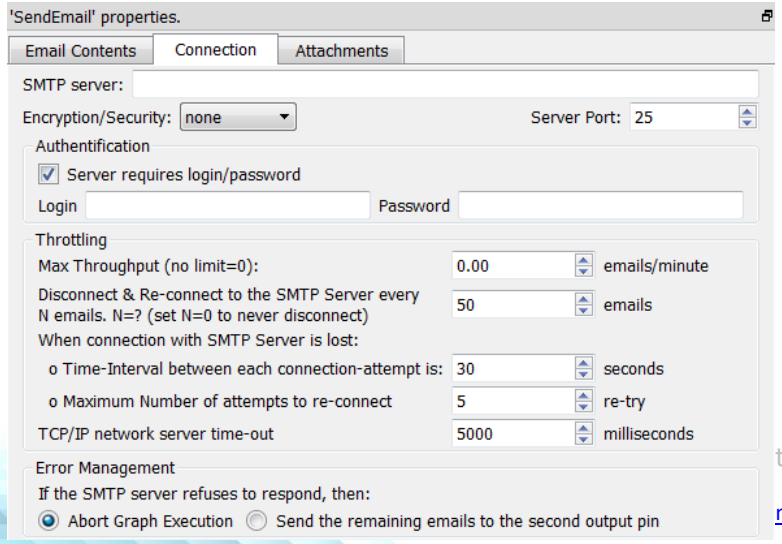
Anatella uses the SQLite engine v3.23.1.

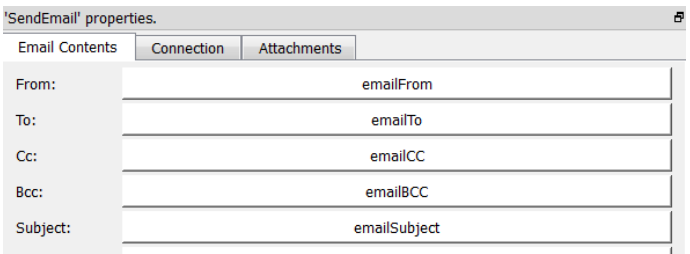
If you attempt to modify a SQLite file created using a SQLite engine that is different from the one included inside Anatella (v3.23.1.), then it's possible that you'll get the error message "*database disk image is malformed*" (It might work, but you'll have to test: i.e. Nothing is guaranteed).

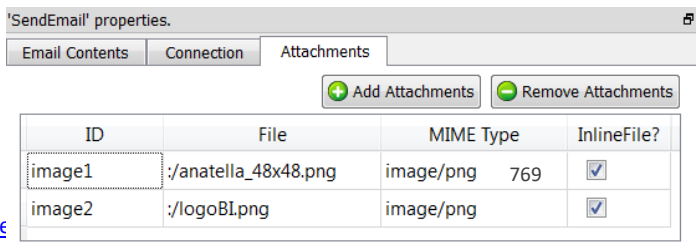
5.27.7. Send E-Mails

Icon:

Property window:







Short description:
Send E-Mails with (and without) attachements.

Long Description:
Send E-Mails with (and without) attachements.

The "To", "Cc" & "Bcc" fields may contain several e-mail addresses separated by comma's. For example, you can have as a destination (i.e. "To") value:

> bob@gmail.com, john@yahoo.com, marc@email.com

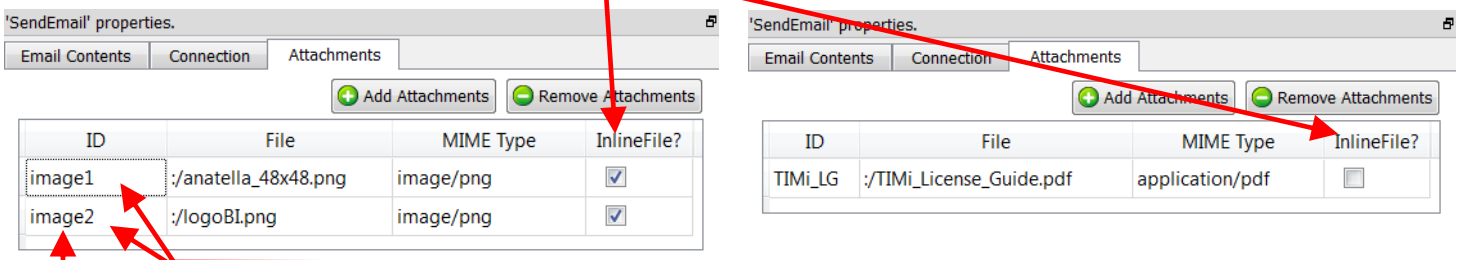
You can also define "labels" for each email: For example:

> my friend bob <bob@gmail.com>, my cousin john <john@yahoo.com>, my brother marc< marc@email.com>

There are, basically, 2 types of attachments to emails:

1. "*Normal attachments*" that you can save in your hard drive for consultation. This typically include PDF files, ZIP files, etc.
2. "*In-line files*": these are displayed inside the body of the html-message. These are typically a JPEG or PNG banner that appears inside the text of the email.

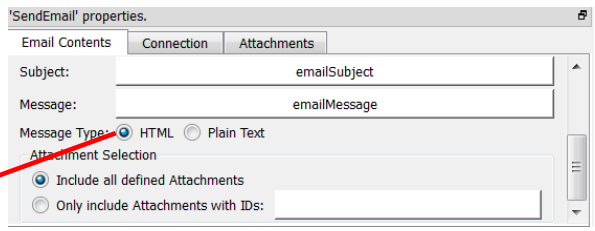
You can select either of the 2 types here:



If an attachment is of the type "In-line File", then you can use it inside the text of the email. Here is an example of html-email-message that uses the above 2 images defined here as "in-line file":

These image ID's must match

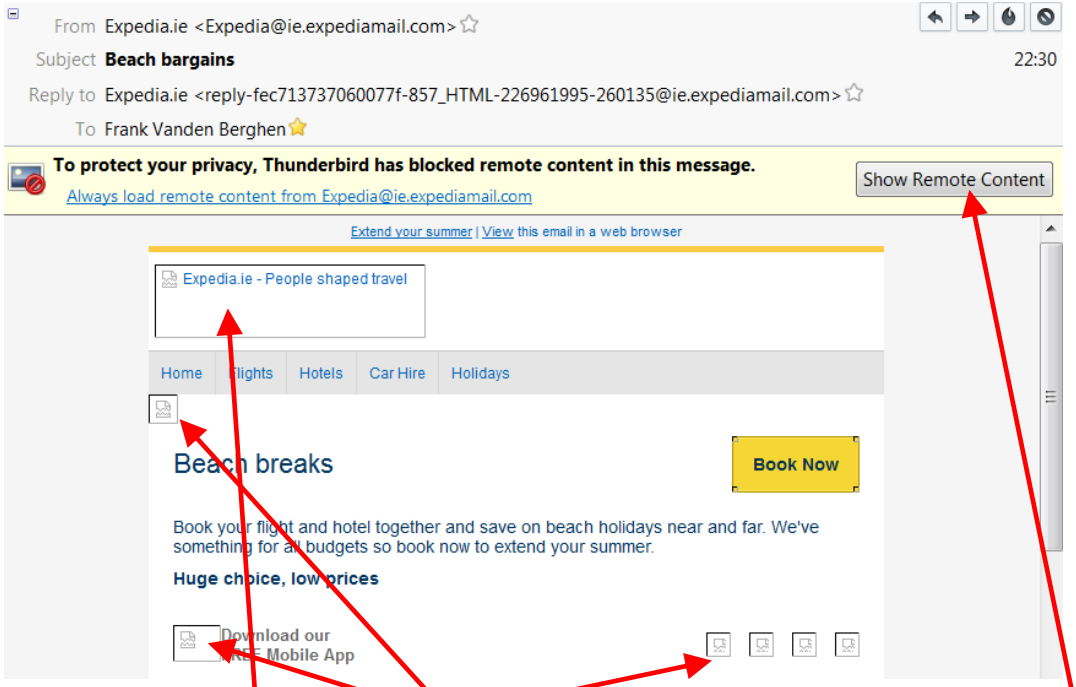
```
<h1> Hello! </h1>
<h2> This is the first image </h2>
<img src='cid:image1'/>
<h2> This is the second image </h2>
<img src='cid:image2'/>
```



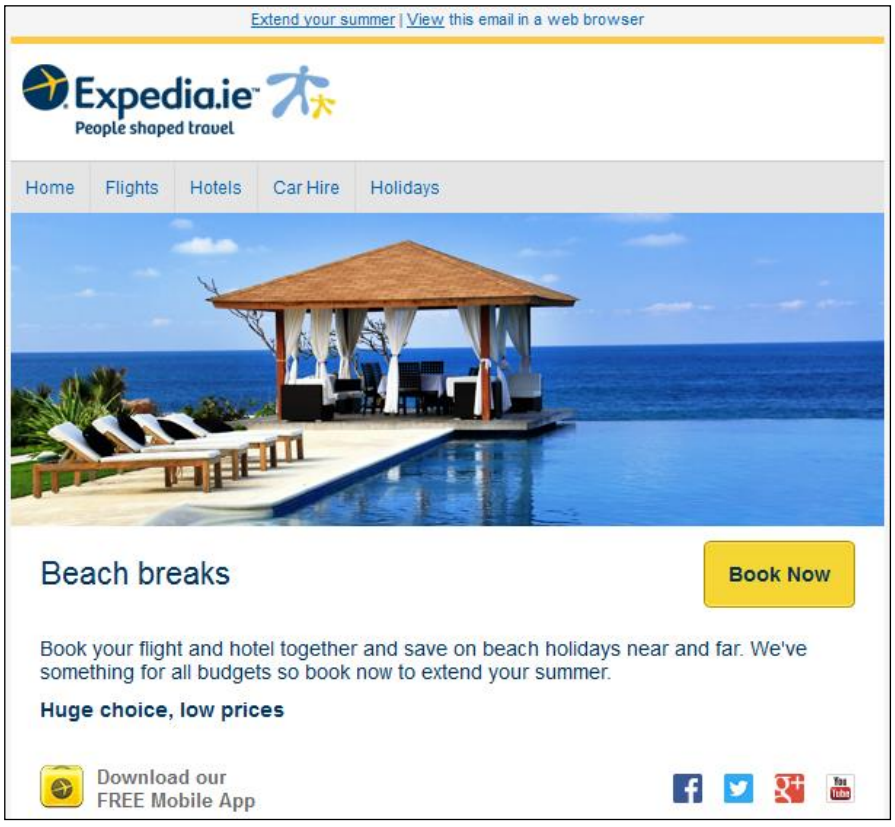
You need to select "HTML" message to able able to use images defined as "in-line" files.

When you'll receive the above email inside your email client, you'll see:

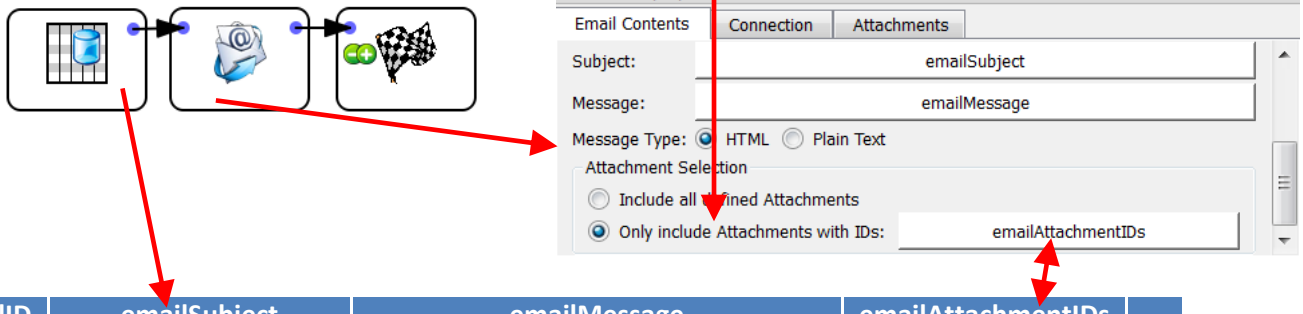
Most of the time, the images that are displayed inside e-mails are simple URL links to remote webservers. This approach has some limitation: The recipient cannot see the images, unless he clicks the "*Show remote content (such as images)*" button inside its email cient (… and personnaly, I am very reluctant to do that because it's against my privacy). For example, here is a message from "Expedia.com", inside Thunderbird:



You'll notice that the banner and the icons are "broken/not visible". The user must click here: to see the banner and the icons (Personnaly, I almost never click this button because it allows the sender of the email to "track me": it's as invasive on your privacy as cookies – maybe more). In opposition, when the images are included inside the emails as "in-line files" (instead of simple URLs), then they are always visible. Here is the same "expedia" email as above but, this time, the images are always visible (because these images are "in-line files") (…and this changes everything, don't you think?! ☺):



The only drawback about including images as "in-line files" inside emails is the time (i.e. the bandwidth) required to send the emails. Since the data from these "in-line" images are included directly inside the email, the emails are becoming "bigger" (An "in-line" image is usually around 50000 bytes. In comparison, an "url" image is only around 20 bytes. "In-line" images require thus more bandwidth when sending the emails). To reduce the size of the emails, you can use the "*Only include Attachments with IDs*" option of the "*Attachment Selection*" parameter:



emailID	emailSubject	emailMessage	emailAttachmentIDs	...

1	test anatella with image 1&2	`<h1> Hello! </h1>`
		`<h2> This is the first image </h2>`
		`<img src='cid:image1' />`
		`<h2> This is the second image </h2>`
		`<img src='cid:image2' />`
			image1, image2	...

2	test anatella with image 1	`<h2>This is the first image </h2>` `<img src='cid:image1'/>`	image1	...
3	test anatella with image 2	`<h2>This is the second image</h2>` `<img src='cid:image2'/>`	image2	...

The email with emailID=2 inside your email client:

This is the first image

The email with emailID=3 inside your email client:

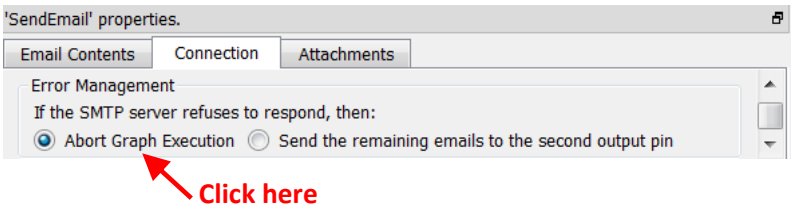
This is the second image

Using the "*Attachment Selection*" parameter, you can decide precisely which attachments are included in which email, thus reducing the required bandwith to send the emails to the minimum.

The parameters inside the "Connection" tab are self-explanatory. If your SMTP server (that is sending the emails) is very slow, increase the "TCP/IP network server time-out" parameter.

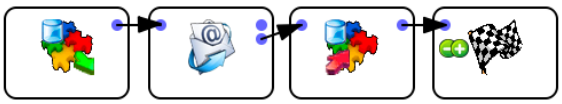
Some SMTP providers are imposing strict limits on the number of emails sent per hour (or per day). When the limit is reached, the SMTP server refuses to send any more email (the best option is still to reduce the flow of emails using the Throtteling options to avoid any interruption of service from the SMTP server). When your SMTP server is "down", you can either:

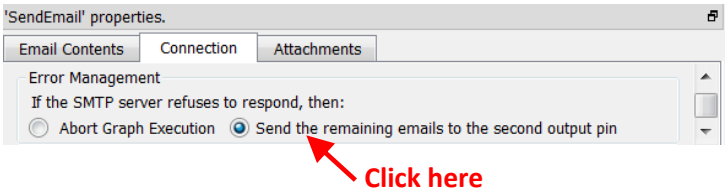
- Abort the whole Anatella-Data-transformation-Graph: For example:



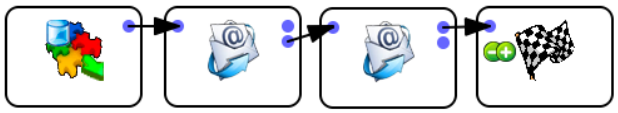
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Save the remaining unsent emails inside a .gel_anatella file to be able to send them later: Use this graph:





- Re-Route the emails that were not sent yet to another, yet active, SMTP server: Use this graph:

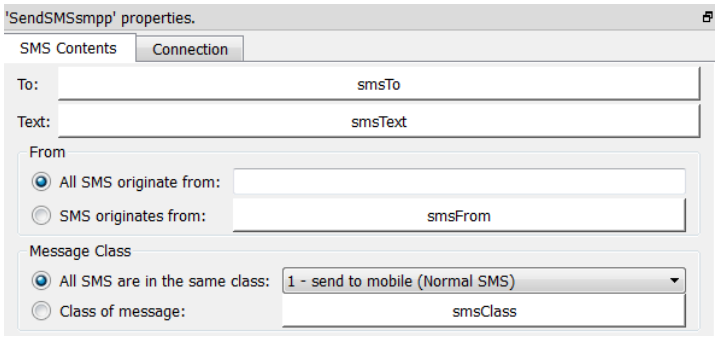


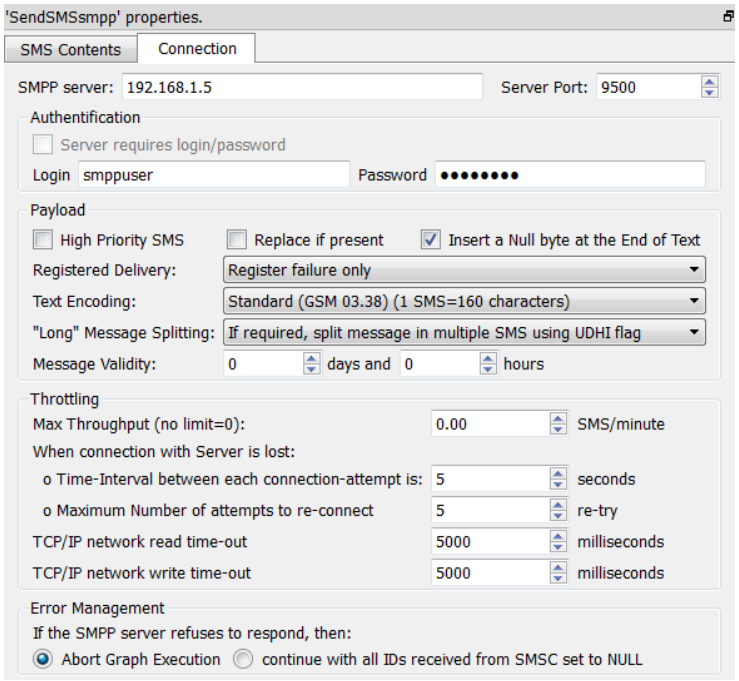
- Use a combination of the 2 above: Use this graph:

5.27.8. Send SMS with SMPP protocol (to a SMSC server)

Icon:

Property window:





Short description:
Send SMS to a SMSC (**S**hort **M**essage **S**ervice **C**enter) server using the SMPP protocol.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:

The advantages of using the SMPP protocol over the simpler HTTP protol are:

- You can define any "source/origin phone number". For example, you can send SMS coming from the phone number "MY_BRAND" (the source-phone-number can be alphabetic!). This allows for a better branding of your SMS.
- You can send FLASH SMS: These are SMS that directly "pop-up" on the phone screen instead of being stored inside the phone memory alongside with the other SMS.
- You can send SMS directly to the SIM toolkit to re-program the GSM (e.g.: To automatically configure internet settings for your network).
- You can have a delivery report: The SMSC server make its best effort to deliver all your SMS to your recipients but it can happen that some GSM phones are off and thus not contactable. In such situation, the SMSC server will continuously try to send your undelivered SMS's during a period of (typically) some days (this period is configurable using the "Message Validity" parameter).
- You have more control on the text encoding used inside the SMS's. The text encoding can be:
  - Unicode UTF-16 (1 SMS=70 characters): All characters are available.
  - GSM 03.38 (1 SMS=160 characters). A reduced set of character is available:

Basic Character Set

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ	SP	0	¡	P	¿	p
0x01	£	_	!	1	A	Q	a	q
0x02	$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	¤	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	ì	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(	8	H	X	h	x
0x09	Ç	Θ	)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	ESC	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	ß	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

Character Set Extension

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00					|			
0x01								
0x02								
0x03								
0x04		^						
0x05							€	
0x06								
0x07								
0x08			{					
0x09			}					
0x0A	FF							
0x0B		SS2						
0x0C				[				
0x0D	CR2			~				
0x0E				]				
0x0F			\					

Each character inside the "extention set" requires 2 "basic characters" for storage.

LF is a Line Feed control.

CR is a Carriage Return control, or filler.

SP is a Space character.

FF is a Page Break control. If not recognized, it shall be treated like LF.

- The SMPP protocol is slightly faster than the HTTP protocol.

If you activated the "Registered delivery" parameter (inside the "connection" tab), you can query, at any moment, the status of all your SMS's: The SMSC returns a delivery report with the status of your SMS's. This status can be: ENROUTE, DELIVERED, EXPIRED, DELETED, UNDELIVERABLE, ACCEPTED, UNKNOWN, REJECTED, QUERY REQUEST FAILED.

For each SMS sent, the

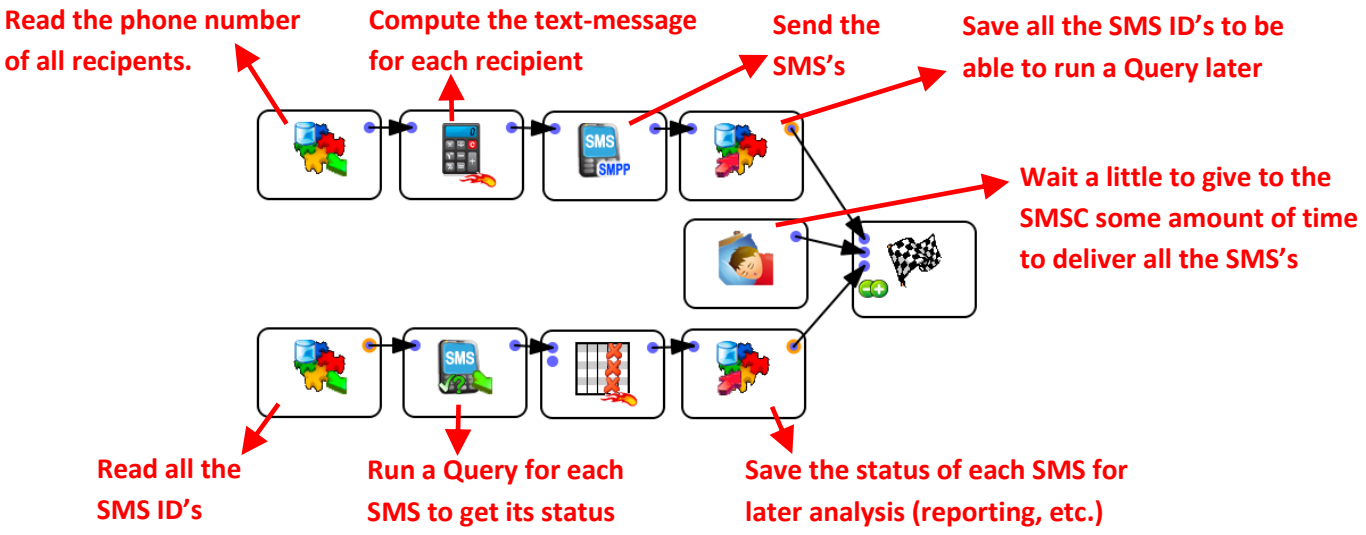
SendSMSsmpp Action returns a unique indentifier attached to the SMS.

You can thereafter use this SMS identifier with the

QuerySMS Action (see section 5.2.12) to

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

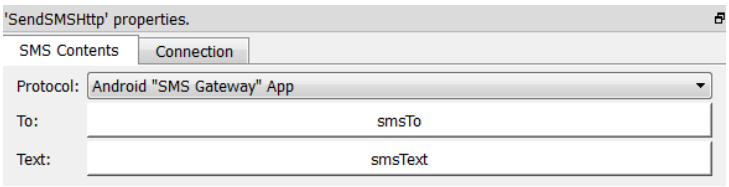
know the status of each of your SMS's. You'll typically have an Anatella-data-transformation graph such as this one:

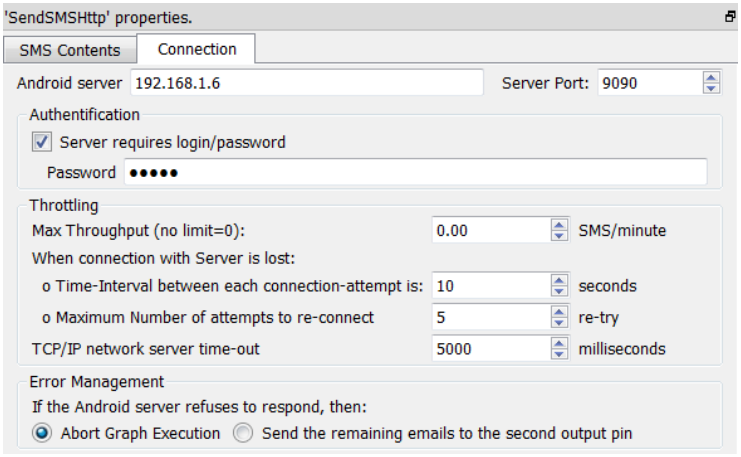


5.27.9. Send SMS with HTTP protocol (to a SMS Gateway)

Icon:

Property window:





Short description:
Send SMS to a "SMS Gateway" using the HTTP protocol.

Long Description:
Send SMS to a "SMS Gateway" using the HTTP protocol.

Anatella uses the HTTP protocol to connect to the following gateway:

- Kannel: See http://www.kannel.org/
- The "*SMS gateway*" Android Application:
  See http://play.google.com/store/apps/details?id=eu.apksoft.android.smsgateway

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- The "*SMS Gateway Ultimate*" Android Application:
  See http://play.google.com/store/apps/details?id=com.icecoldapps.smsgatewayultimate

The Android applications have the following limitations:

- No "branding" inside the "source/origin phone number".
- No delivery report.
- No FLASH SMS, No "SMS to the SIM toolkit".
- No more than 100 SMS per hour: This is a limitation imposed by the Android OS. You can remove this limitation if you install the following application on your Android phone (Warning: This app requires root access):
  http://play.google.com/store/apps/details?id=com.infinit.sms

Concerning the "delivery reports": You can configure the Kannel software to obtain delivery reports stored inside a MySQL, Oracle, PostgreSQL database: More about this subject here:
http://www.kannel.org/download/1.5.0/userguide-1.5.0/userguide.html#AEN3161

Thus, when using Kannel, you can access the delivery reports with the simple

ReadODBC Action from Anatella.

5.27.10. MS-Office-Charts Report Writer

Icon:

Property window:

'writeReport' properties.

Template: :/Anatella_report_template_demo.pptx

Output FileName: :/Anatella_AutoGenerated_Report_out.pptx Browse

Chart(s) in Template: Open output file

Tag Name	X:Pin	X:Column	Y:Pin	Y:Column	Z:Pin	Z:Column
nOcc1	0	education	0	count	-1	
Target1	0	education	0	taxable income...	-1	
nOcc2	1	marital stat	1	count	-1	

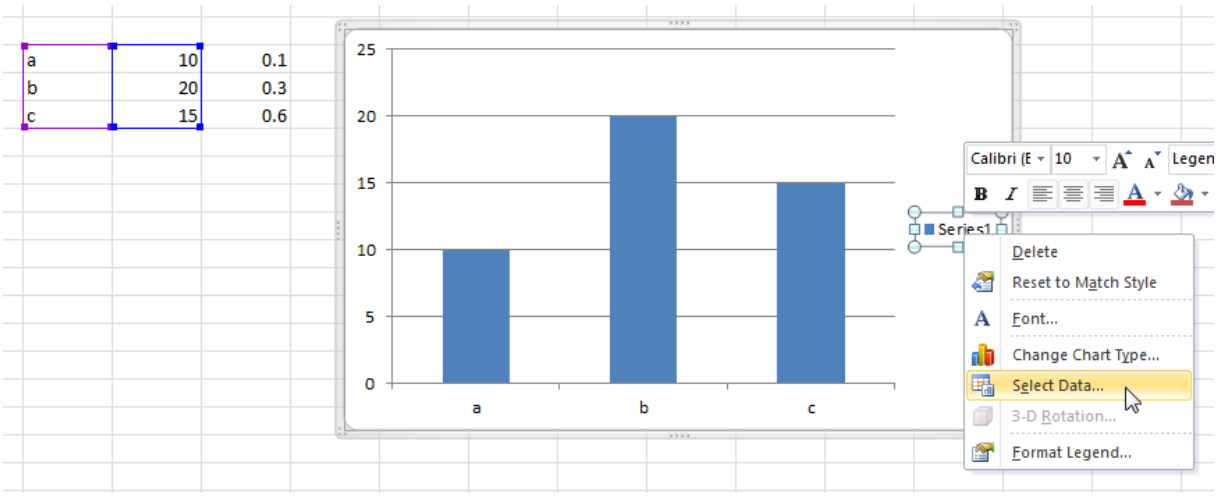
Short description:
Injects into the charts of a MS-Office document (excel, word or powerpoint) some data coming from the transformation graph.

Long Description:
This Action injects into the charts of a MS-Office document (MSExcel, MSWord or MS-Powerpoint) some data coming from the transformation graph. This allows you to automatically& instantaneously update & refresh all the charts contained into a set of "report templates" defined with MSOffice.

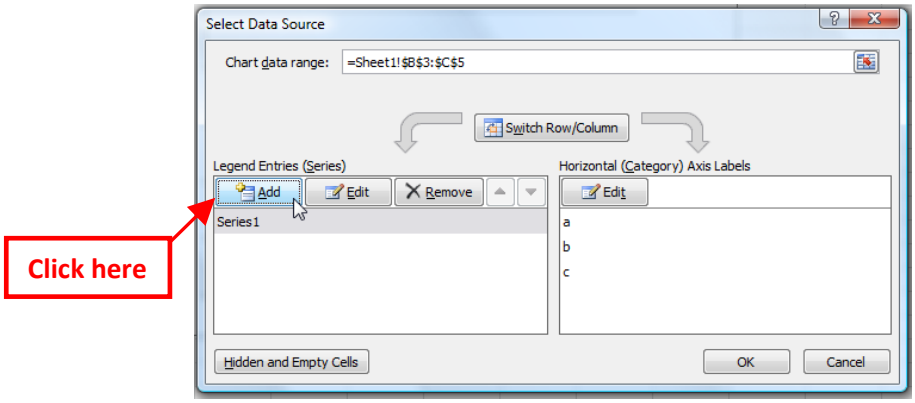
Let's give a small example. We want to update, using Anatella, two charts inside a MSPowerpoint presentation. Let's first create the two charts using MSExcel: The procedure is quite trivial for anyone trained in MSExcel:

1. The first chart contains two "series" (in MSOffice technical terms): The first serie is a *histogram* and the second serie is a *line.* We will use fake data to create the chart. These "fake data" will, anyway, be replaced by the data coming from the Anatella Graph. Let's create the first serie (histogram):

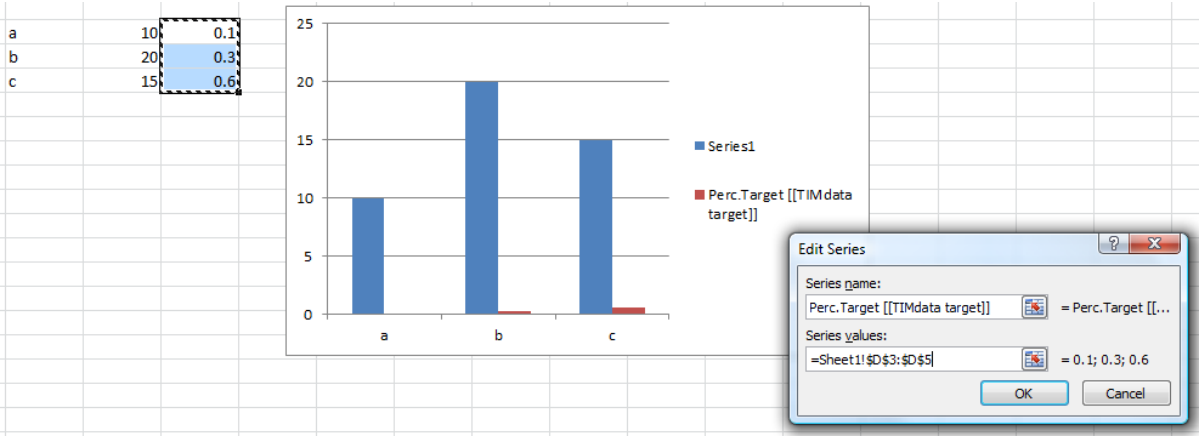
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Let's create the second serie: right-click on the "Series1" legend and select the option "Select Data" inside the dropdown-menu. A new window opens. Click the "Add" button inside the new window to add the second serie:



Configure the title and the data from the second serie:

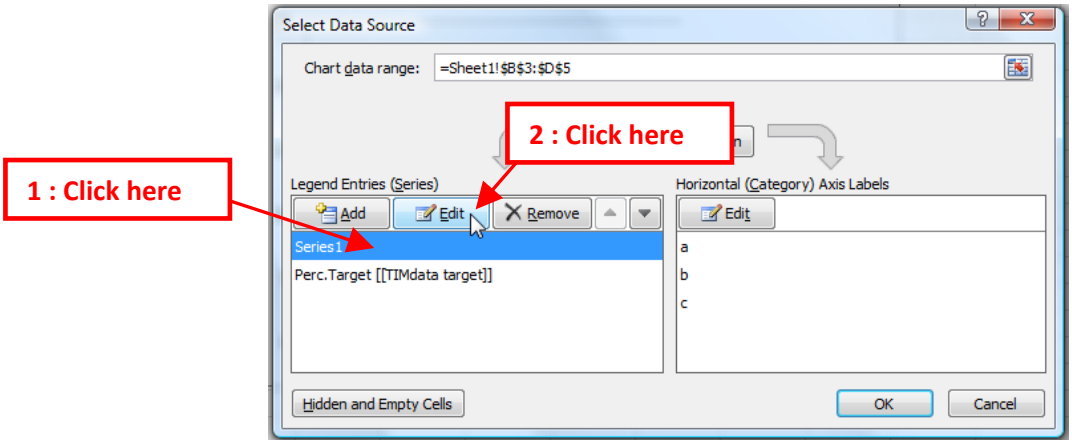


Please note the title of the second serie: "Perc.Target [[TIMdata target]]".
This title is special it contains the tag "TIMData".

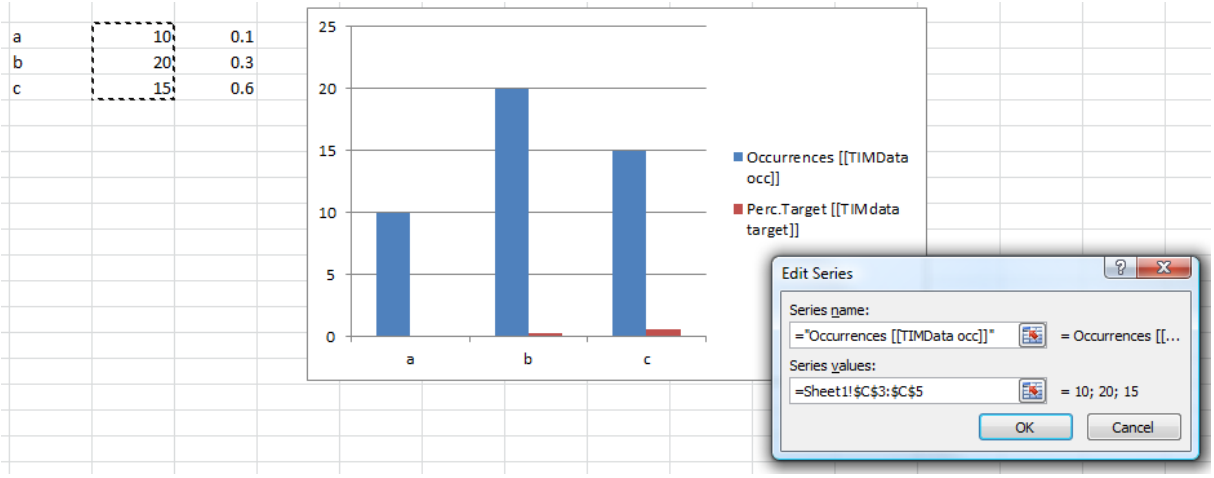
Anatella reads the MSOffice document and extract all the names of all the "TIMData" tags (in the example here above the name of the "TIMData" tag is "target"). You will thereafter use these tag

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

names to configure, for each different serie, where the data coming from the Anatella graph must be injected into the MSOffice document. Let's also change the title of the first serie: Click the "OK" button to close the "edit serie" window, select the first serie and click the "Edit Button":

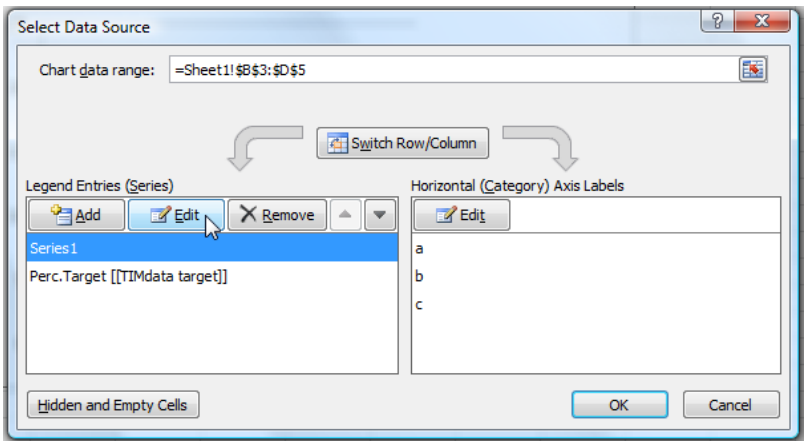


The Title of the first serie is "Occurences". The name of the TIMData tag is "occ".

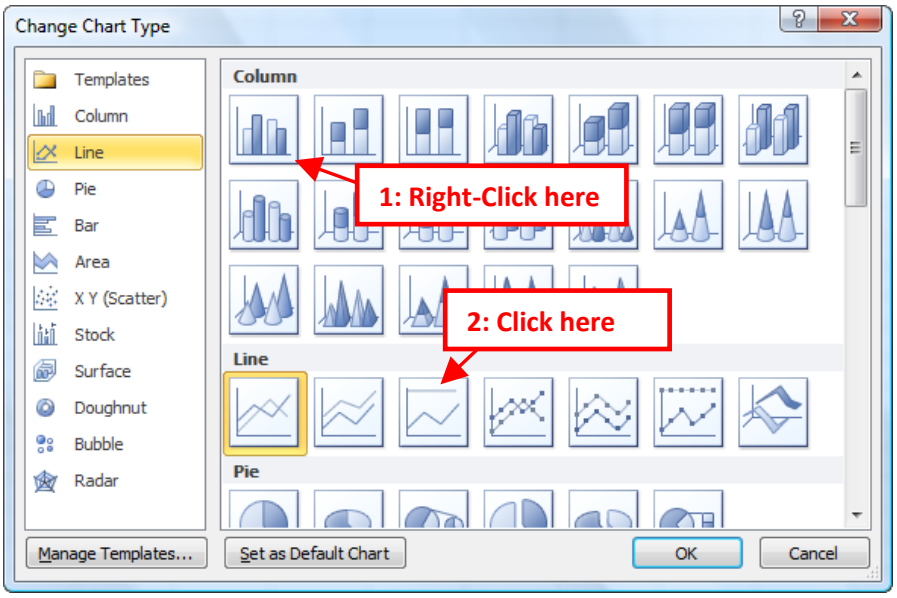


Let's beautify a little bit the chart:

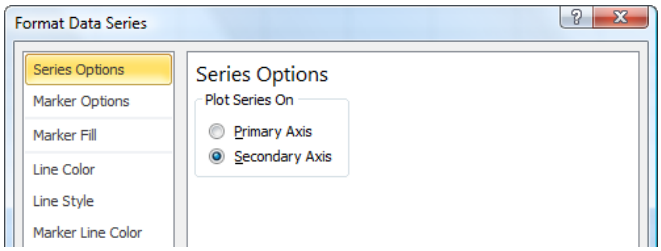
- Change the "Serie Chart type" of the second serie to "Line": Right-click on the "Perc.Target" legend and select the option "Change serie chart type" inside the dropdown-menu. A new window opens. Select the "line" type:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

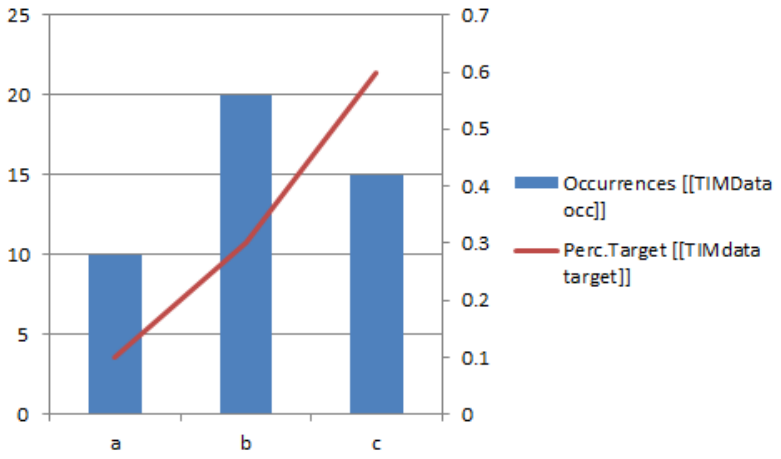


- Change the axis of the second serie to "Secondary Axis": Right-click on the "Perc.Target" legend and select the option "Format Data Series" inside the dropdown-menu. A new window opens. Select "Secondary Axis":



- You can also change the colors and the overall look of the chart in any way you want.

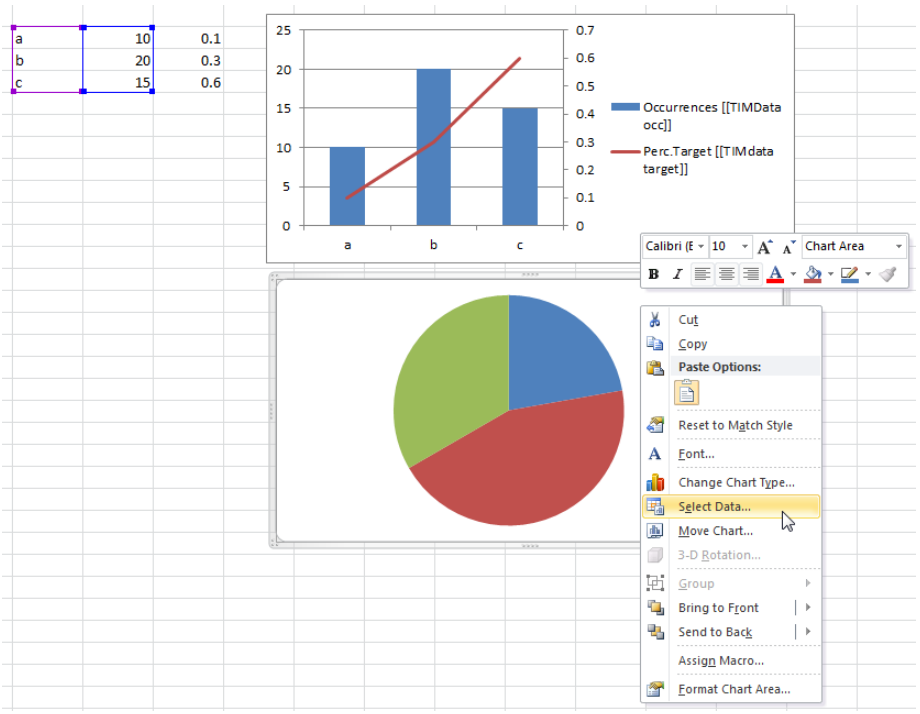
We finally have: For the first chart:



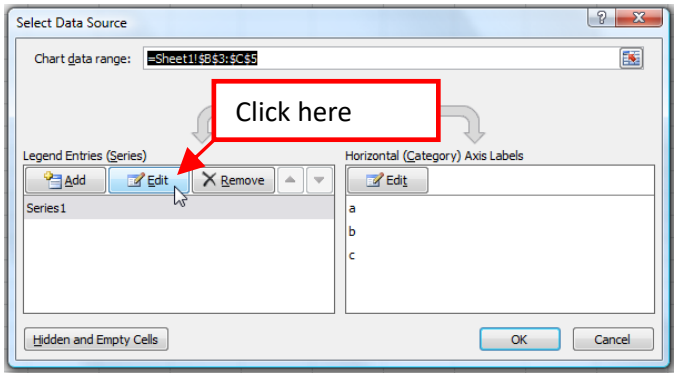
Copy-paste this chart inside your MSPowerpoint presentation.

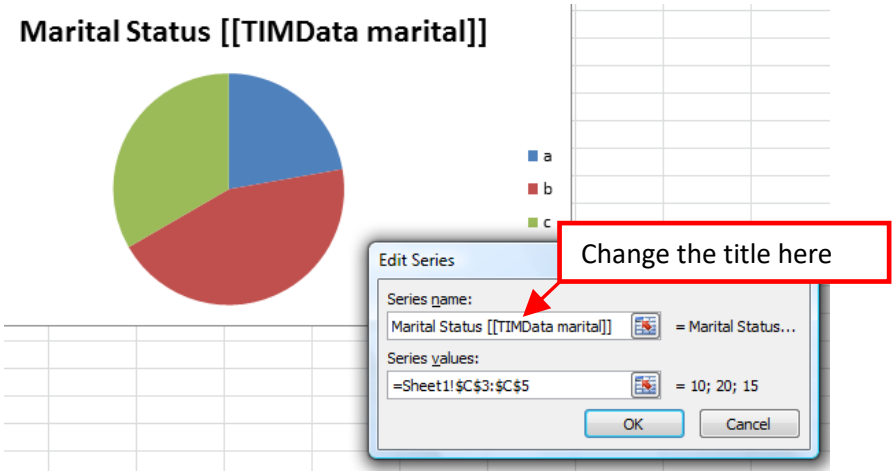
2. The second chart contains only one "serie": a pie chart. Create a simple pie-chart, using, once again, fake data:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



…and change the name of the serie to “Marital Status [[TIMData marital]]”:

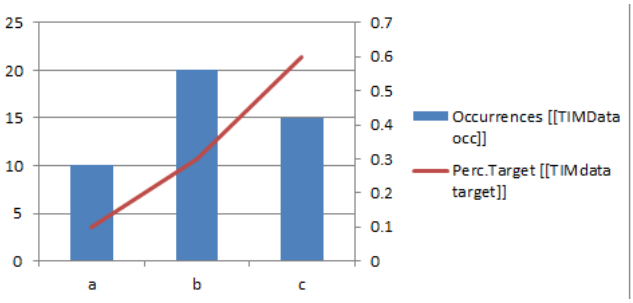


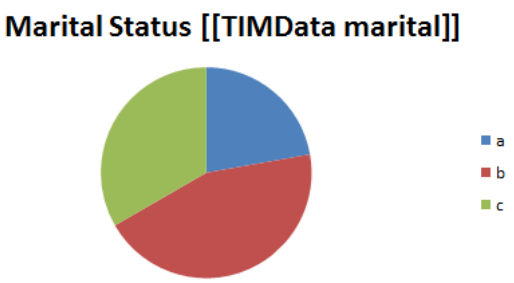


Copy-paste this second chart inside your MSPowerpoint presentation.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We have now a MSPowerpoint document that contains these 2 charts:





Our MSPowerpoint document now contains 3 series that are identified using their "TIMdata" tag:

occ
target
marital

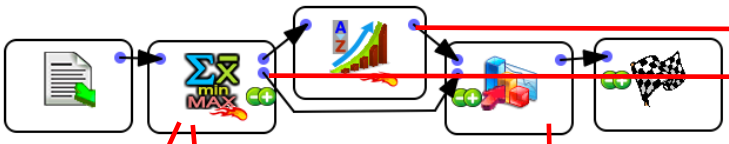
We will use Anatella to update the data displayed in these 3 series. Anatella will replace the "fake" data that we used to build the charts by data computed inside the transformation graph. We will use the first chart to illustrate:

- The distribution of the "Education" variable (using the serie named "Occ"): More precisely: We want on the Y-Axis the number of people for a specific education on the X-Axis.
- The percentage of people earning more than $50.000 per year (i.e. "wealthy" people) in function of their education (using the serie named "target"): More precisely: We want on the Y-Axis the percentage of wealthy people for a specific education on the X-Axis.

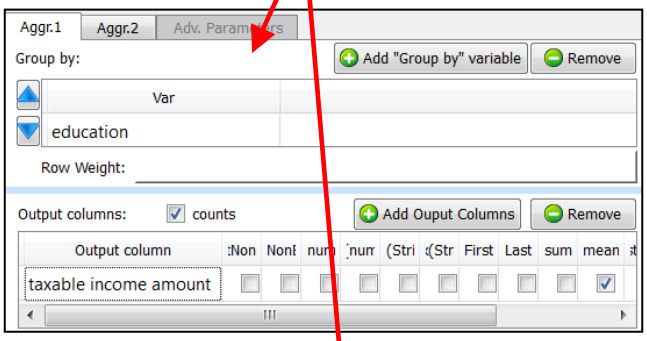
We will use the second chart (i.e. the pie chart) to illustrate the distribution of the "Marital Status" variable inside our population (using the serie named "marital"): More precisely: Each pie represents the number of people with a specific "Marital Status".

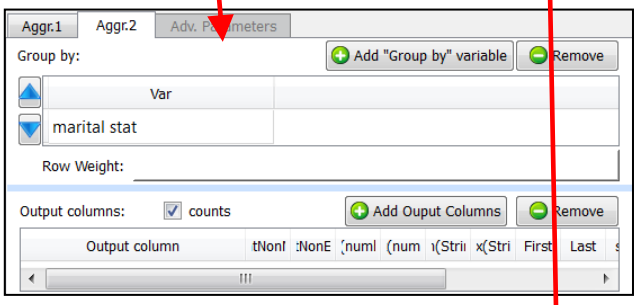
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's first generate all the data that we need:



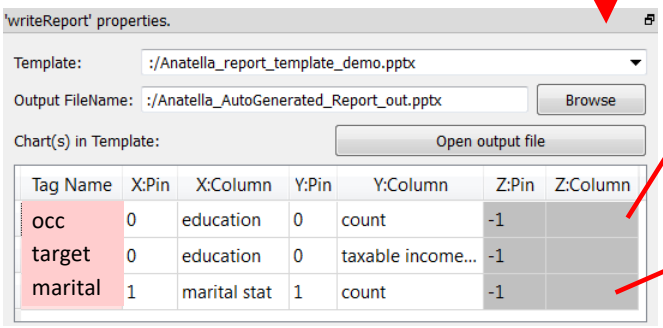
Marital stat	count
Separated	3460
Divorced	12710
Married-spouse absent	1518
Widowed	10463
Never married	86425
Married-A F spouse present	665
Married-civilian spouse present	84222





Education	count	taxable income amount_mean
Prof school degree	1793	0.540435
Doctorate degree(PhD EdD)	1263	0.52019
Masters degree	6541	0.311573
Bachelors degree(BA AB BS)	19865	0.19708
Associates degree-academic program	4363	0.09443
Associates degree-occup /vocational	5358	0.077081
Some college but no degree	27820	0.064234
High school graduate	48407	0.038817
12th grade no diploma	2126	0.015993
11th grade	6876	0.01018
7th and 8th grade	8007	0.008992
10th grade	7557	0.008204
1st 2nd 3rd or 4th grade	1799	0.007226
5th or 6th grade	3277	0.006713
9th grade	6230	0.0061
Less than 1st grade	819	0.001221

When you open your MSOffice PowerPoint template inside the WriteReport Action, Anatella automatically extracts all the "tags" of all the series inside the document. The only thing that we still needs to do is to assign to each Axis of each serie (identified by its "tag") a specific column from a specific Anatella table on a specific input pin:



The "occ" serie uses:

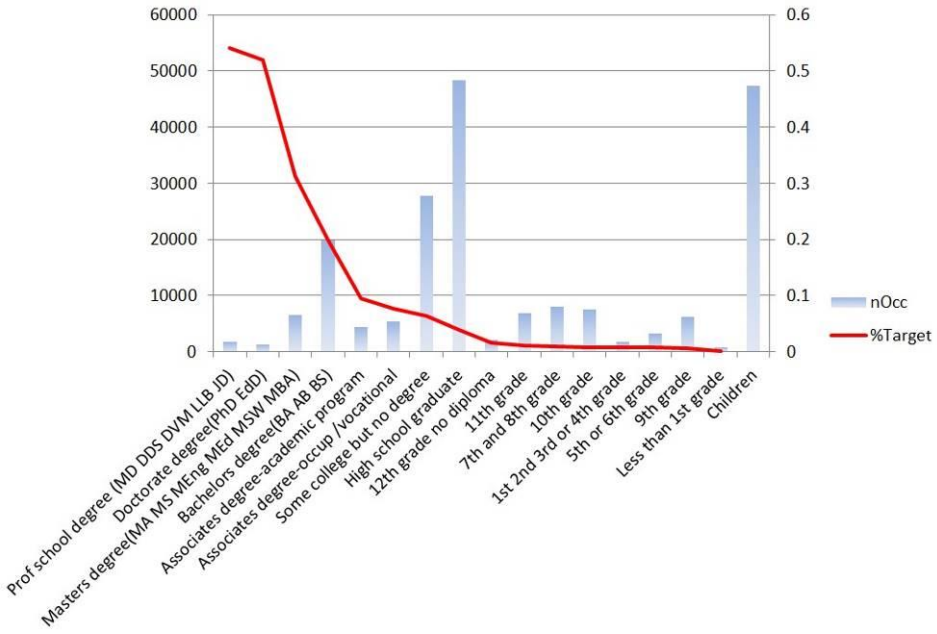
- On the X-axis: the content of the column "education" from the table on pin 0
- On the Y-axis: the content of the column "count" from the table on pin 0

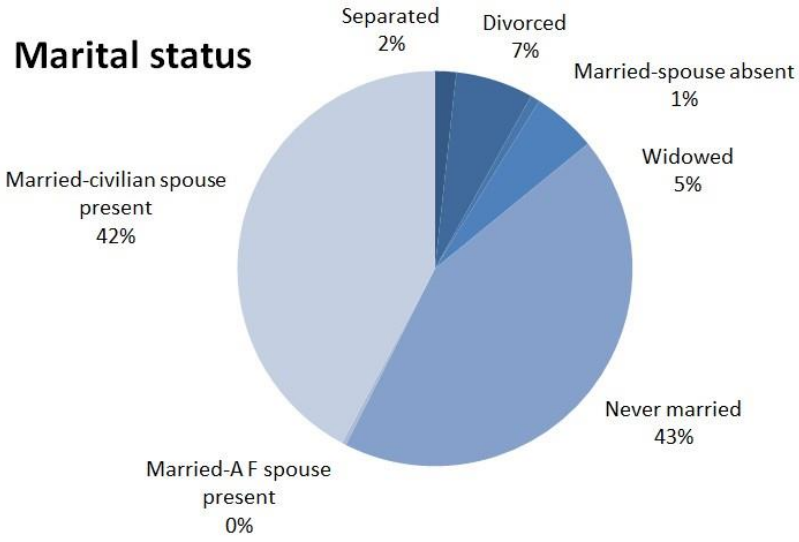
The "target" serie uses:

- On the X-axis: the content of the column "marital stat" from the table on pin 1
- On the Y-axis: the content of the column "count" from the table on pin 1

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

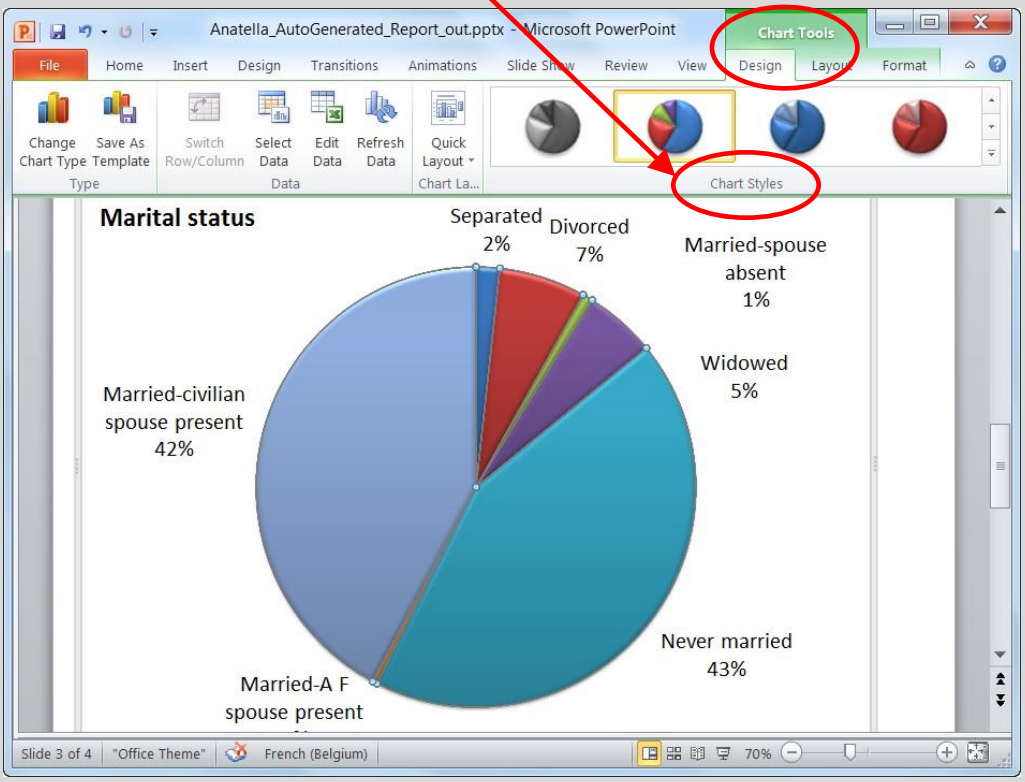
At the end, we get the 2 following charts :





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Tip: You can use the "Styling" option to quickly "beautify" your charts:



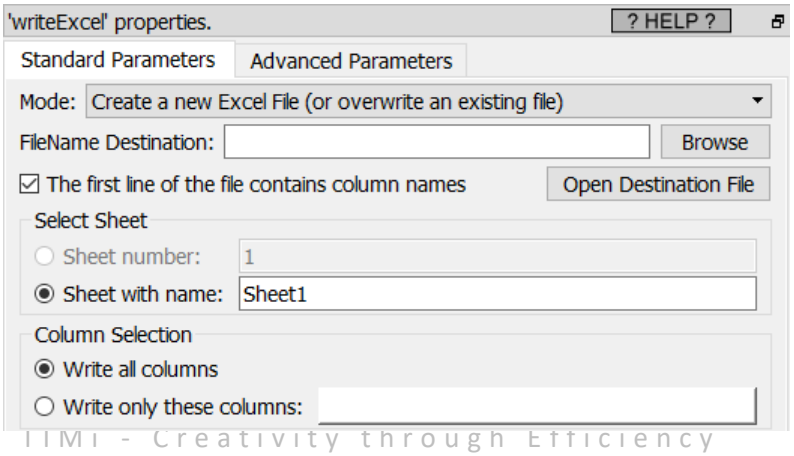
If you intend to redistribute your charts to many people, you should know that there exists an annoying bug inside MSPowerpoint related to "Styles". If you try to open a .pptx file containing a chart that has a "user-defined-style" that is not installed on your computer, then MSPowerpoint resets the colors & layout (i.e. the "style") of the chart to a default ugly value. There are 3 solutions to this problem:

1. Save the file (i.e. the.pptx, .docx, .xlsx file) as a PDF file and distribute the PDF file (and not the .pptx file). This is the easiest solution.
2. Change the colors & layout of your charts **without using** the "quick styling" option in the toolbar (i.e. use the slower "Format plot Area", "Format data points" options).
3. Verify that the styles that you are using are also on the PC of the people that are receiving your charts.

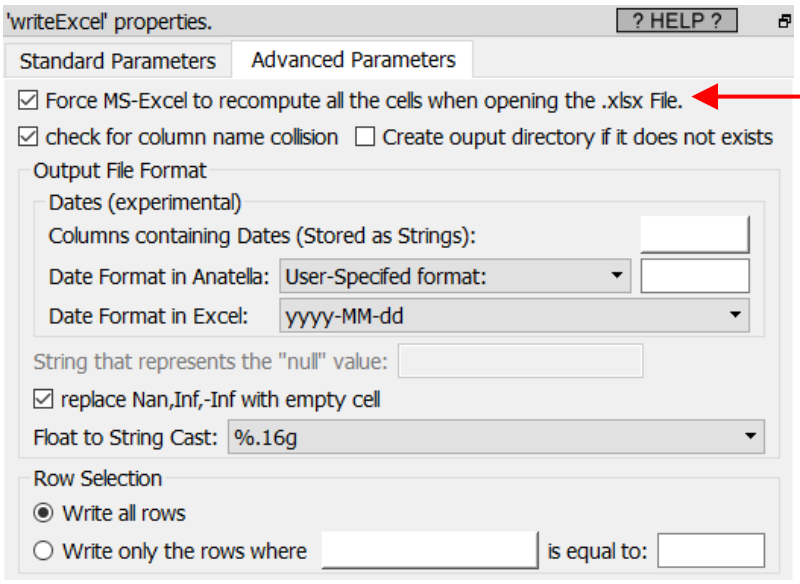
5.27.11. Excel file Writer

Icon:

Property window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

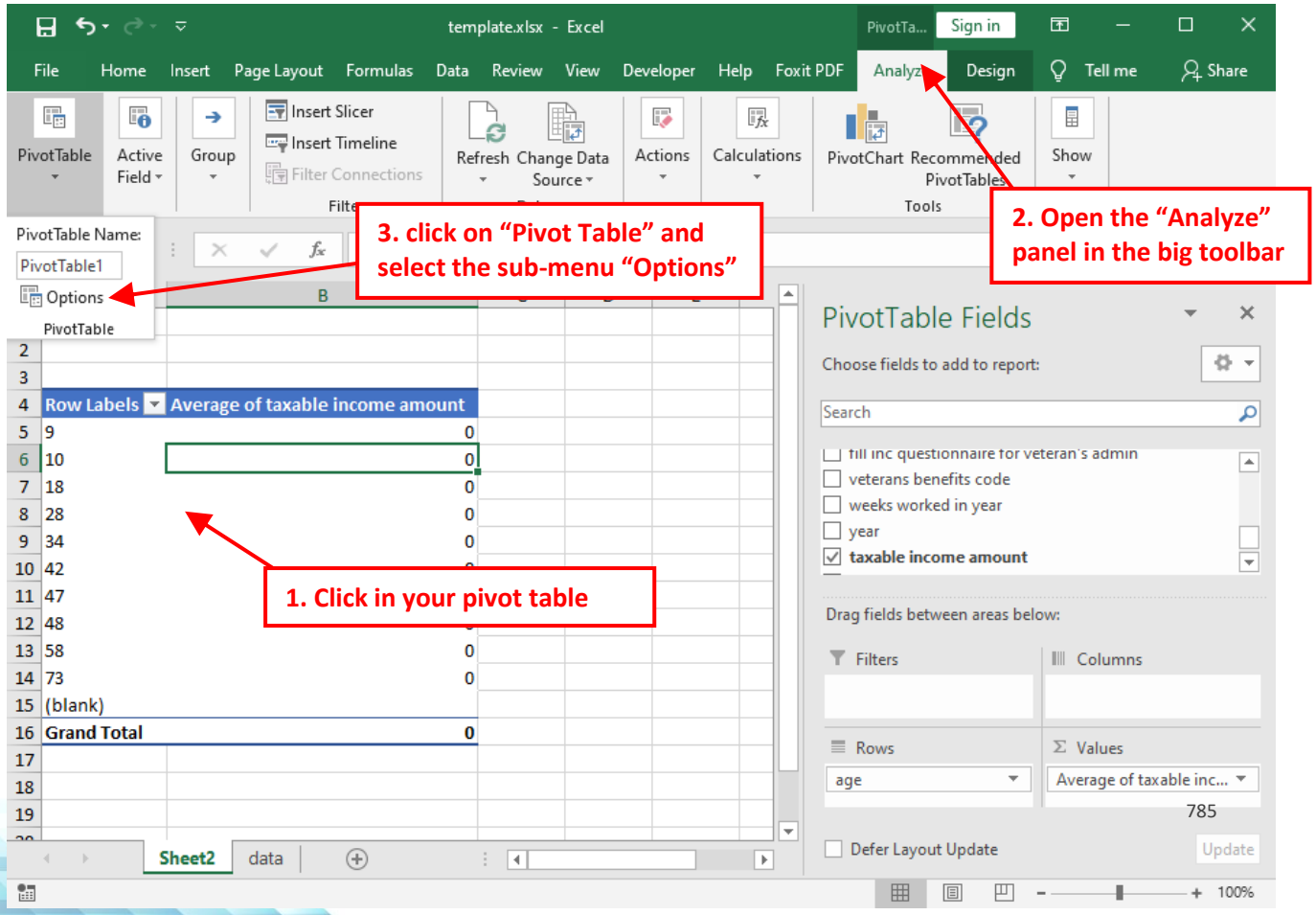


Short description:
Create/Update a .xlsx Excel file using data coming from the transformation graph.

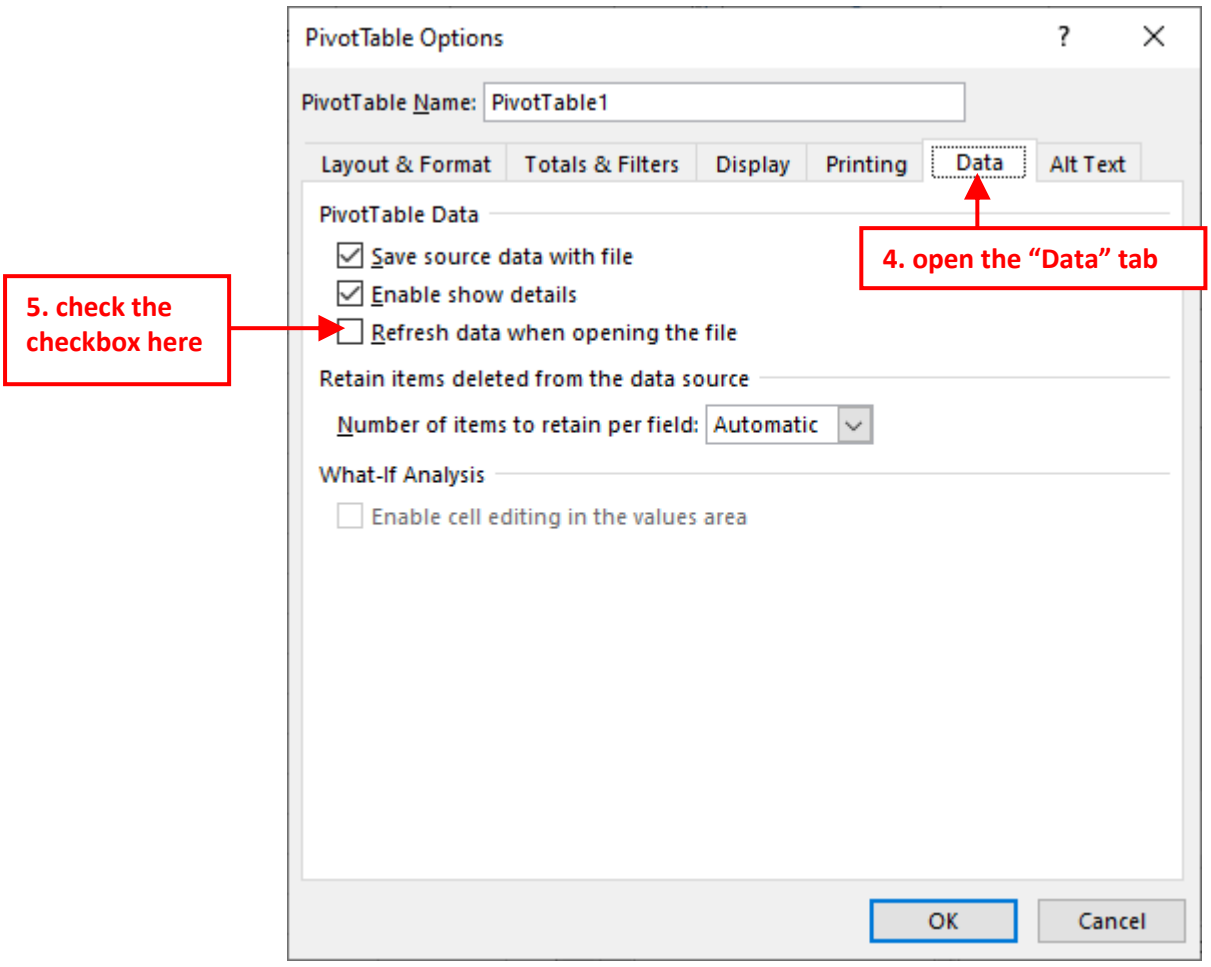
Long Description:
Self-explanatory

Warning: The Anatella option "*Force MS-Excel to recompute all the cells when opening the .xlsx file*" is not refreshing the cells inside the pivot tables. To refresh automatically the pivot tables, follow this procedure:

1. Click anywhere inside your pivot table
2. Open the “Analyze” panel in the big toolbar
3. In the big toolbar, click on “Pivot Table” and select (click) the sub-menu “Options”



4. Inside the “Pivot Table Options” window, open the “Data” tab
5. Check the option named “Refresh Data when opening the file”

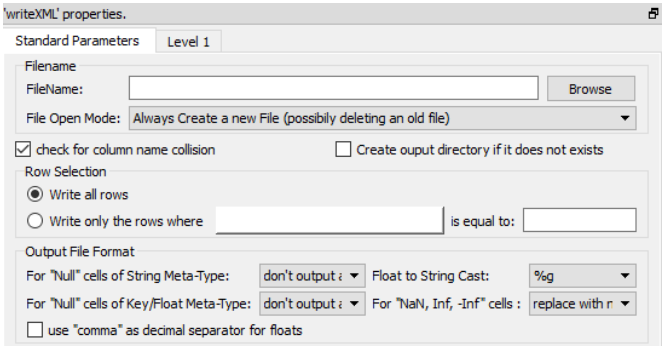


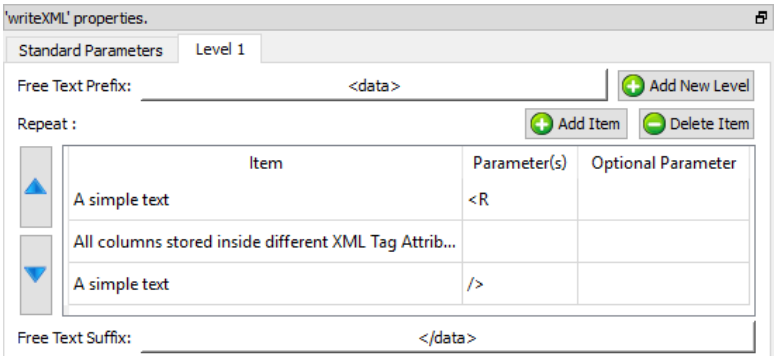
6. Save your .xlsx file (press [CTRL]+[S]).

5.27.12. XML File Writer

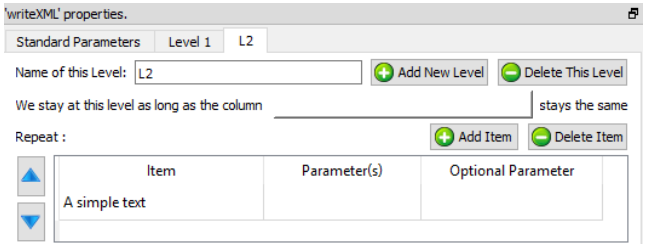
Icon:

Property window:





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



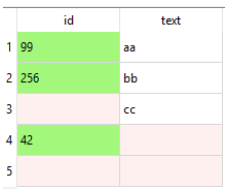
Short description:
Create a XML file that contains a table coming from the Anatella transformation graph.

Long Description:
Create a XML file that contains a table coming from the Anatella transformation graph.

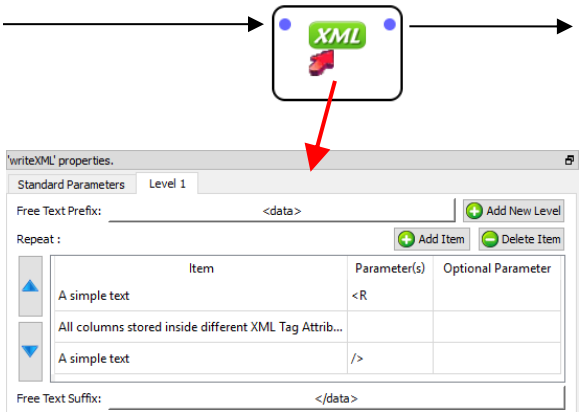
Please refer to section 5.1.1 to have more information on how to specify the filename of the XML file (i.e. You can use relative path and Javascript to specify your filename).

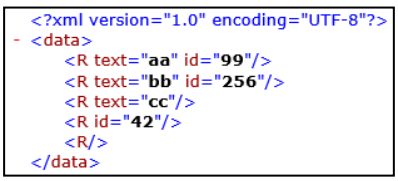
The maximum memory consumption of the writeJSON action is equal to the memory required to store one row of the input table. This means that you can create any JSON file, of any size regardless of the quantity of RAM of your server.

The default settings are producing a simple "one-level" XML file (you'll find more information on the "level" principle below) that represents the input table in a quite straight forward way.
Here are some examples:

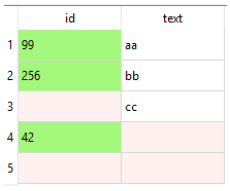


Input table

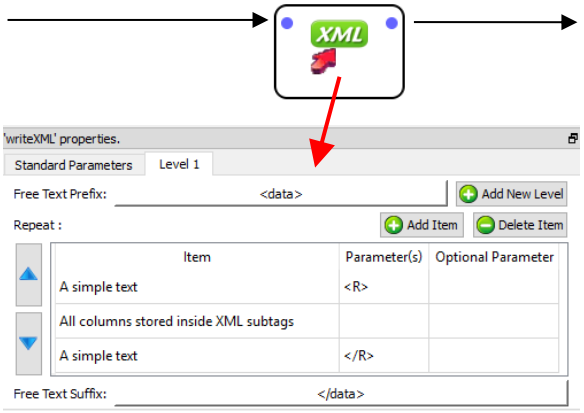


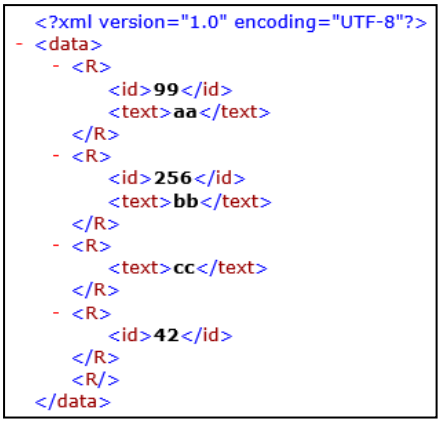


Each XML Tag represents one row of the input table. The different attributes are the different columns.



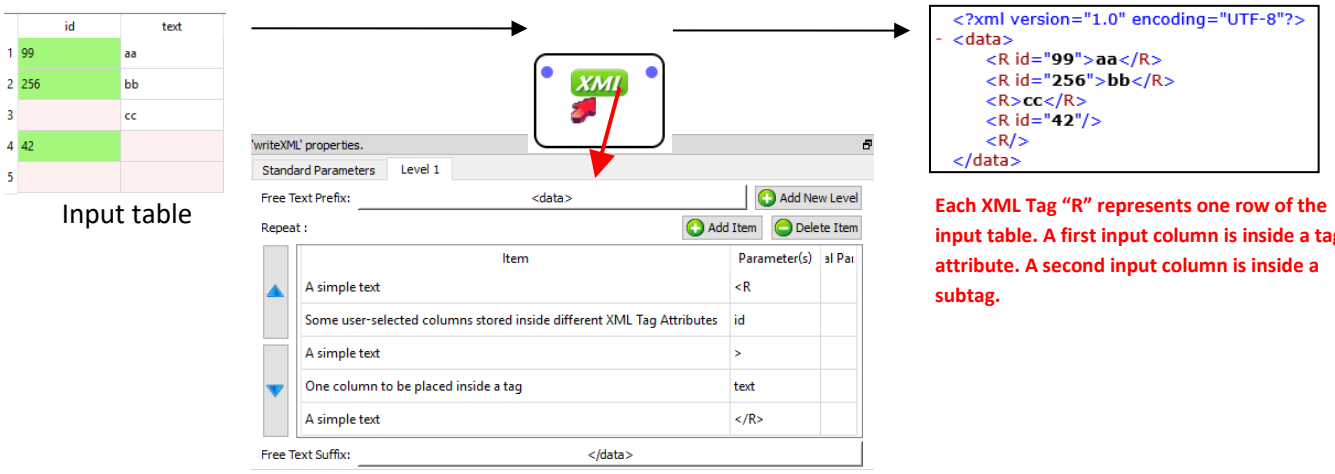
Input table





Each XML Tag "R" represents one row of the input table. The different attributes are the different subtags inside the "R" tag.

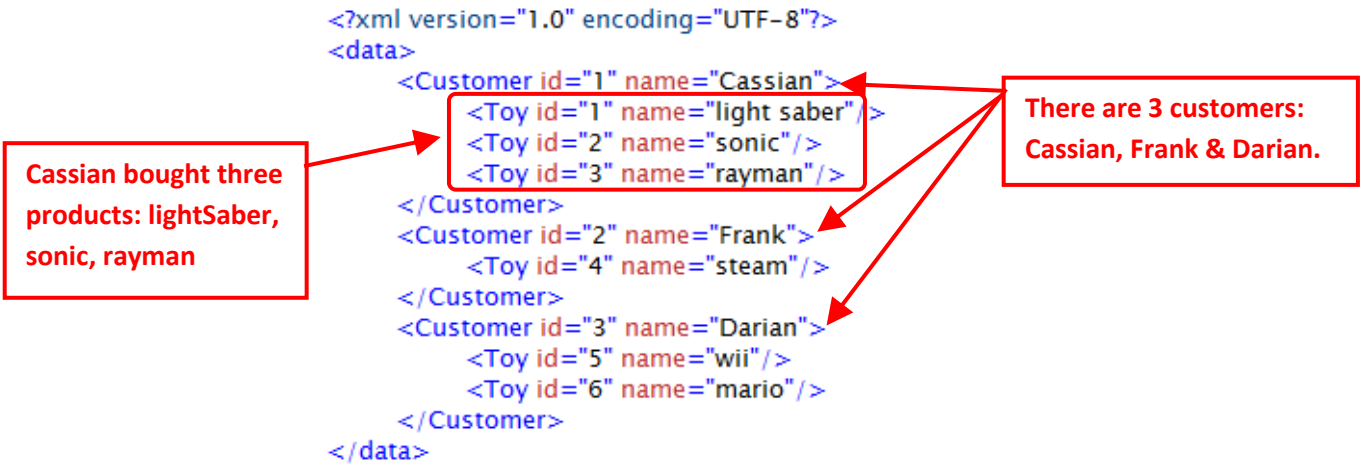
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Let's now assume that you have a XML file that has a 2-level structure:

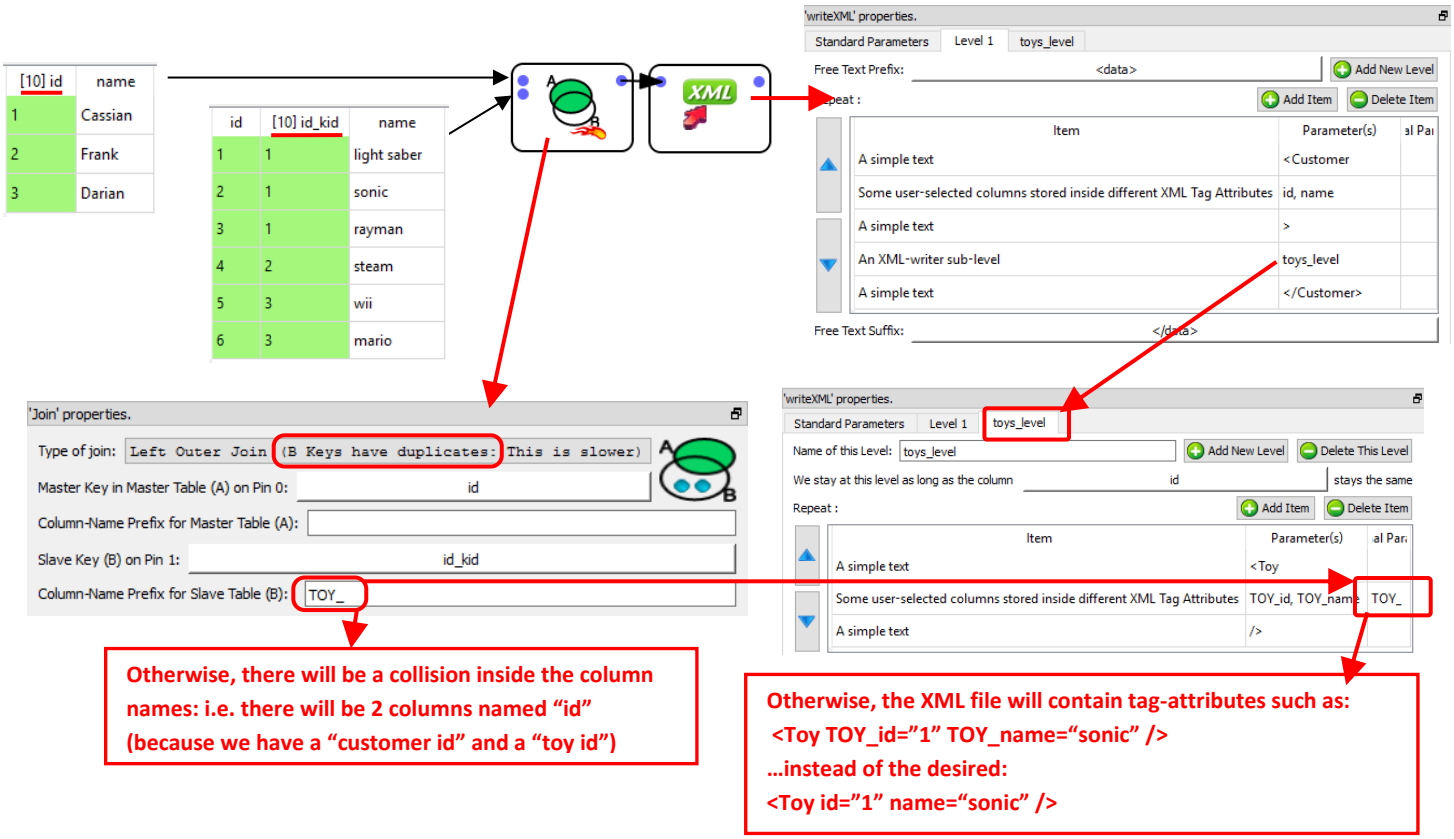
- The first, top level contains informations about different customers.
- The second, bottom level contains informations about the purchases of each of the customer.

For example, we'll have something like this:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The data illustrated in the above XML file is typically initially stored inside two tables (e.g. a "Customers" table and a "Purchases" table) inside a database. To create the above XML file, we'll use the Join Action to join these two tables and simply forward the output table inside the writeXML action: See illustration below:



The above example shows you how to create a two-level XML file (i.e. the two levels are "Customers" and "Purchases"). Inside Anatella, you can create any XML files with any number of imbricated levels.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

It can also happen that a specific level contains several different sub-levels: For example, consider the following XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
	<Item ID="101" Name="blue car">
		<Parts>
			<Part ID="11" Name="wheel"/>
			<Part ID="13" Name="engine"/>
			<Part ID="14" Name="blue car frame"/>
		</Parts>
		<Factories>
			<Factory ID="F1" Country="Belgium"/>
			<Factory ID="F2" Country="France"/>
		</Factories>
	</Item>
	<Item ID="102" Name="red car">
		<Parts>
			<Part ID="11" Name="wheel"/>
			<Part ID="13" Name="engine"/>
			<Part ID="15" Name="red car frame"/>
		</Parts>
		<Factories>
			<Factory ID="F3" Country="LatAm"/>
			<Factory ID="F4" Country="UK"/>
		</Factories>
	</Item>
	<Item ID="103" Name="green car">
		<Parts>
			<Part ID="11" Name="wheel"/>
			<Part ID="13" Name="engine"/>
			<Part ID="16" Name="green car frame"/>
		</Parts>
		<Factories>
			<Factory ID="F5" Country="USA"/>
		</Factories>
	</Item>
	<Item ID="104" Name="blue bike">
		<Parts>
			<Part ID="17" Name="handlebar"/>
			<Part ID="18" Name="blue bike frame"/>
		</Parts>
		<Factories>
			<Factory ID="F1" Country="Belgium"/>
		</Factories>
	</Item>
	<Item ID="105" Name="mecanno">
		<Parts>
			<Part ID="19" Name="screw"/>
			<Part ID="20" Name="bolt"/>
		</Parts>
		<Factories>
			<Factory ID="F3" Country="LatAm"/>
		</Factories>
	</Item>
</data>
```

The above XML file contains data about different Items (i.e. the first level is about “Items”).
Each item is :

- composed of different parts
- produced in different Factories.

There are thus two different second-level data: One sub-level is about “Parts” and the other is about “Factories”. Typically, the data illustrated in the above XML file is initially stored inside (at least) three tables (e.g. an “Items” table, a “Parts” table and a “Factories” table) inside a database. To create the above XML file, we’ll join these three tables and simply forward the output table inside the writeXML action: See the illustration below:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

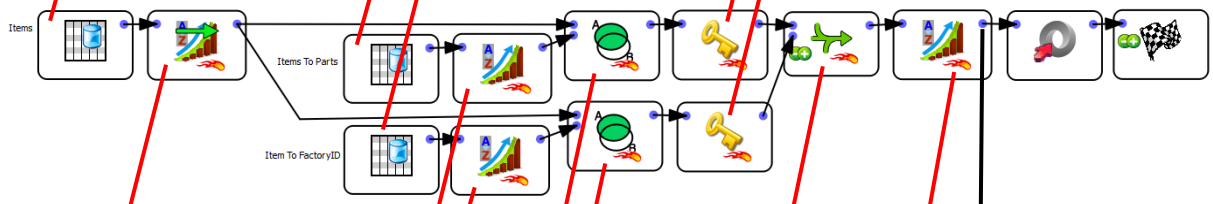
ItemID	ItemName
101	blue car
102	red car
103	green car
104	blue bike
105	mecanno

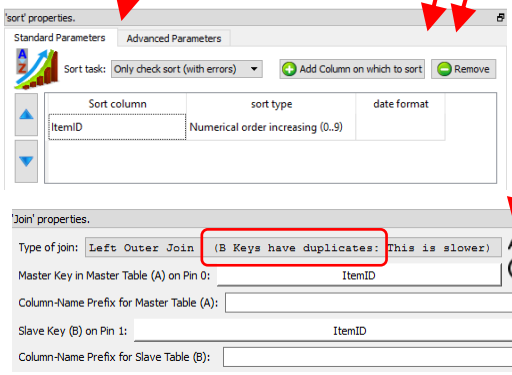
ItemID	PartID	PartName
101	11	wheel
101	13	engine
101	14	blue car frame
102	11	wheel
102	13	engine
102	15	red car frame
103	11	wheel
103	13	engine
103	16	green car frame
104	17	handlebar
104	18	blue bike frame
105	19	screw
105	20	bolt

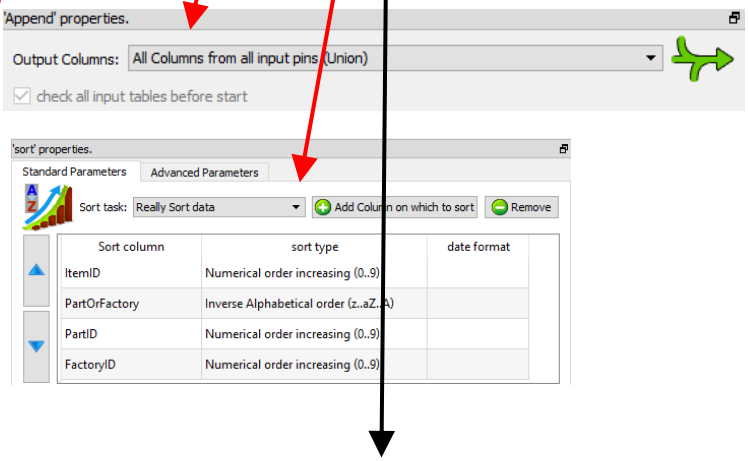
ItemID	FactoryID	FactoryCountry
101	F1	Belgium
101	F2	France
102	F3	LatAm
102	F4	UK
103	F5	USA
104	F1	Belgium
105	F3	LatAm

Create a constant column named "PartOrFactory" that contains "Part"

Create a constant column named "PartOrFactory" that contains "Factory"





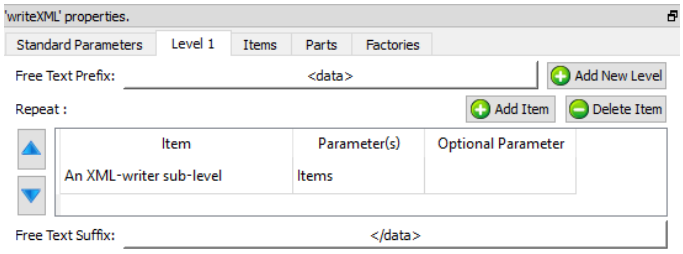


ItemID	ItemName	PartOrFactory	PartID	PartName	FactoryID	FactoryCountry
101	blue car	Part	11	wheel		
101	blue car	Part	13	engine		
101	blue car	Part	14	blue car frame		
101	blue car	Factory			F1	Belgium
101	blue car	Factory			F2	France
102	red car	Part	11	wheel		
102	red car	Part	13	engine		
102	red car	Part	15	red car frame		
102	red car	Factory			F3	LatAm
102	red car	Factory			F4	UK
103	green car	Part	11	wheel		
103	green car	Part	13	engine		
103	green car	Part	16	green car frame		
103	green car	Factory			F5	USA
104	blue bike	Part	17	handlebar		
104	blue bike	Part	18	blue bike frame		
104	blue bike	Factory			F1	Belgium
105	mecanno	Part	19	screw		
105	mecanno	Part	20	bolt		
105	mecanno	Factory			F3	LatAm

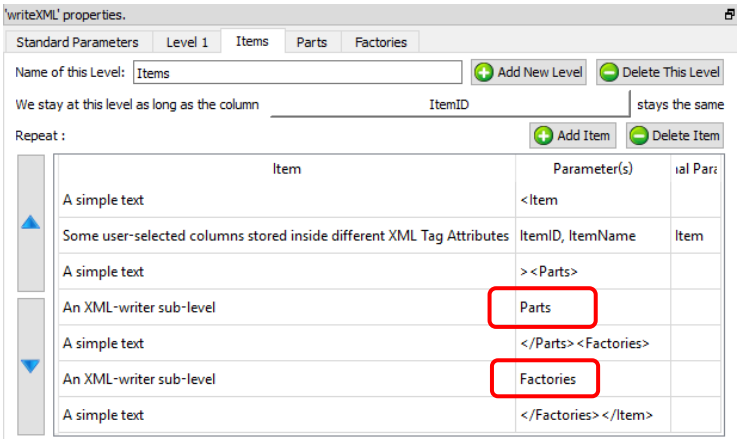
Items Parts Factories

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameters of the XML writeXML action are:



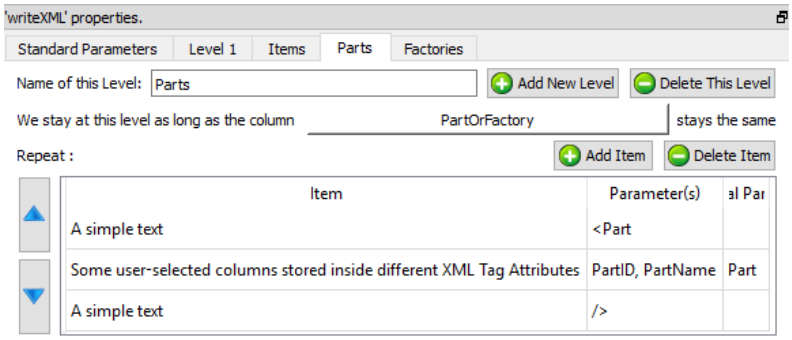
The "Level 1" is actually almost empty

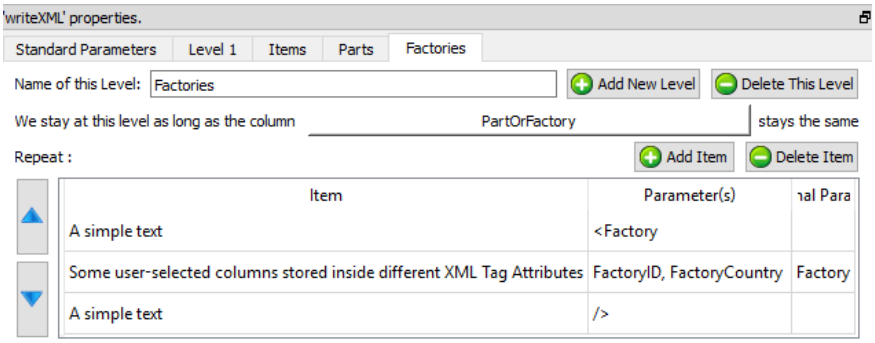


The "Item" level is composed of two sub-levels "Parts" and "Factories".

The order is important: first "Parts", second "Factories": i.e. the order must match the order used inside the input table.

The "Parts" level and the "Factories" level are straight forward:





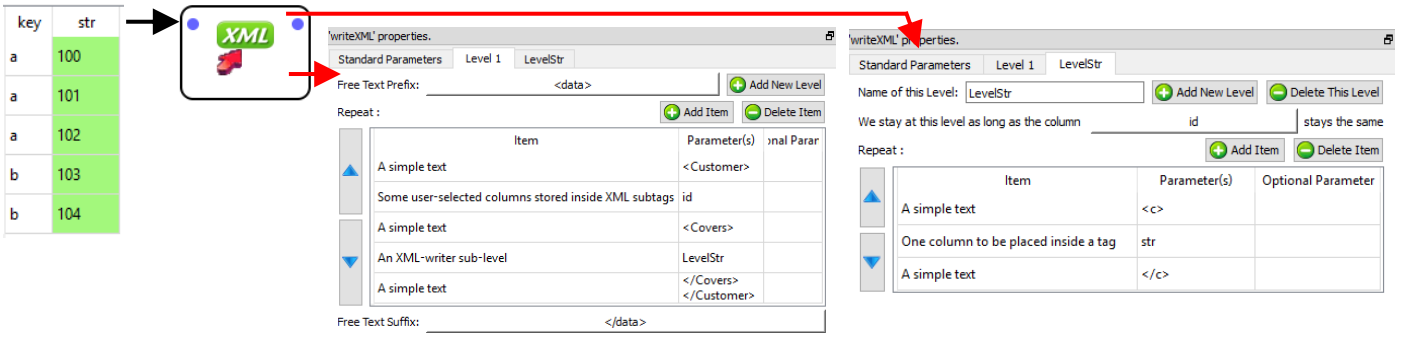
You now know how to create any XML file with any structure and of any size (i.e. the RAM memory consumption of the XML writeXML action is only the memory required to store ONE row of the input table).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

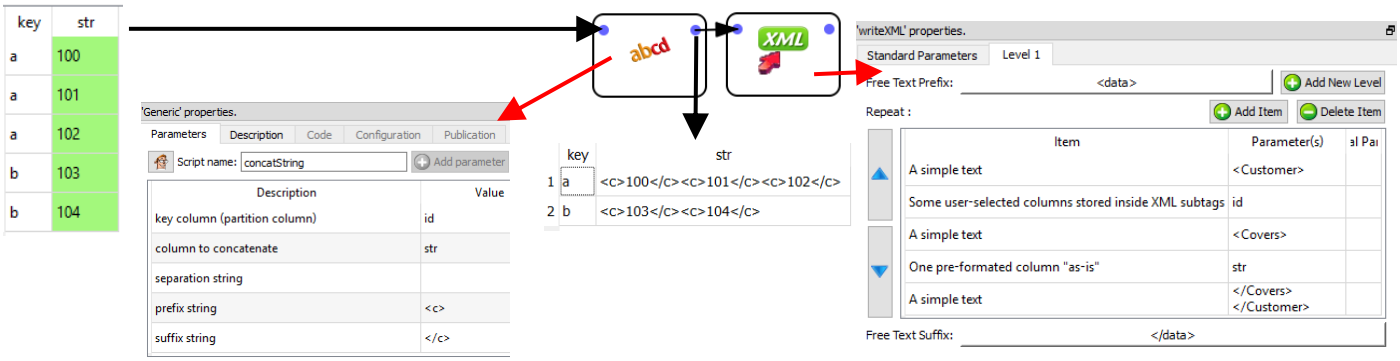
We'll now introduce a small trick to reduce computation time.
Let's assume that we want to create the following XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
    <Customer>
        <id>1</id>
        <Covers><c>100</c><c>101</c><c>102</c></Covers>
    </Customer>
    <Customer>
        <id>2</id>
        <Covers><c>103</c><c>104</c></Covers>
    </Customer>
</data>
```

To create the above XML file, we can use 2 techniques. Here is the first one (that follows the general principles already explained here above):



…and here is the second technique:



The advantages of the second technique are:

- the input table to the XML writeXML action (i.e. the table

	key	str
1	a	<c>100</c><c>101</c><c>102</c>
2	b	<c>103</c><c>104</c>

) has only 2 rows (compared to 5 rows for the first technique). In general, manipulating tables that contain a smaller quantity of rows leads to a reduced computing time. This is particularly true if the rows are very long (i.e. one thousand columns). The input table in the above example has only a limited number of columns (less than 300), thus the computing-time gain will be inexistent.
- the input table to the XML writeXML action consumes less disk space since there are less un-needed repetitions. For example, you can see that, for the first technique, the key "a" is repeated 3 times while, for the second technique, the key "a" only appears one time.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

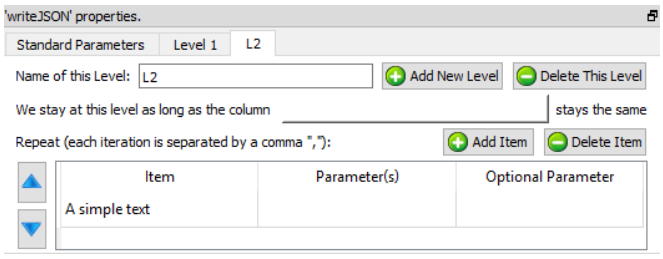
The disadvantages of the second technique are:

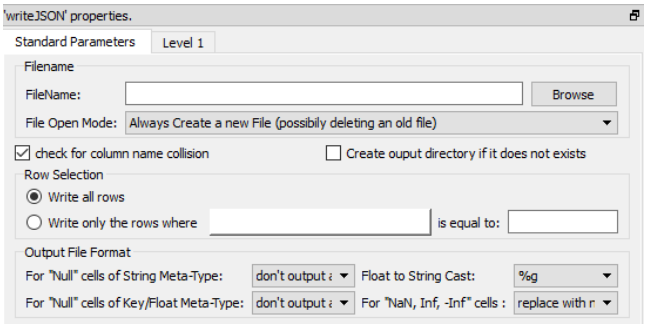
- We were forced to use the concatString Action: This action is very slow because it involves concatenating a large quantity of (possibly very large) strings (and concatenating many strings is always a slow operation). One trick to lessen the impact of this slow speed is to put the concatString Action inside a N-Way multithreaded section.
- The second technique only works for very simple XML files where we don't have many different imbricated levels inside the XML file.
- When using the second technique, we create a row that consumes a large quantity of RAM memory (since the row contains a cell that is the concatenation of all the data required to write the level-2 for a particular level-1). The second technique has thus a quite large memory consumption and it can possibly lead to crashes if the XML file is too big.

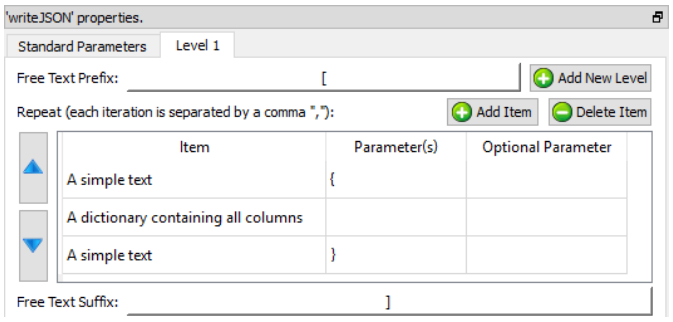
5.27.13. JSON File Writer

Icon:

Property window:







Short description:
Create a JSON file that contains a table coming from the Anatella transformation graph.

Long Description:
Please refer to section 5.1.1 to have more information on how to specify the filename of the JSON file (i.e. You can use relative path and Javascript to specify your filename).

The maximum memory consumption of the writeJSON action is equal to the memory required to store one row of the input table. This means that you can create any JSON file, of any size regardless of the quantity of RAM of your server.

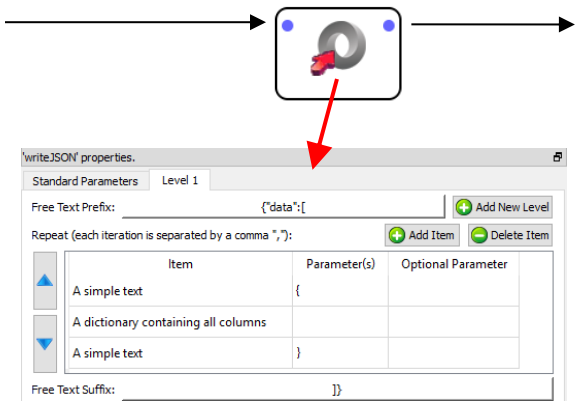
The default settings are producing a simple "one-level" JSON file (you'll find more information on the "level" principle below) that represents the input table in a quite straight forward way.

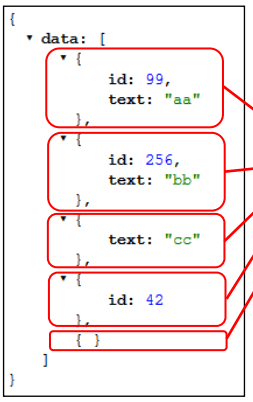
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here are some examples:

	id	text
1	99	aa
2	256	bb
3		cc
4	42	
5		

Input table



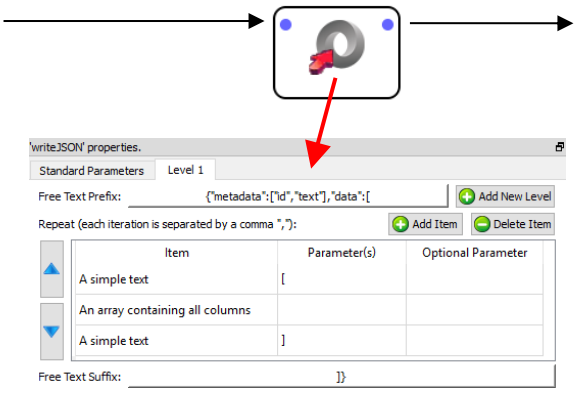


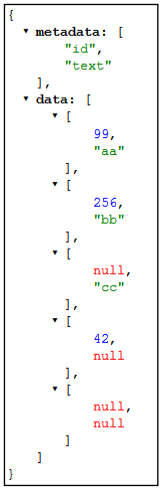
Each dictionnary represents one row of the input table

Output JSON file: Array of Dictionnaries

	id	text
1	99	aa
2	256	bb
3		cc
4	42	
5		

Input table

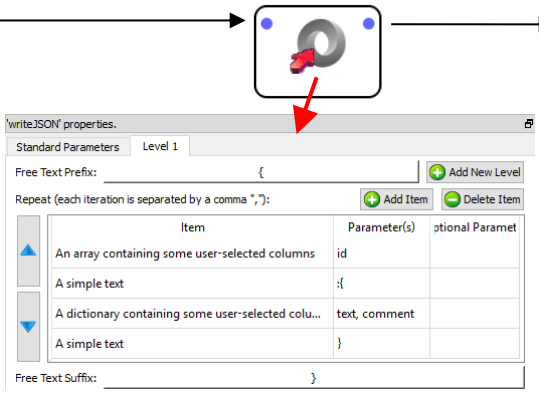




Output JSON file: Array of Arrays

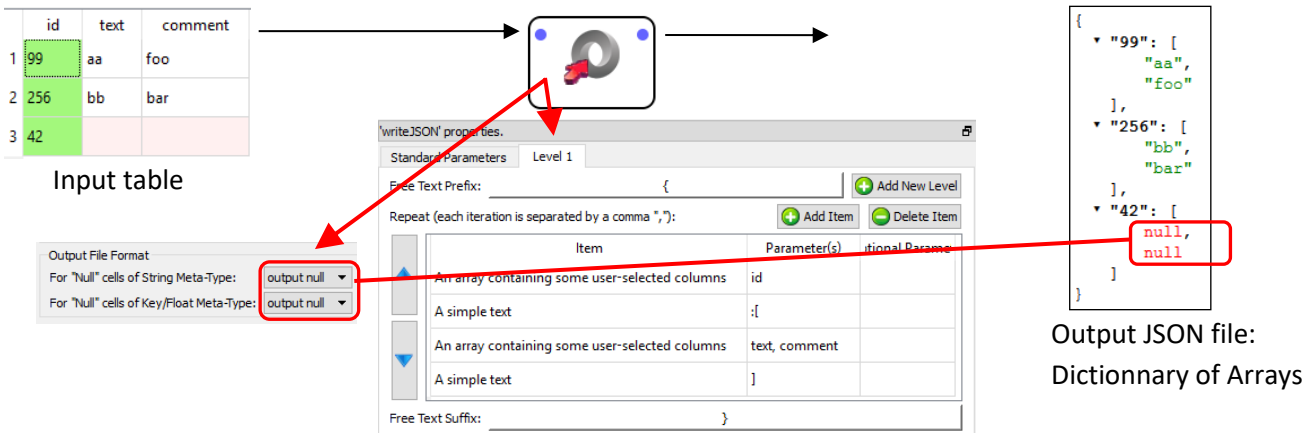
	id	text	comment
1	99	aa	foo
2	256	bb	bar
3	42		

Input table



Output JSON file: Dictionnary of Dictionnaries

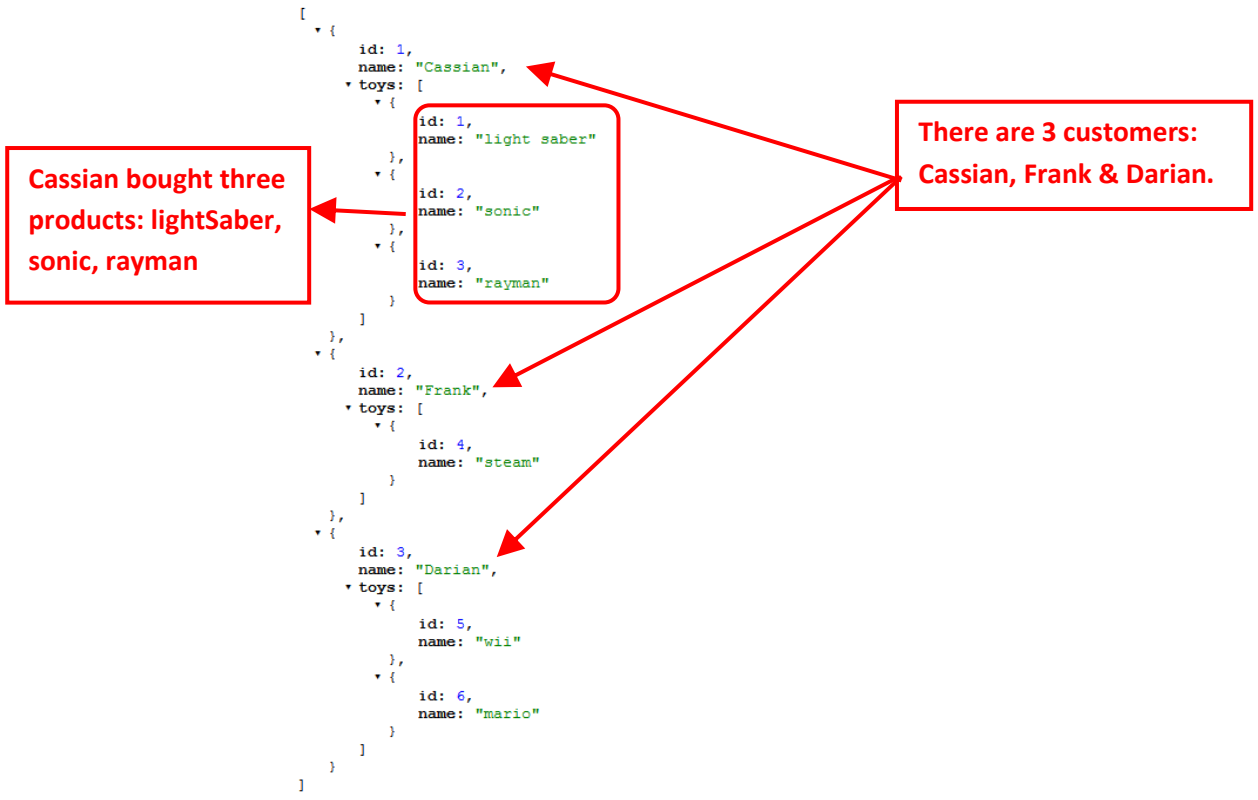
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Let's now assume that you have a JSON file that has a 2-level structure:

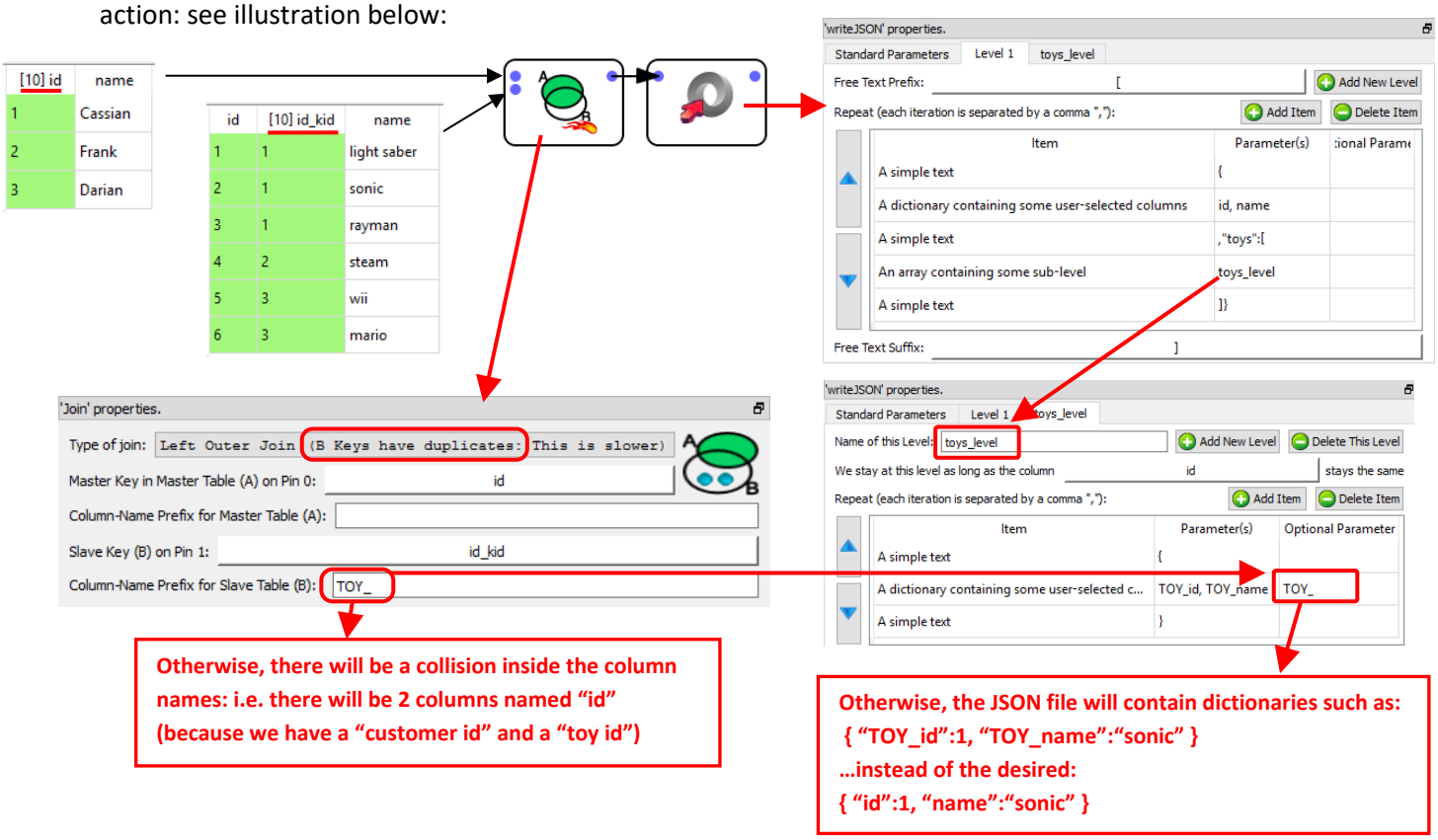
- The first, top level contains informations about different customers.
- The second, bottom level contains informations about the purchases of each of the customer.

For example, we'll have something like this:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The data illustrated in the above JSON file is typically initially stored inside two tables (e.g. a "Customers" table and a "Purchases" table) inside a database. To create the above JSON file, we'll use the Join Action to join these two tables and simply forward the output table inside the writeJSON action: see illustration below:



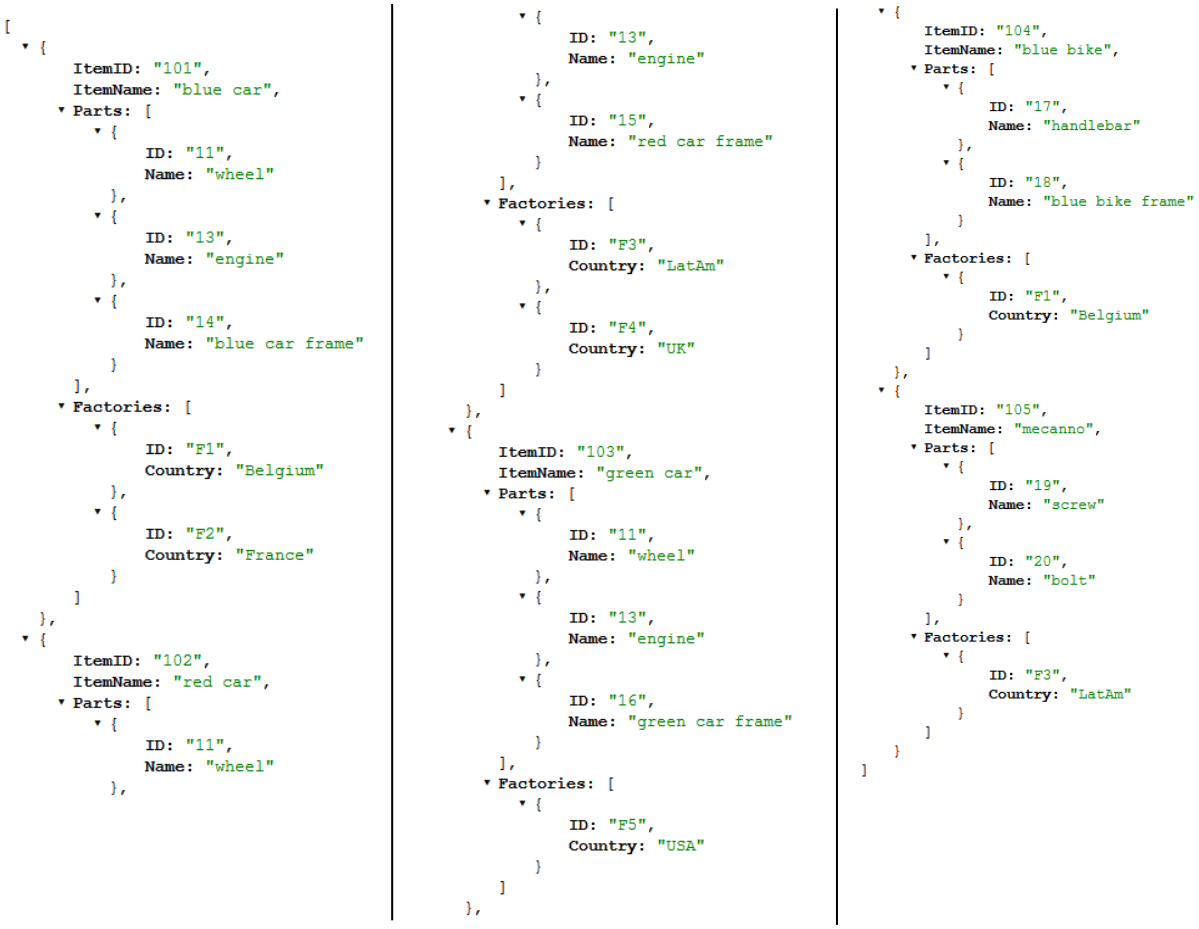
Otherwise, there will be a collision inside the column names: i.e. there will be 2 columns named "id" (because we have a "customer id" and a "toy id")

Otherwise, the JSON file will contain dictionaries such as:
{ "TOY_id":1, "TOY_name":"sonic" }
…instead of the desired:
{ "id":1, "name":"sonic" }

The above example shows you how to create a two-level JSON file (i.e. the levels are "Customers" and "Purchases"). Inside Anatella, you can create any JSON files with any number of imbricated levels.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

It can also happen that a specific level contains several different sub-levels: For example, consider the following JSON file:



The above JSON file contains data about different Items (i.e. the first level is about “Items”).
Each item is :

- composed of different parts
- produced in different Factories.

There are thus two different second-level data: One sub-level is about “Parts” and the other is about “Factories”. Typically, the data illustrated in the above JSON file is initially stored inside (at least) three tables (e.g. an “Items” table, a “Parts” table and a “Factories” table) inside a database. To create the above JSON file, we’ll join these three tables and simply forward the output table inside the writeJSON action: See the illustration below:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

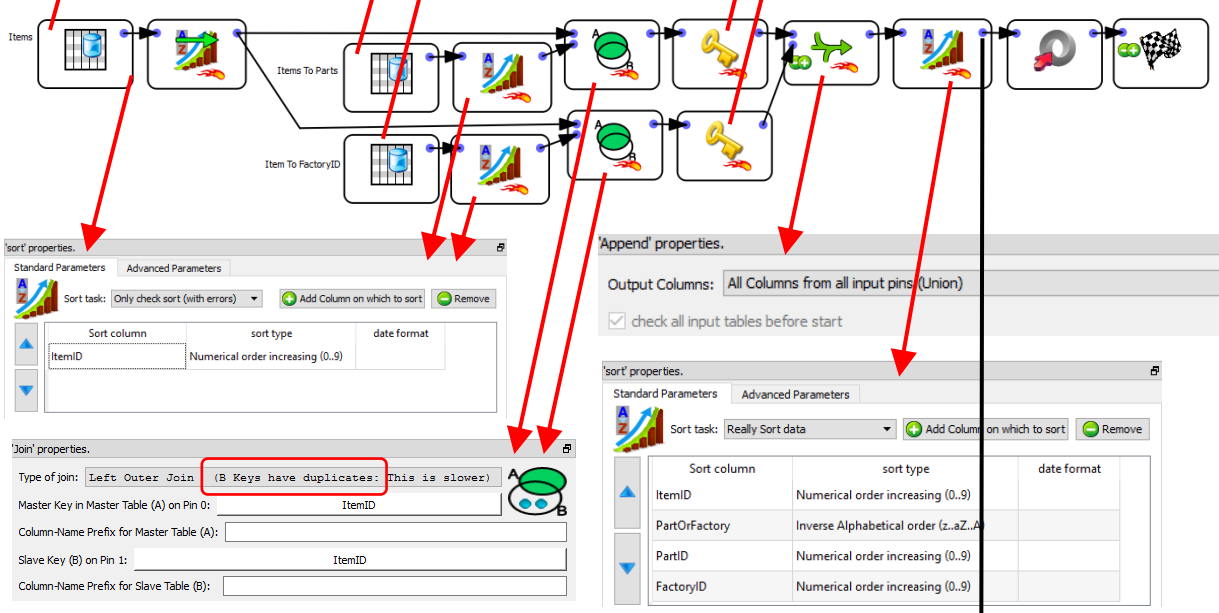
ItemID	ItemName
101	blue car
102	red car
103	green car
104	blue bike
105	mecanno

ItemID	PartID	PartName
101	11	wheel
101	13	engine
101	14	blue car frame
102	11	wheel
102	13	engine
102	15	red car frame
103	11	wheel
103	13	engine
103	16	green car frame
104	17	handlebar
104	18	blue bike frame
105	19	screw
105	20	bolt

ItemID	FactoryID	FactoryCountry
101	F1	Belgium
101	F2	France
102	F3	LatAm
102	F4	UK
103	F5	USA
104	F1	Belgium
105	F3	LatAm

Create a constant column named "PartOrFactory" that contains "Part"

Create a constant column named "PartOrFactory" that contains "Factory"



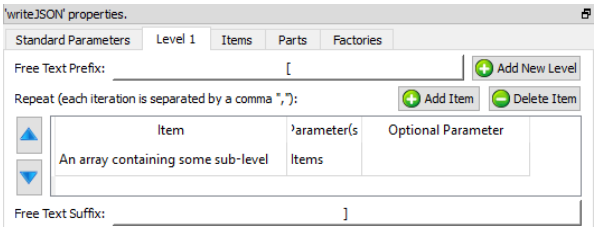
We finally obtain the following table as input to the writeJSON action:

ItemID	ItemName	PartOrFactory	PartID	PartName	FactoryID	FactoryCountry
101	blue car	Part	11	wheel		
101	blue car	Part	13	engine		
101	blue car	Part	14	blue car frame		
101	blue car	Factory			F1	Belgium
101	blue car	Factory			F2	France
102	red car	Part	11	wheel		
102	red car	Part	13	engine		
102	red car	Part	15	red car frame		
102	red car	Factory			F3	LatAm
102	red car	Factory			F4	UK
103	green car	Part	11	wheel		
103	green car	Part	13	engine		
103	green car	Part	16	green car frame		
103	green car	Factory			F5	USA
104	blue bike	Part	17	handlebar		
104	blue bike	Part	18	blue bike frame		
104	blue bike	Factory			F1	Belgium
105	mecanno	Part	19	screw		
105	mecanno	Part	20	bolt		
105	mecanno	Factory			F3	LatAm

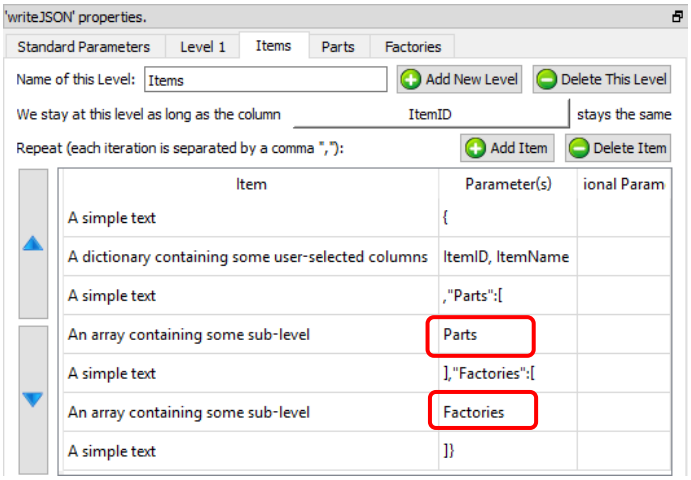
Items | Parts | Factories

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameters of the writeJSON action are:



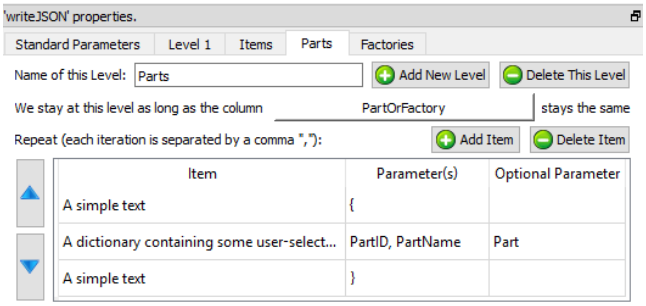
The "Level 1" is actually empty

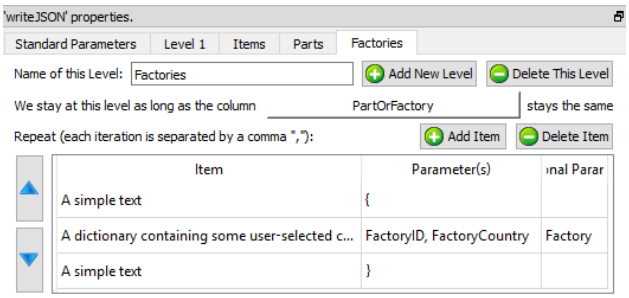


The "Item" level is composed of two sub-levels "Parts" and "Factories".

The order is important: first "Parts", second "Factories": i.e. the order must match the order used inside the input table.

The "Parts" level and the "Factories" level are straight forward:





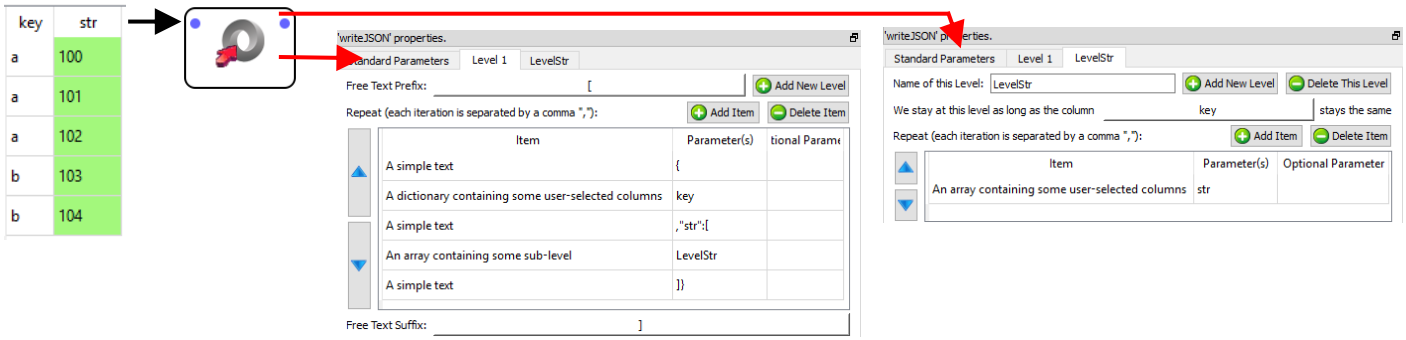
You now know how to create any JSON file of any size with any structure (i.e. the RAM memory consumption of the writeJSON action is only the memory required to store ONE row of the input table).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

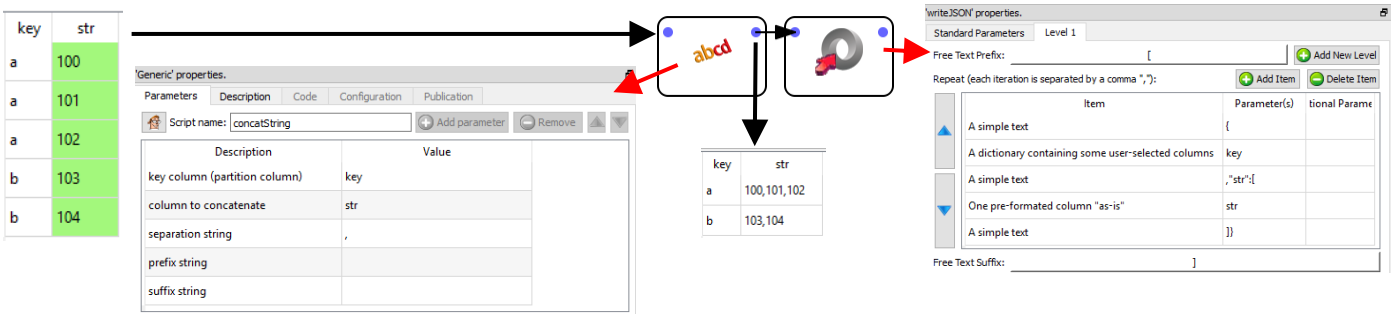
We'll now introduce a small trick to reduce computation time.
Let's assume that we want to create the following JSON file:


```
[
  {
    key: "a",
    str: [
      100,
      101,
      102
    ]
  },
  {
    key: "b",
    str: [
      103,
      104
    ]
  }
]
```

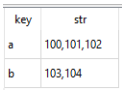

To create the above JSON file, we can use 2 techniques. Here is the first one (that follows the general principles already explained here above):



…and here is the second technique:



The advantages of the second technique are:

- the input table to the writeJSON action (i.e. the table ) has only 2 rows (compared to 5 rows for the first technique). In general, manipulating tables that contain a small quantity of rows leads to a reduced computing time. This is particularly true if the rows are very long (i.e. one thousand columns). The input table in the above example has only a limited number of columns (less than 300), thus the computing-time gain will be inexistent.
- the input table to the writeJSON action consumes less disk space since there are less un-needed repetitions. For example, you can see that, for the first technique, the key "a" is repeated 3 times while, for the second technique, the key "a" only appears one time.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

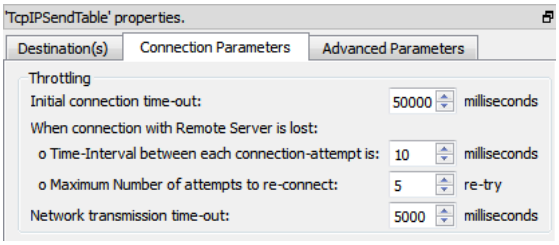
The disadvantages of the second technique are:

- We were forced to use the

concatString Action: This action is very slow because it involves concatenating a large quantity of (possibly very large) strings (and concatenating many strings is always a slow operation). One trick to lessen the impact of this slow speed is to put the

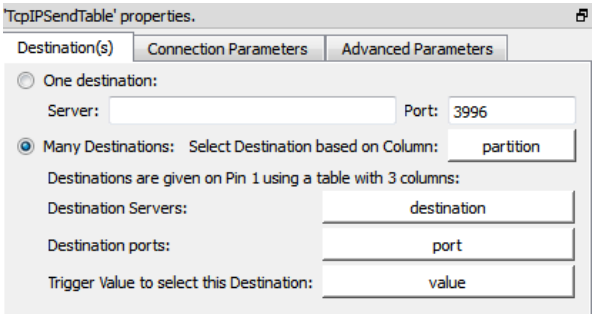
concatString Action inside a N-Way multithreaded section.
- The second technique only works for very simple JSON files where we don't have many different imbricated levels inside the JSON file.
- When using the second technique, we create a row that consumes a large quantity of RAM memory (since the row contains a cell that is the concatenation of all the data required to write the level-2 for a particular level-1). The second technique has thus a quite large memory consumption and it can possibly lead to crashes if the JSON file is too big.

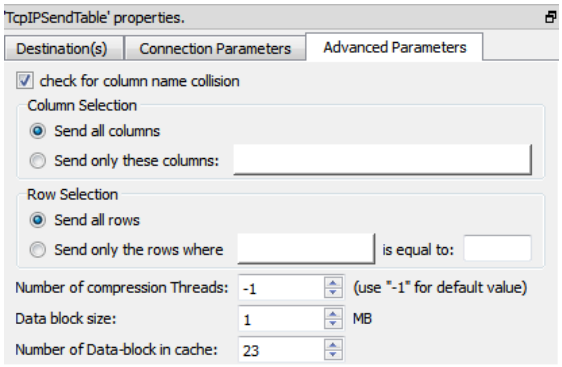
5.27.14. TCP-IP Send table



Icon:

Property window:





Short description:

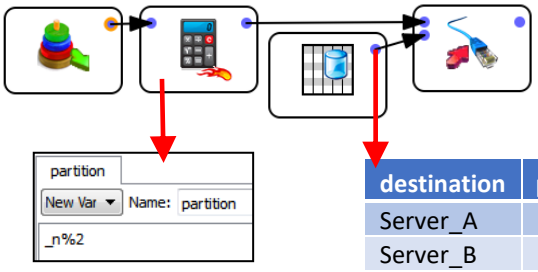
Sends a table to possibly many destinations, using TCP/IP connection(s).

Long Description:

The TcpIPReceiveTable Action and the TcpIPSendTable Action are used together to form the "Map" part inside a "Map-Reduce" algorithm (i.e. it allows Anatella to make distributed computations on several PC's).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's have a look at a small example:



destination	port	value
Server_A	3996	0
Server_B	3996	1

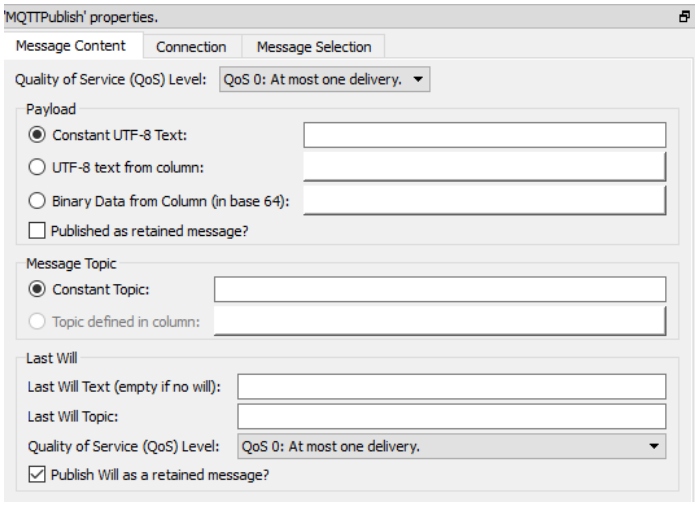
The above example sends half the row of the input table to the "Server_A" and the other half to the "Server_B". The split is based on the "partition" column: i.e. All the rows where "partition" equals zero are sent to the "Server_A" (… and the others to the "Server_B").

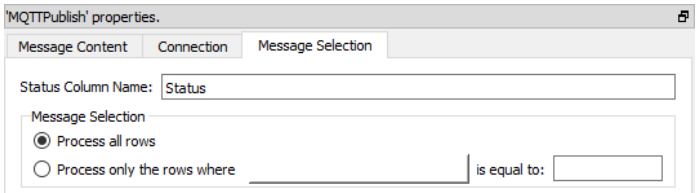
The TcpIPSendTable Action use an asynchronous (i.e. non-blocking) I/O algorithms (See the section 5.2.6.2. about asynchronous I/O algorithms). This means that Anatella is sending (and also compressing) several data-blocks (full of rows) while, simultaneously, running the rest of the data transformation. The maximum quantity of data-block kept in RAM is defined inside the parameter "*Number of data-block in cache*". A large value allows to cope with sudden&brief "speed drops" of your computer network (but it also implies a higher RAM consumption).

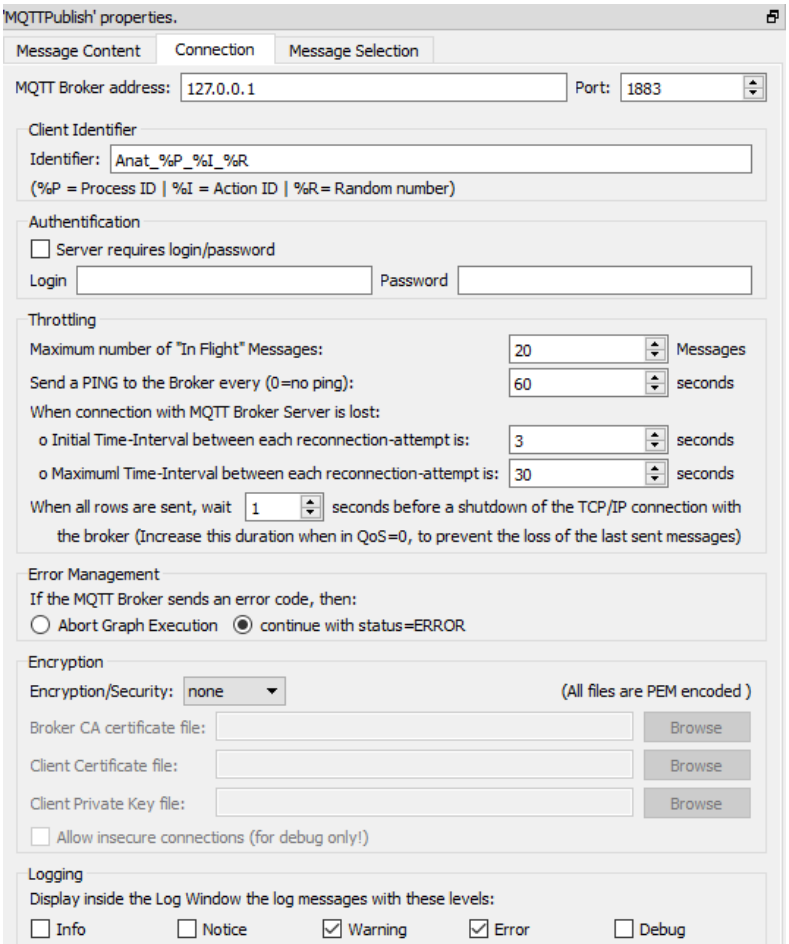
5.27.15. MQTT Publish (IoT connector)

Icon:

Property window:







©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

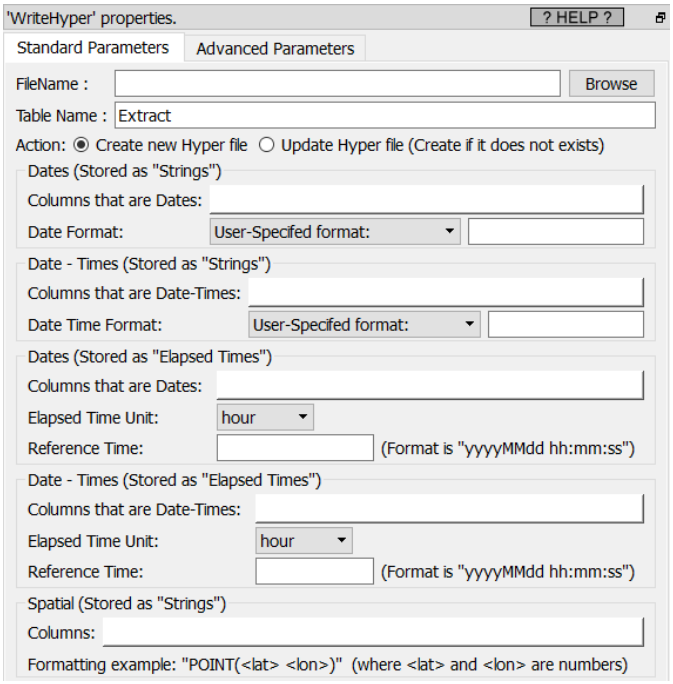
Short description:
Allow to connect to an "IoT" broker using the MQTT protocol to send (publish) messages.

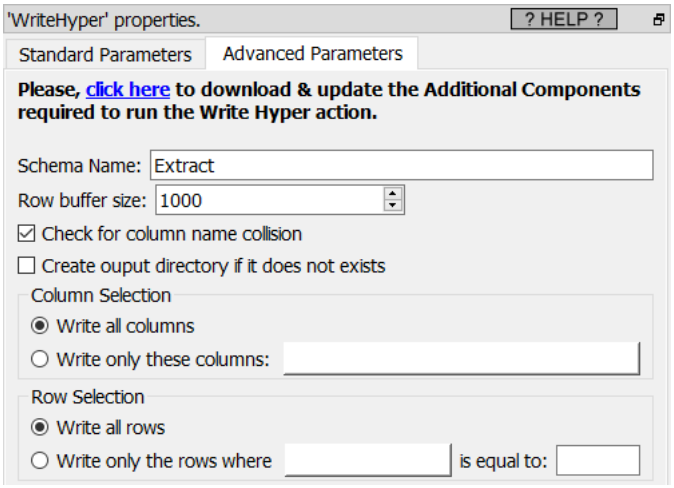
Long Description:
Publish MQTT messages

5.27.16. Create Tableau Hyper File (.hyper)

Icon:

Property window:





Short description:
Create/Update the new Tableau Hyper files.

Long Description:
Self-explanatory.

Once you have created (or updated) your .hyper file, you can easily copy (i.e. "publish") your .hyper file to your remote tableau server using the

"Tableau Publish" action (see section 5.22.5. for more details on this action).

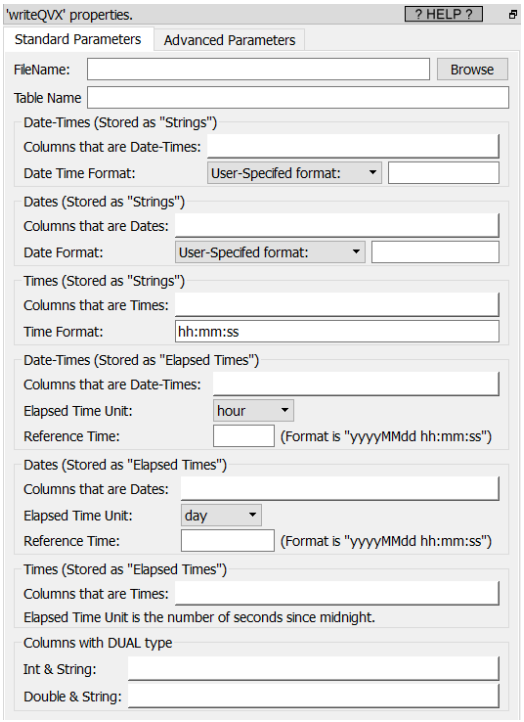
The action surpersedes and replaces the old Action from section 5.26.4. that was previously used to interact with Tableau.

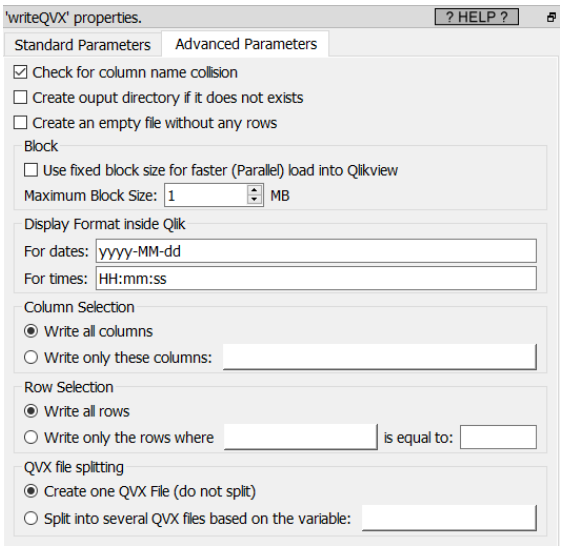
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.27.17. Write Qlikview QVX

Icon:

Property window:





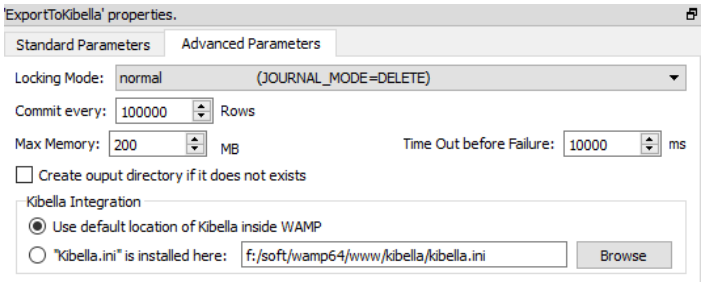
Short description:
Create Qlickview QVX files

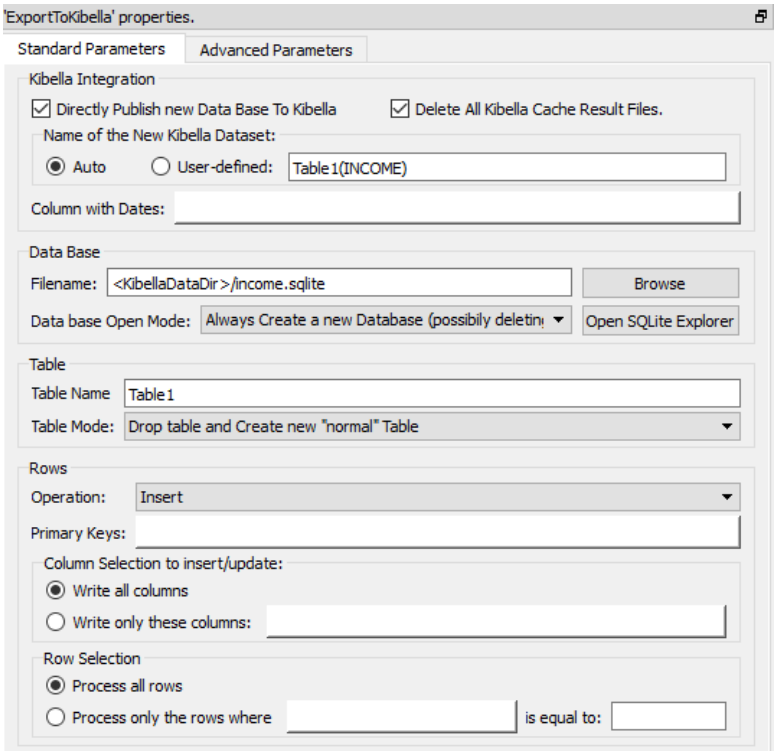
Long Description:
For the DUAL column type: the int/double and the string must be separated by a white space char.

5.27.18. Export To Kibella

Icon:

Property window:



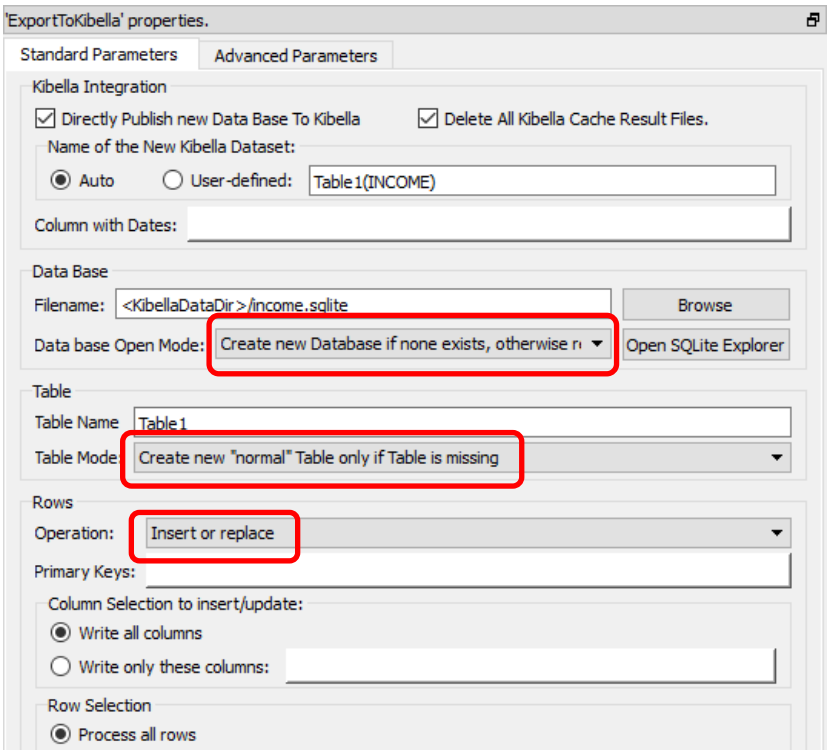


Short description:
Export Data to Kibella

Long Description:
For more information about Kibella, see the document KibellaQuickUserGuide.pdf

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

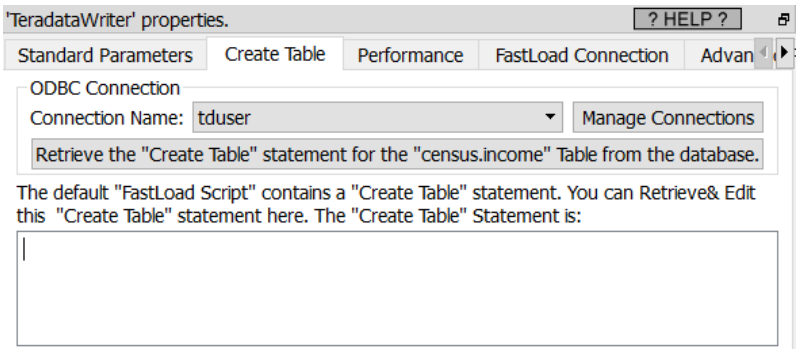
With the default settings, you erase&replace the previous database.
To incrementally add rows inside the Kibella database (e.g. for real-time display), use:

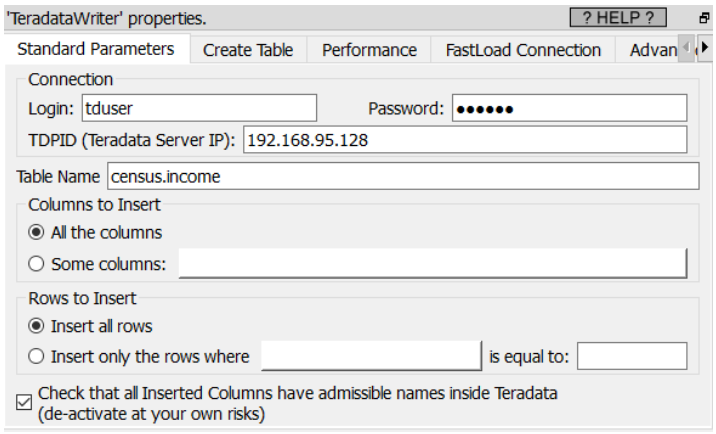


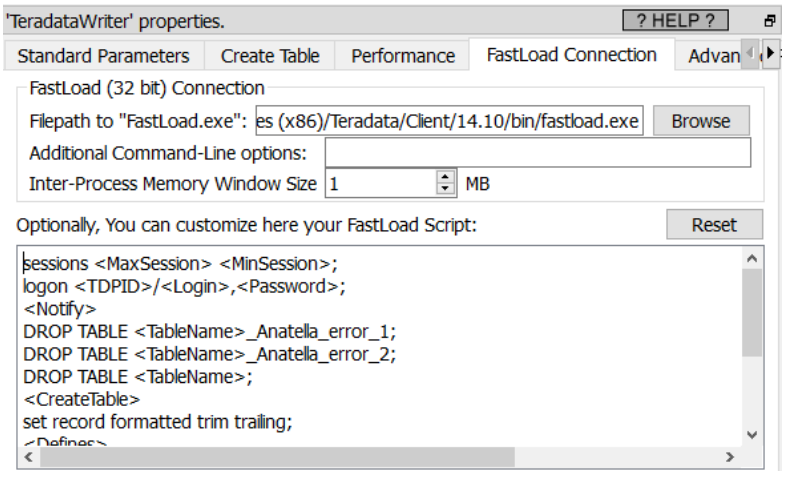
5.27.19. Teradata Writer

Icon:

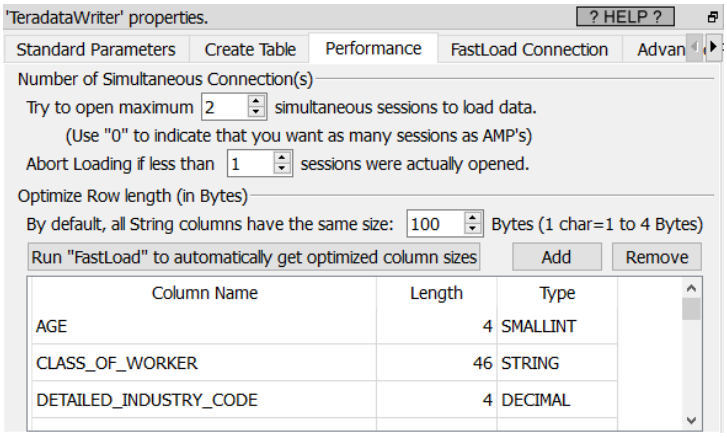
Property window:

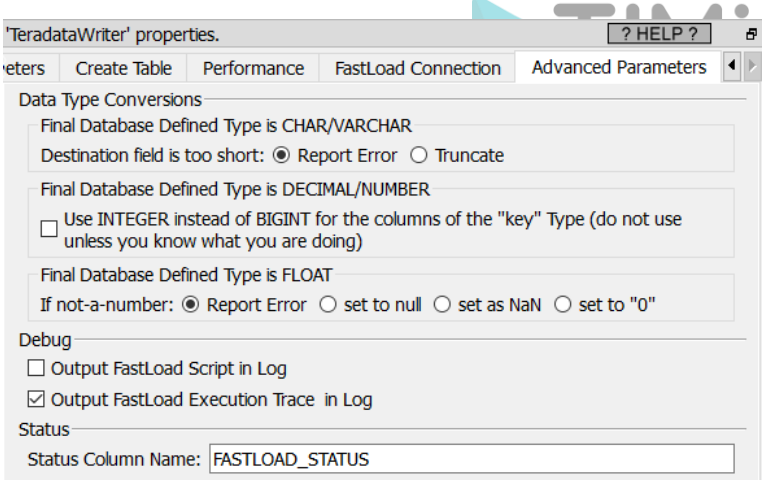






©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu





Short description:
Export Tables to Teradata at high speed

Long Description:

The Upsert Action (to insert rows into a database) is barely usable with Teradata because it's extremely slow. So, if you want to INSERT rows into Teradata, the best & fastest option is the TeradataWriter Action. Behing the scene, the TeradataWriter Action communicates with Teradata using the "FastLoad" Teradata utility. The "FastLoad" Teradata utility has some limitations:

- The "FastLoad" tool can only INSERT rows into an empty table.
- The "destination" table cannot have any INDEX.

If your destination table is non-empty or contains an INDEX, you can still use the TeradataWriter Action to INSERT rows at high speed into Teradata: More precisely, in such situation you'll use the TeradataWriter Action to INSERT rows into an empty intermediary table (without any INDEX) and thereafter copy the content of the intermediary table into the "final, destination" table using the following SQL command: "*INSERT INTO <Final_Destination_Table> SELECT * FROM <Intermediary Table>*". You'll find more details about this procedure in the paragraph named "*FAST INSERT: Solution 2*" from section 5.27.4.1.

The "Fastload" tool might seem too restrictive because it does not allow us to add rows into a non-empty table or into a table that has an INDEX: i.e. In such situations, "Fastload" is forcing us to use an intermediary table.

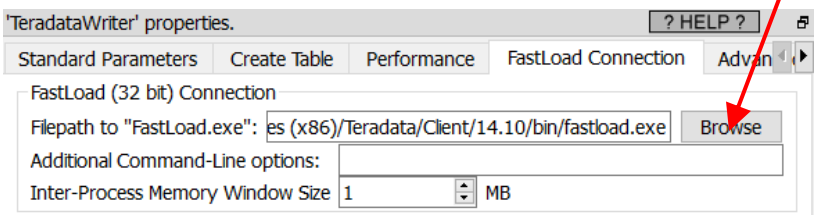
The reality is that we are ***anyway*** forced to use an intermediary table for other reasons than the limitation imposed by the "fastload" tool. These other reasons are:
* The INSERT operations inside a table with an INDEX are too slow (and a way to by-pass that very slow speed is to use an intermediary table: See the section 5.27.4.1 for more information about this subject).
* An intermediary table is required to have an INSERT procedure that is guaranteed to have no duplicated or lost rows when uploading rows inside a database (even in the case of a computer crash or power-off in the middle of the upload): See the section 10.10.4.2. for more information about this subject.

Since the limitations imposed by "fastload" are already imposed to us for other reasons, it means that, actually, "fastload" is not so restrictive after all and, 99% of the time, it's the best/fastest tool to use to INSERT rows into Teradata.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

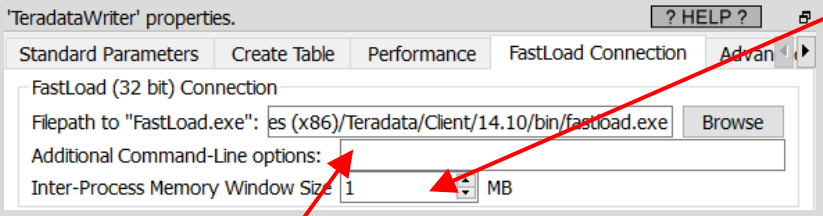
To guarantee the highest speed possible, the communication between Anatella and the "FastLoad" tool is 100% "in-memory" (in RAM memory): i.e. Anatella writes rows into an "in-memory" window and the "FastLoad" tool directly reads back these rows from memory to inject them at high speed into Teradata. No data is ever written on the hard drive (i.e. Anatella uses a high efficiency INMOD routine to exchange data with "FastLoad").

Before using the TeradataWriter Action for the first time, you must define the location of your "FastLoad.exe" executable. This is only required one time: i.e. the first time that you use the TeradataWriter Action on your machine: Open the parameter window of the TeradataWriter Action, go to the "FastLoad Connection" panel, click the "Browse" button: and locate the "fastload.exe" executable:



The "FastLoad" tool executes asynchroneously meaning that, at the same time Anatella is computing new rows, the "FastLoad" tool is sending to Teradata the rows that it already received (that are already stored in its internal buffer). This guarantees a maximum throughput because, if the data flow coming from Anatella is briefly reduced (e.g. because of a temporary higher load on the CPU), the "FastLoad" tool is still able to continue to send to Teradata at maximum speed (without any slow down) all the rows that are currently in its internal buffer.

This asynchronous mechanism (that uses an "internal buffer" that guarantees maximum throughput) is not part of the default distribution of "FastLoad": It's only available inside Anatella. The size of this "internal buffer" is defined here:

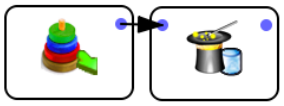


All the Text data sent to "Fastload" is in Unicode UTF-8: i.e. we run "fastload.exe" with the command-line parameters: "`-i UTF8 -c UTF8`" to force UTF-8 encoding. Optionally, you can add more user-defined command-line parameters here:

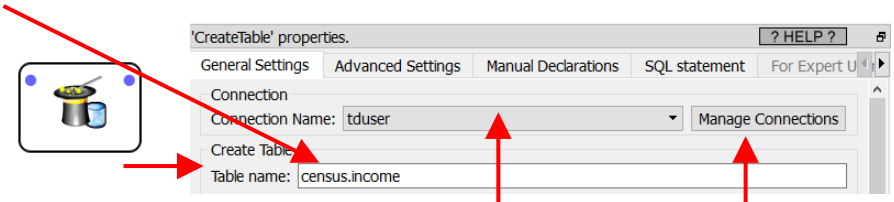
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When the "FastLoad" procedure fails in the middle of a "data upload" procedure, The Teradata engine completely locks the destination table and no access (reading or writing) to the "destination" table is possible anymore. The only solution to "unlock" the "destination" table is to DROP the table (and then, just after, re-CREATE it). Only once the "destination" table is "unlocked", you can write into it again. Fortunately for us, it is possible to directly execute inside the "FastLoad" tool a DROP statement and a CREATE TABLE statement before running any INSERT, so that we have the guarantee that the "destination" table will never be locked when we start INSERTing data into it. Thus, to have the guarantee that the destination table is always "unlocked", the default procedure used inside the TeradataWriter Action is always to start by executing a DROP statement and thereafter a CREATE TABLE statement (before any INSERT statement). This is why, inside the second panel of the the TeradataWriter Action, you must write the CREATE TABLE statement required to create your "destination" table. Hopefully, Anatella is helping you to write this CREATE TABLE statement: i.e. the procedure to easily get the CREATE TABLE statement is the following:

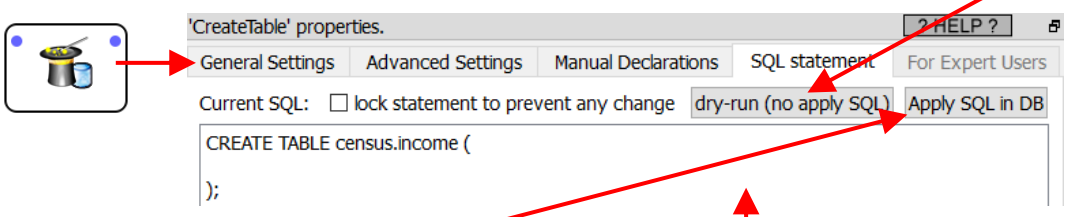
- **Optional Step:** Use the CreateTable Action to automatically generate the CREATE TABLE statement: see section 5.27.5. for more information about this subject. To summarize:
  - Connect your "source" Table to a CreateTable Action: For example:



  - Open the parameter window of the CreateTable Action and enter a Table Name:



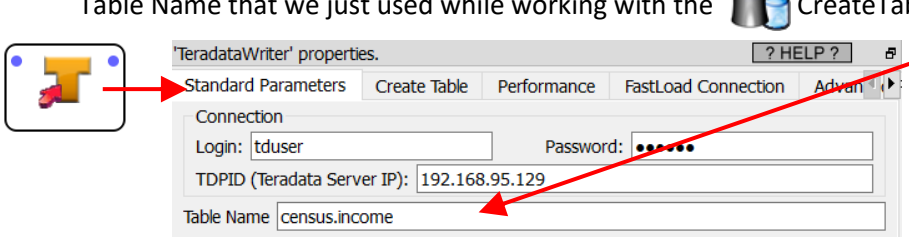
  - Setup your ODBC connection to Teradata: if there is already an ODBC connection available, you just have to select from here: . Otherwise, click on the "Manage Connections" button to create a new ODBC connection to Teradata: .
    See the sections 5.1.6. and 5.1.6.3. for more information about configuring an ODBC connection between Teradata and Anatella. Once we have finished configuring the TeradataWriter (i.e. once we get the CREATE STABLE statement), the ODBC connection is not required anymore.
  - Click the "Dry-Run" button inside the "SQL Statement" Panel (and wait a little):



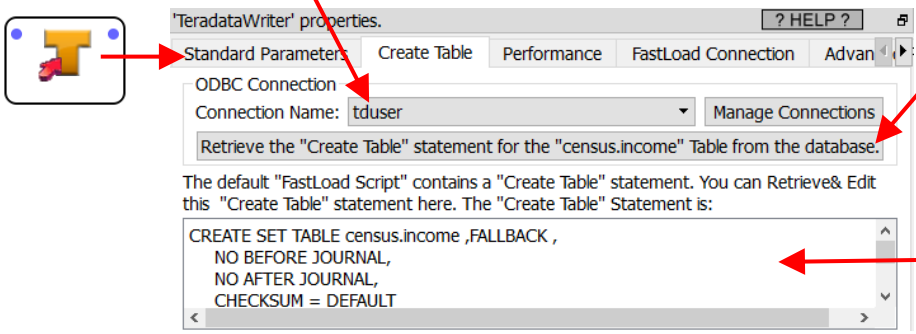
  - You should now see a CREATE TABLE statement here: . Validate this SQL statement: Click the "Apply SQL" button: . This should create your table inside your database. Fix any error in the sql CREATE TABLE statement until Teradata accept to create the desired table.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Open the parameter window of the TeradataWriter Action and enter here the same Table Name that we just used while working with the CreateTable Action:

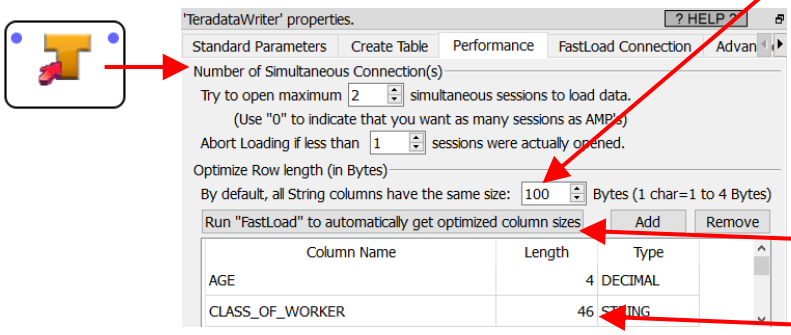


- Open the "Create Table" panel of the TeradataWriter Action and select the correct ODBC connection here: … and click the "Retrieve CREATE TABLE statement" button:



You should finally see the correct "CREATE TABLE" statement here.

The "FastLoad" tool is sending "data blocks" to Teradata using the computer network. Each "data block" contains several rows to INSERT. Each row is composed of several columns. All the columns have a fixed, user-defined size (i.e. Anatella presents its data to "Fastload" using the "Formatted Data" structure). By default, the size allocated for each cell of a column of the "String/Unknown" type is 100 bytes (i.e. from 25 to 100 characters): This "default" size is defined here:



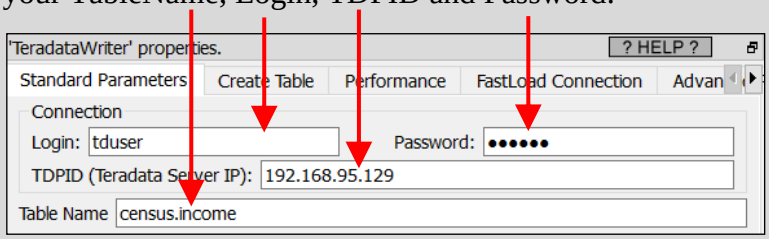
You can "overwrite" this default value, for some specific columns. For example, we defined here: that we'll use 46 bytes (instead of the default value of 100 bytes) to store the column "CLASS_OF_WORKER". For performance reasons, you should always use a storage space as little as possible to store all your columns (because this directly translates to less traffic on the computer network and thus higher throughtput on the computer network and, ultimately, higher INSERT speed into Teradata). To help you optimize the storage space of each column, there exists a specific "FastLoad" command (named "HELP TABLE") that tells you how much bytes each column is using inside Teradata. To automatically retrieve the results of the "HELP TABLE" command, click here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The button named "*Run FastLoad to automatically get optimized column size*" is actually executing the following FastLoad script:

```
logon <TDPID>/<Login>,<Password>;
help table <TableName>;
show;
logoff;
```

This means that this button will only work as expected if you entered previously your TableName, Login, TDPID and Password:



To get a higher INSERT speed into Teradata, Anatella optimizes (minimizes) the storage of each column by considering:

(1) the Type and Length of each column inside Teradata
(2) the type of each column inside Anatella

Here are some examples:

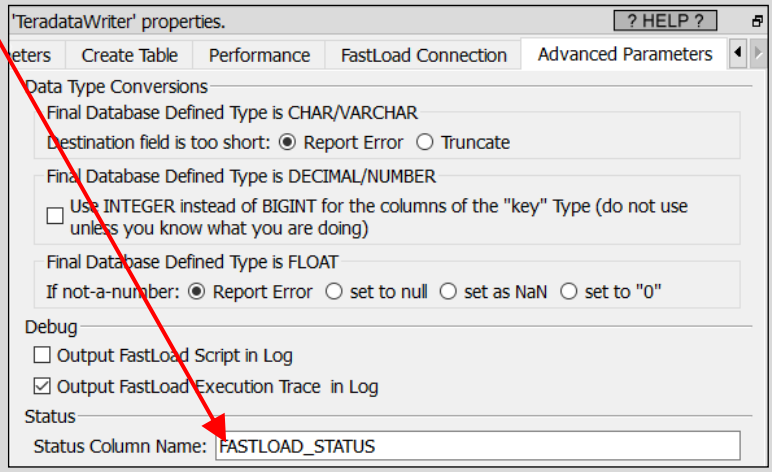
- A column that was declared inside Teradata as an INTEGER with 1 to to 4 digits (i.e. a column that was declared inside the CREATE TABLE statement as "DEC(4)") may contain integer numbers from -9999 to 9999. Thus, Anatella will inject this column into "fastload.exe" using only 2 bytes per cell (because 2 bytes can contain all ingerer numbers from -32768 to 32767).
- A column that was declared inside Teradata as a high-precision Floating point number (i.e. declared as "FLOAT(53)") and stored inside Anatella as text will be converted to binary representation before being injected into "FastLoad" (because, then, it only uses only 8 bytes of storage).

Error Handling

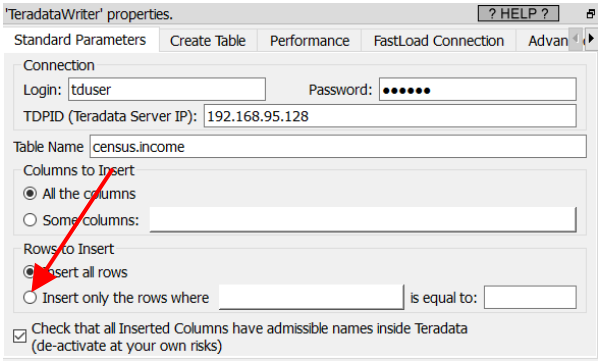
There exist many reasons why one row could not be inserted into Teradata. Anatella tries to detect in advance all the rows that will "fail": These "failing" rows are NOT sent to "FastLoad". Furthermore, these "failing" rows receives a special STATUS as output of the TeradataWriter Action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can select the name of the output column containing the STATUS of each row here:



The Satus codes visible inside the status output column of the TeradataWriter Action are:

Status Code	Comment
0	No error detected: Teradata should insert this row inside the destination table.
1	No error detected: The row was simply "skipped" because of this option: 
2	The source string has too many characters for the declared width of the destination column
3	The source number has too many digits for the declared width of the destination column
4	The source column contains non-numbers characters (but it was declared as a DECIMAL/NUMBER inside Teradata)
5	The source number is too big ($x > 2^{31}$) for the declared type (INTEGER) inside Teradata
6	The source number is too big ($x > 127$) for the declared type (BYTEINT) inside Teradata
7	The source number is too big ($x > 32767$) for the declared type (SMALLINT) inside Teradata
8	The source column is not a floating-point number (but it was declared as FLOAT inside Teradata)

There might still be a small quantity of rows, out of all the rows that were sent to Teradata, that were not properly INSERTed into the destination table. At the end of the "FastLoad" procedure, Teradata initializes 2 tables that contains more information about the rows that were not INSERTed. By default, these two tables are named "<TableName>_Anatella_error_1" and "<TableName>_Anatella_error_2".

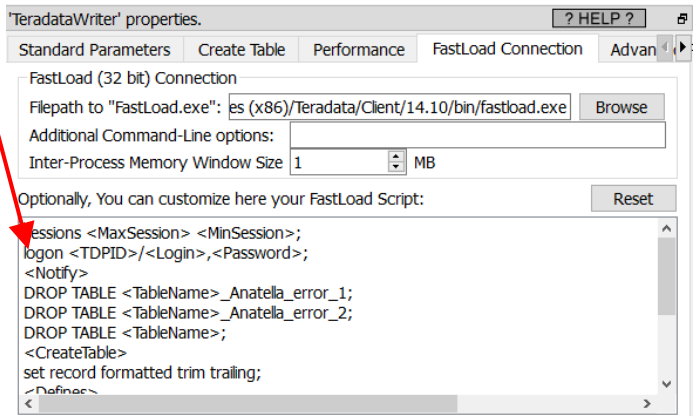
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Thus, after each "FastLoad" procedure, to have a complete look at all the possible errors, you should look at:

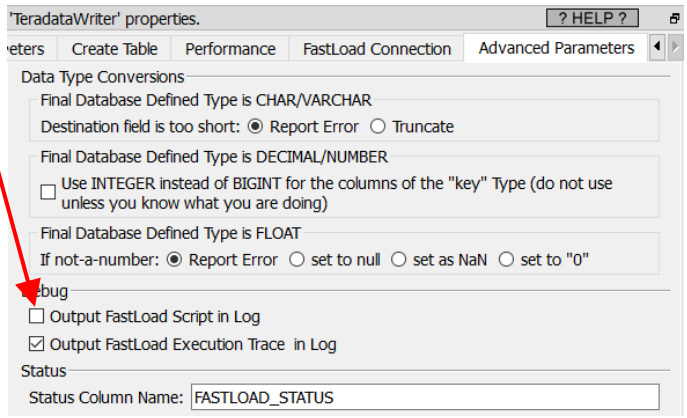
- …the Rows with a STATUS code that is above 1 (as output of the TeradataWriter Action).
- …the two tables named "<TableName>_Anatella_error_1" and "<TableName>_Anatella_error_2" inside Teradata (e.g. using the readODB action, execute the following SQL statement: "*SELECT * FROM "<TableName>_Anatella_error_1*")

Customizing the FastLoad script.

The "default" FastLoad script can handle almost all situations. Nevertheless, you can still tweak the FastLoad script template here, just in case:



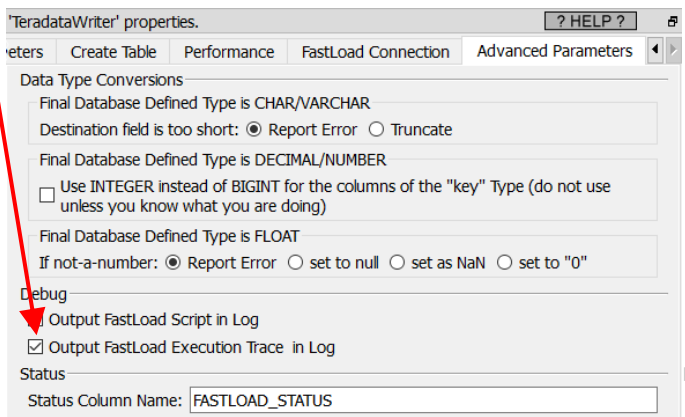
To help you to create your own fastload script, you can start from the script that is auto-generated by Anatella at each run: To see this script, click here:



Debugging Various FastLoad Errors

When "FastLoad" is working, it's very fast. When it's not working, it usually means that it's "stuck" somewhere waiting for a specific user-input. If nothing moves after a few seconds, it usually means that "FastLoad" is exhibiting an unexpected behavior (that is not handled by the default "FastLoad" template). To understand why "FastLoad" got "stuck", we need to look at the content of the console where "FastLoad" was executed: To do so:

1. Click here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

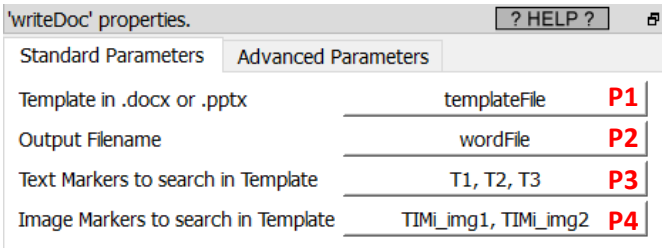
2. Run the TeradataWriter Action normally.
3. Wait a little until you see that nothing moves anymore.
4. Click the button on the Anatella toolbar: This has two effects:
    - It stops the "FastLoad" process.
    - It displays the whole execution trace inside the Log Window.

Unfortunately, you cannot see the "FastLoad" execution trace before clicking the button on the Anatella toolbar.

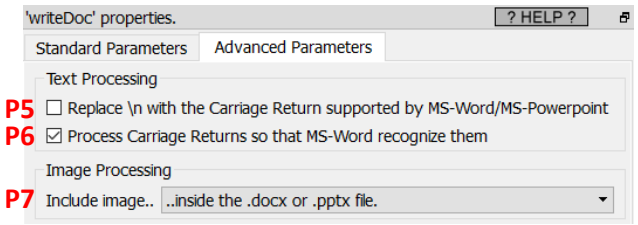
5.27.20. Write in Docx and Pptx

Icon:

Property window:



Short description:
Process a MS-Word template or a MS-PowerPoint template to create a serie of personalized documents.



Long Description:
For each input row, this Action will create a new .docx or .pptx document. This new document is a copy of the "template" document where:

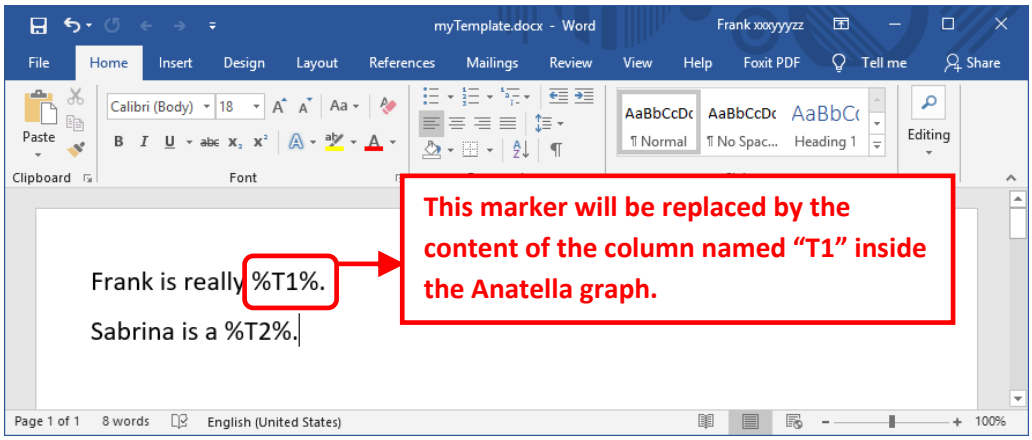
- …all the "Text Markers" have been replaced by the values found on the current row. Inside the MS-Word template or MS-Powerpoint documents, the markers are the strings starting and ending with the "%" characters. The marker's names must match the column's names given in the parameter **P3**.
  You'll find a complete example of .docx file processing with "Text Markers" inside the next section 5.27.20.1.
  The .docx template document must be "prepared" following the instructions given inside the section 5.27.20.2.
- …all the "Image Markers" have been replaced by other images. Only the .png, .jpg or .jpeg image-file-formats are supported. The filepath of these other images are on the current row. The "Image Markers" are image-files that:
    - …have with a specific name (typically: TIMi_img1, TIMi_img2, TIMi_img3, etc.)
    - …have been inserted inside the .docx/.pptx document using a very specific procedure: see the detail of this procedure inside the section 5.27.20.3.

  You'll find a complete example of .docx file processing with "Image Markers" inside the section 5.27.20.4.

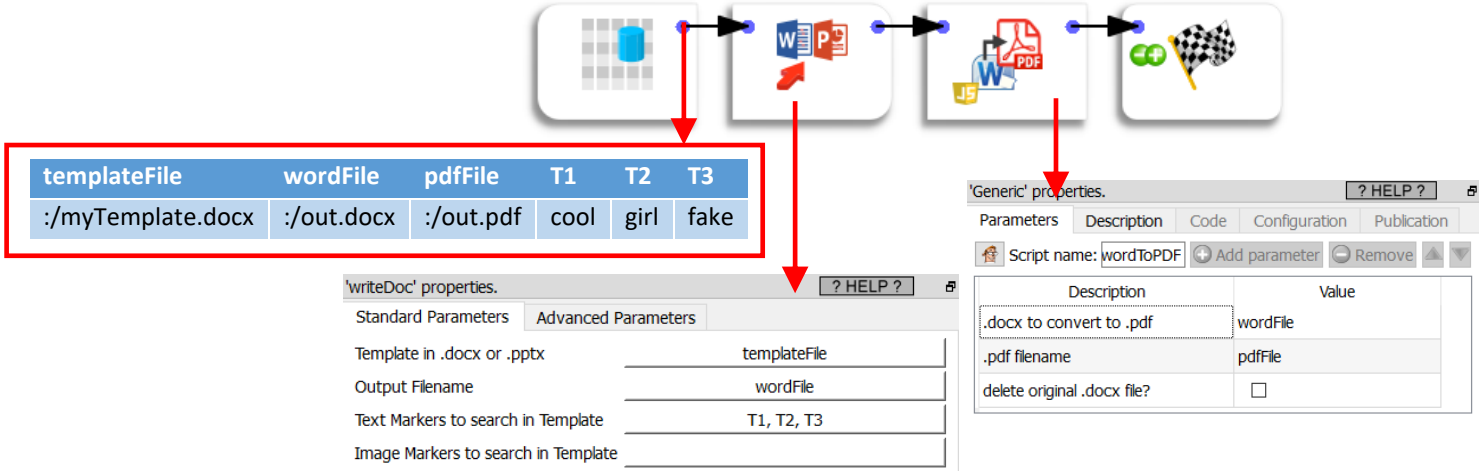
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.27.20.1. Example of processing for "Text Markers"

Let's give an example of processing for "Text Markers": Let's assume that we have the following template named "myTemplate.docx":



We run this Graph:



templateFile	wordFile	pdfFile	T1	T2	T3
:/myTemplate.docx	:/out.docx	:/out.pdf	cool	girl	fake

This will produce a MS-Word file named "out.docx" (and a PDF file named "out.pdf") that contains:

> Frank is really cool.
> Sabrina is a girl.

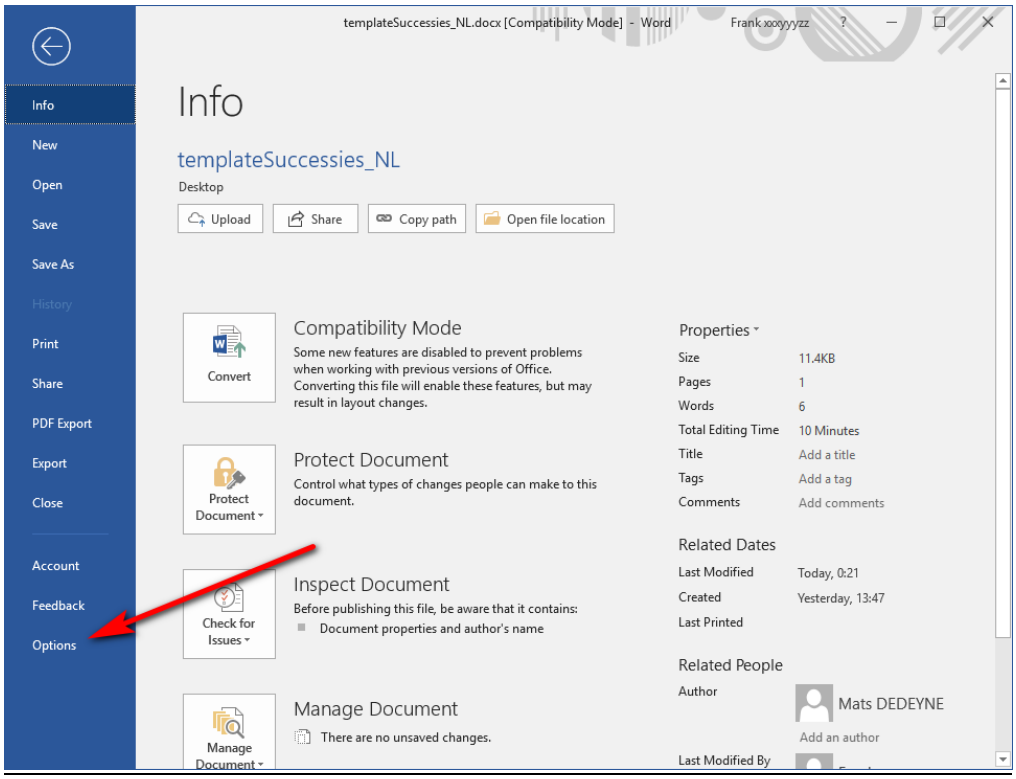
5.27.20.2. MS-Word Template pre-processing.

New versions of MS-Word are adding some invisible tags inside the .docx documents. These invisible tags might prevent the "marker substitution process" to work. Hopefully, there is a way to remove all these invisible tags and have a "template" that is cleaned and working properly. The process to clean your template .docx files is in 3 steps:

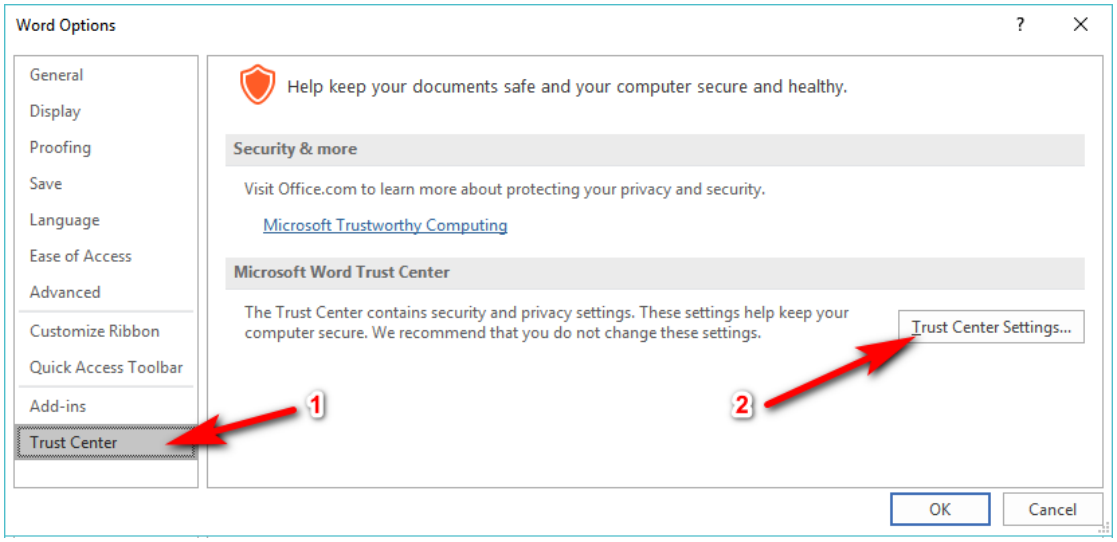
Step 1 (get rid of the "w:rsidR" tags).

1. Open your ".docx" template document in MS-Word.
2. Click on the "File" menu and click on the "Option" menu:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

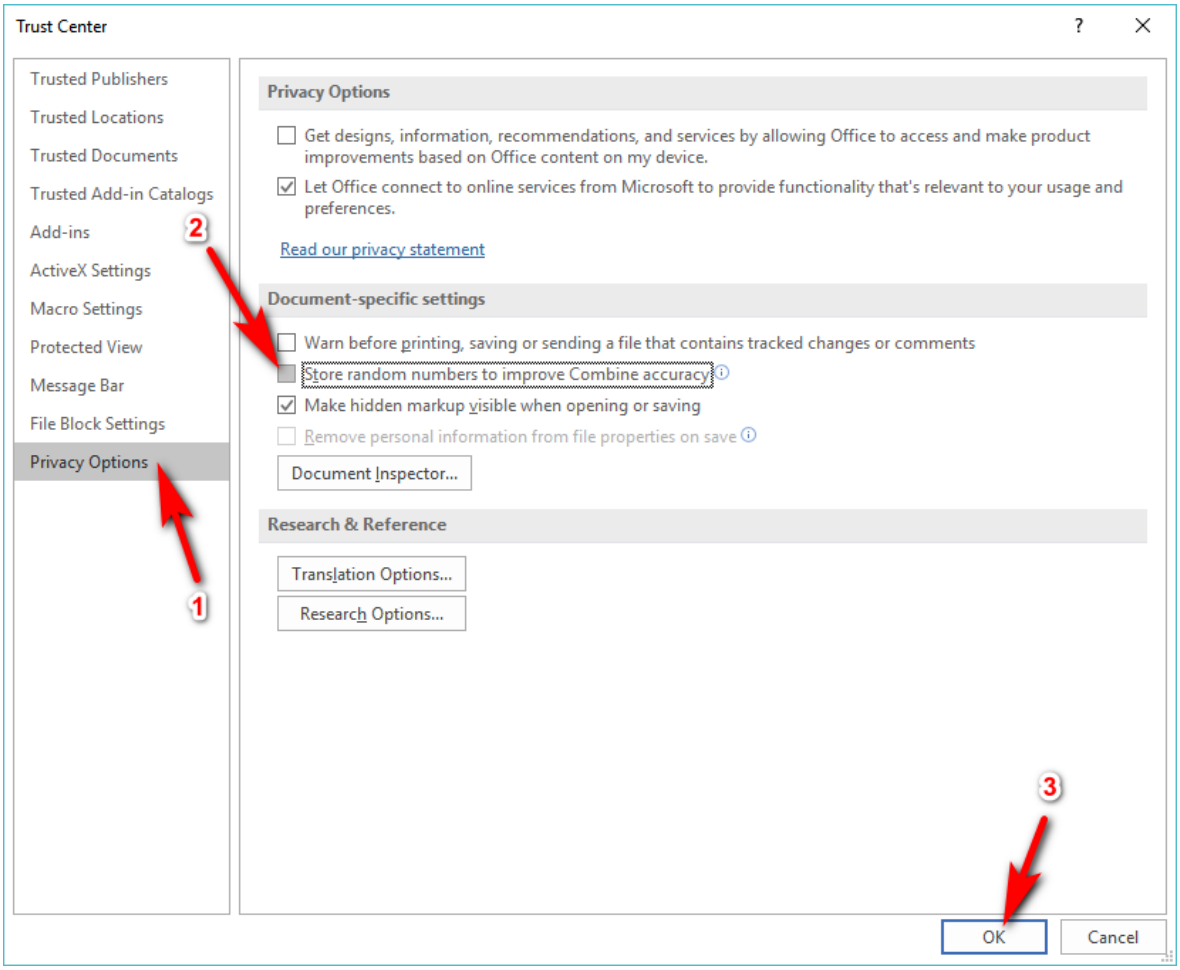


3. Click on the "Trust Center" option and on the "Trust Center Settings" button:



4. Click on the "*Privacy Options*", uncheck the checkbox "*Store random numbers to improve combine accuracy*". Then, click the "OK" button at the bottom to confirm:

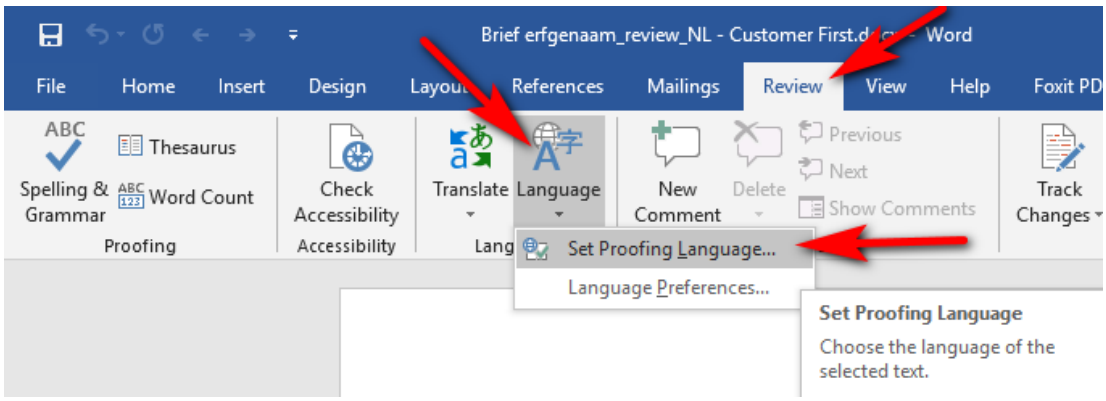
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



5. Close the "Word Options" window (click the "Ok" button) and save the .docx file (press [CTRL]+[S]).

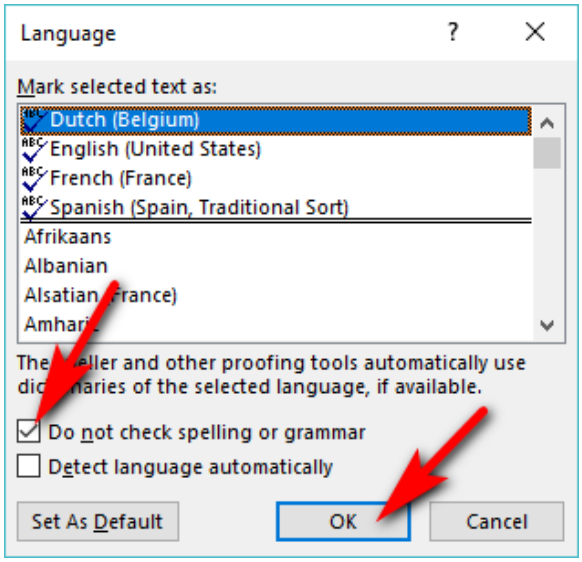
Step 2 (get rid of the auto-correct/proofing tags).

1. Open your ".docx" template document in MS-Word.
2. Select the whole document: Press the keys [CTRL]+[A] on your keyboard.
3. Go to the "Review" tab, select "Language" -> "Set Proofing Language":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

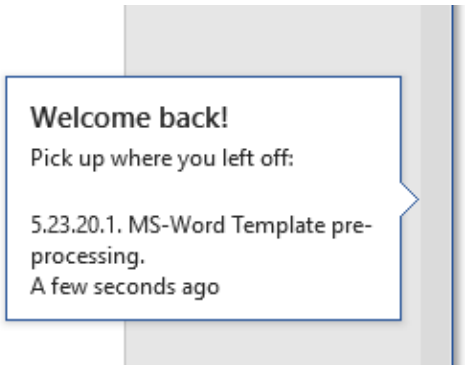
4. Check the checkbox named "*Do not check spelling or grammar*" and click the "OK" button:



5. Save the .docx document (i.e. press [CTRL]+[s]).

Step 3 (get rid of the "_goback" bookmarck)

When you open a .docx focument for the second time, MS-Word offers you to "go back" where you made your last change inside the document: i.e. You see on your right scroll-bar something like:



To offer this functionality, MS-Word insert an invisible "_goback" bookmarck tag at the exact location of your last edition. If this "_goback" tag is in the middle of an "Anatella Marker" (to replace), then Anatella won't be able to process the marker.

The solution is simple:

1. Open your ".docx" template document in MS-Word.
2. Move your cursor outside of any "Anatella Marker" (e.g. press the [CTRL]+[End] keys on your keyboard) and do a small edition (e.g. add a space character and remove it just afterward).
3. Save the .docx document (i.e. press [CTRL]+[s]).

5.27.20.3. Insertion of "Image Markers" inside .docx and .pptx documents

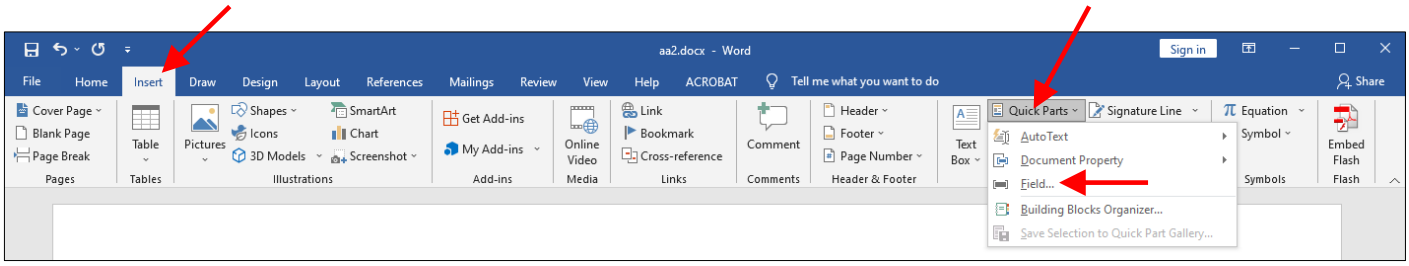
Anatella automatically replaces a serie of specific images inside your .docx/.pptx template documents. These images are your "Image Markers". These "marker images" have specific filenames that must

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

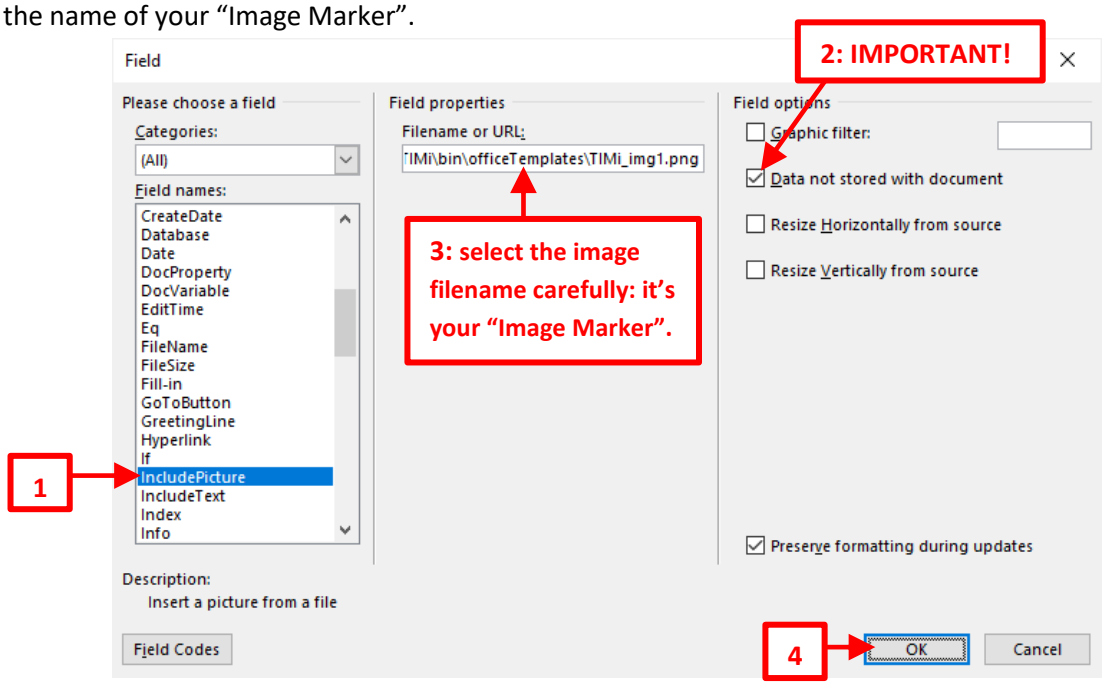
match the names given inside the parameter **P4** of the writeDoc action. These “marker images” must be inserted inside your document following a very specific procedure:

Insertion of “Marker Images” inside a .docx document

1. Open your .docx document with MS-Word and position your cursor where you want to insert your “marker image”.
2. Inside the “Insert” toolbar in MS-Word, click on “Quick Parts”, and select “Field”:

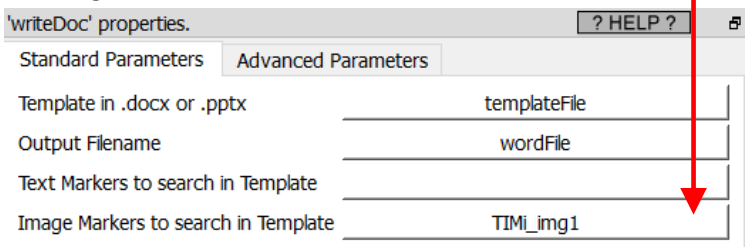


3. Inside the list on the left column, select “IncludePicture”.
   IMPORTANT: Inside the “Field options” on the right column, check the checkbox named “Data not stored with document”.
   Inside Filename parameter the middle column, select an image filepath. The filename of this image is the name of your “Image Marker”.

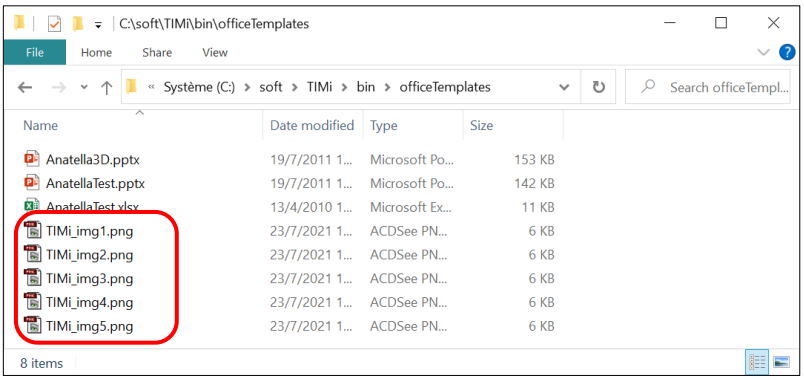


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

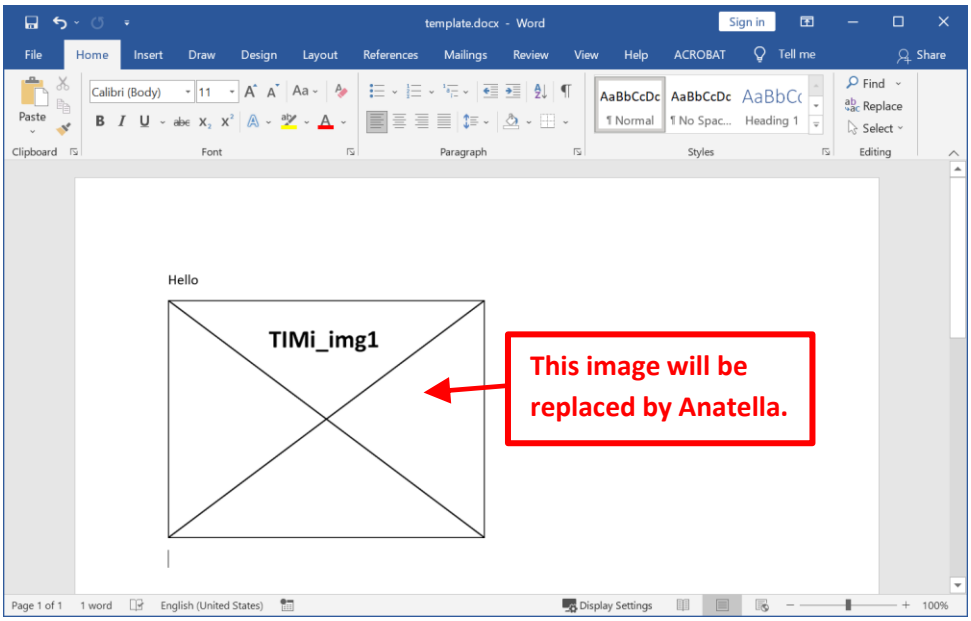
In the example above we choosed (as our image filepath), the filepath "C:\soft\TIMi\bin\officeTemplates\TIMi_img1.png". This means that the name of the "Image Marker" related to this image is "TIMi_img1" and we need to set "TIMi_img1" inside the parameter **P4** here so that Anatella is able to change this image.



You'll find a prepared set of images (with meaningful filenames) to help you design your templates inside the directory "<TIMi_install_dir>/bin/officeTemplates" (at this point, you can really use any image content and any image filename but these images might still help you):



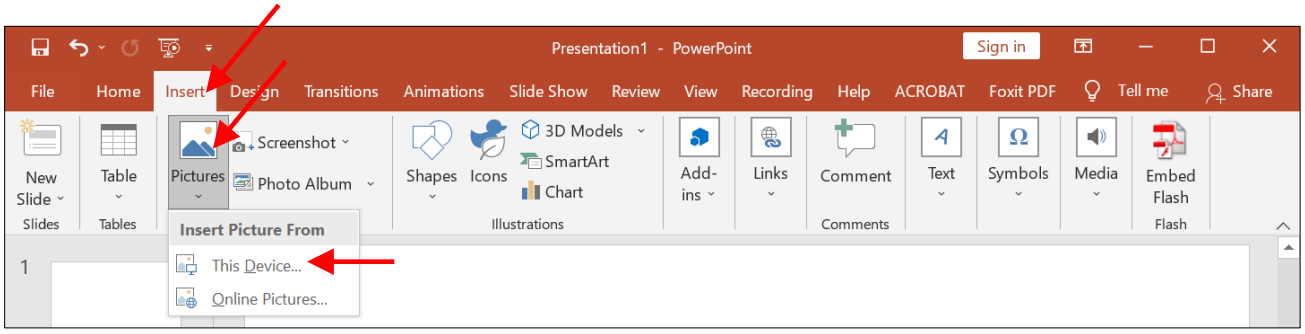
4. We now see inside MS-Word:



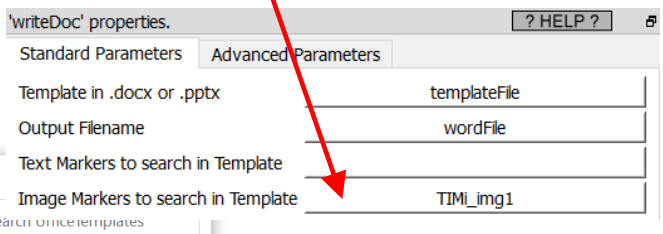
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

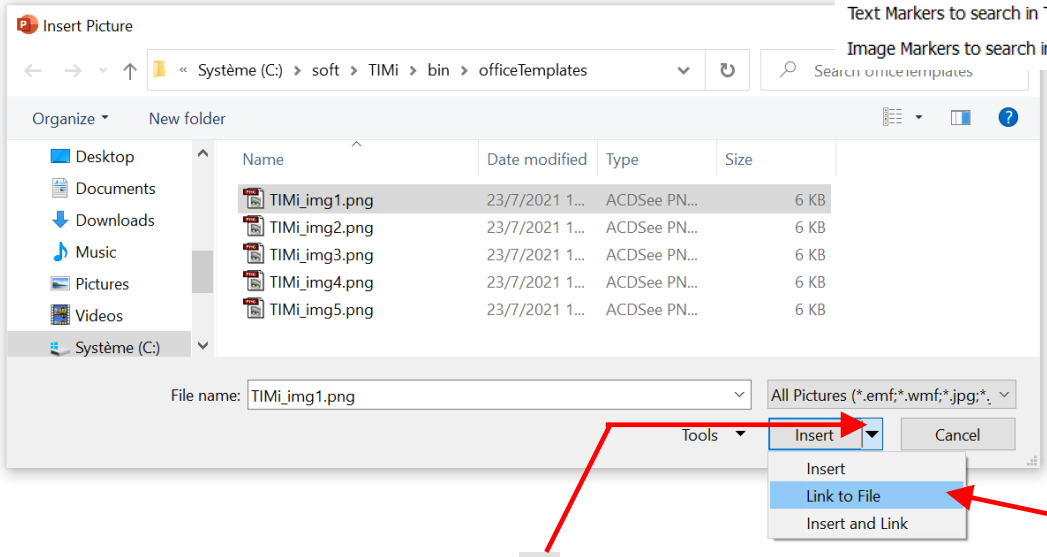
Insertion of “Marker Images” inside a .pptx document

1. Open your .pptx document with MS-PowerPoint and go to the slide where you want to insert your “marker image”.
2. Inside the “Insert” toolbar in MS-PowerPoint, click on “Pictures”, and select “This Device…”:

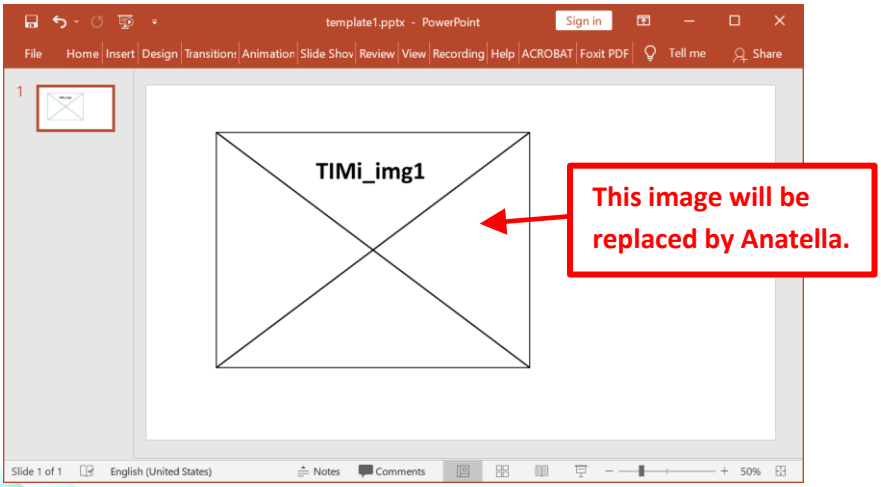


3. Inside the file browser window, select an image file. The filename of this image is the name of your “Image Marker”. In the example here below, the “Image Marker” is “TIMi_img1”. This means that we need to set “TIMi_img1” inside the parameter **P4** here so that Anatella is able to change this image.



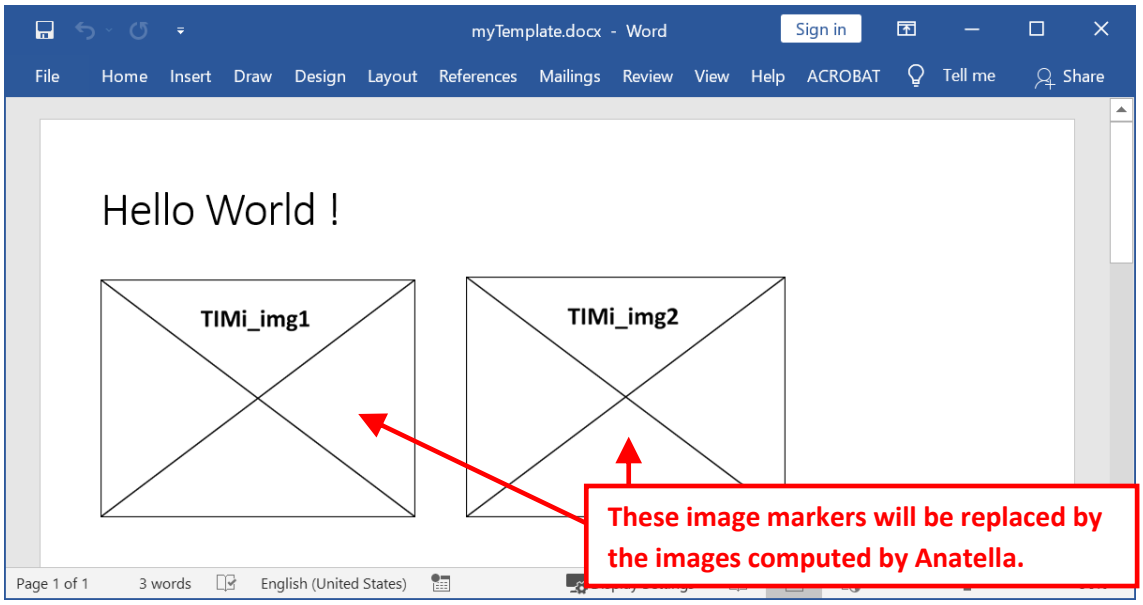


4. **IMPORTANT:** Click on the ▼ arrow (inside the “insert” button) and select “Link to File”. Do not simply click the “Insert” button directly: you must click on “Link to File” sub-option.
5. We now see inside MS-PowerPoint:

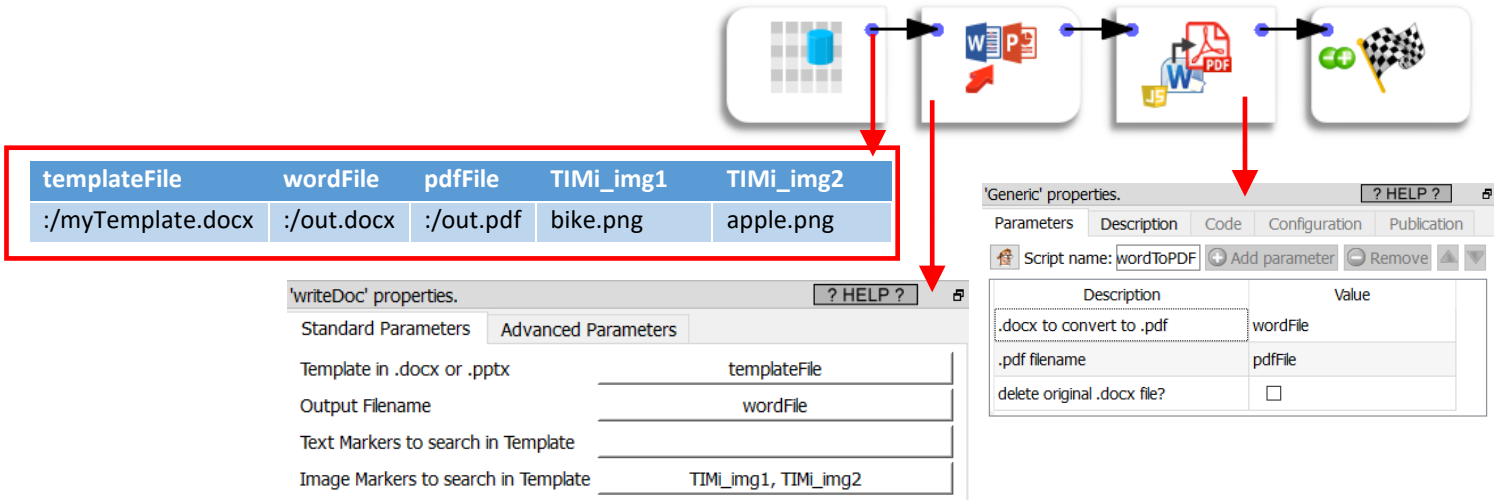


5.27.20.4. Example of processing for "Image Markers"

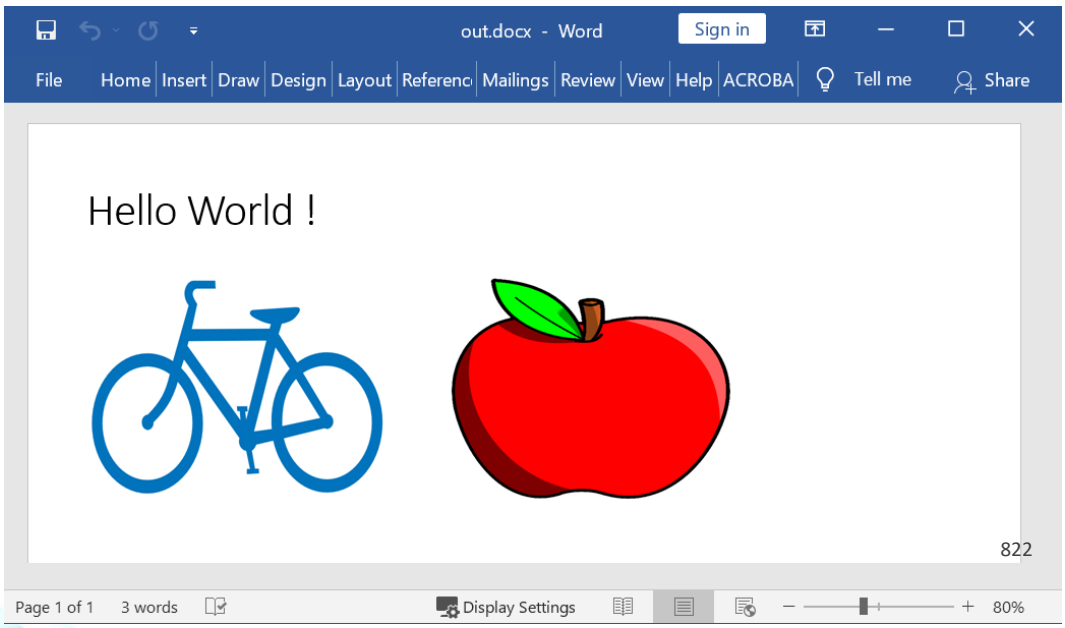
Let's give an example of processing for "Image Markers": Let's assume that we have the following template named "myTemplate.docx":



We run this Graph:

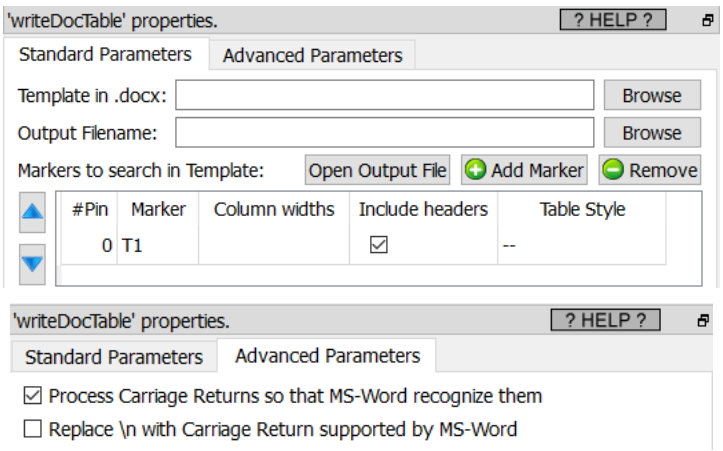


This will produce a MS-Word file named "out.docx" (and a PDF file named "out.pdf") that contains:



5.27.21. Write Tables in Docx

Icon:



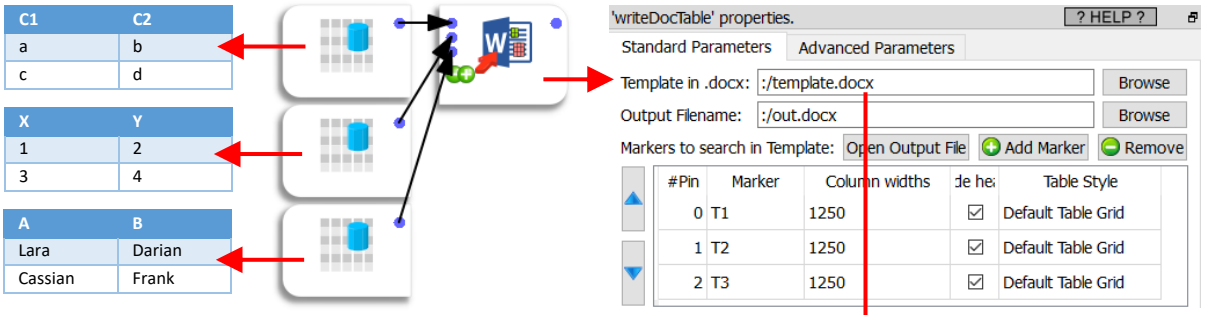
Property window:

Short description:
Inject a data table inside a MS-Word template document with Table-Style support!

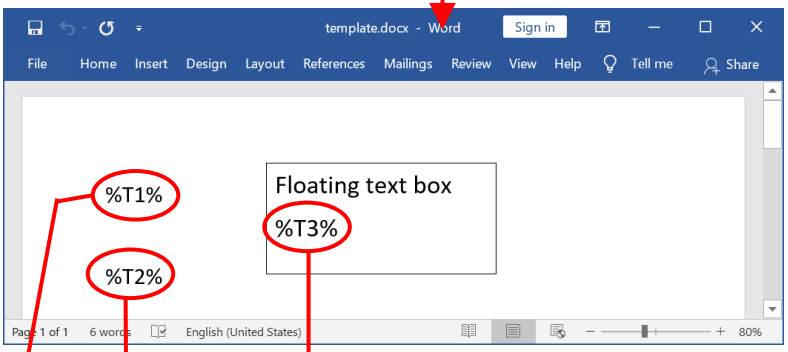
Long Description:
Anatella searches for the different markers and replaces them with the tables given in the input pins.

New versions of MS-Word are adding some invisible tags inside the .docx documents. These invisible tags might prevent the "marker substitution process" to work. Please refer to the section 5.27.20.2. for the procedure to get a "clean" template document (without all these invisible tags).

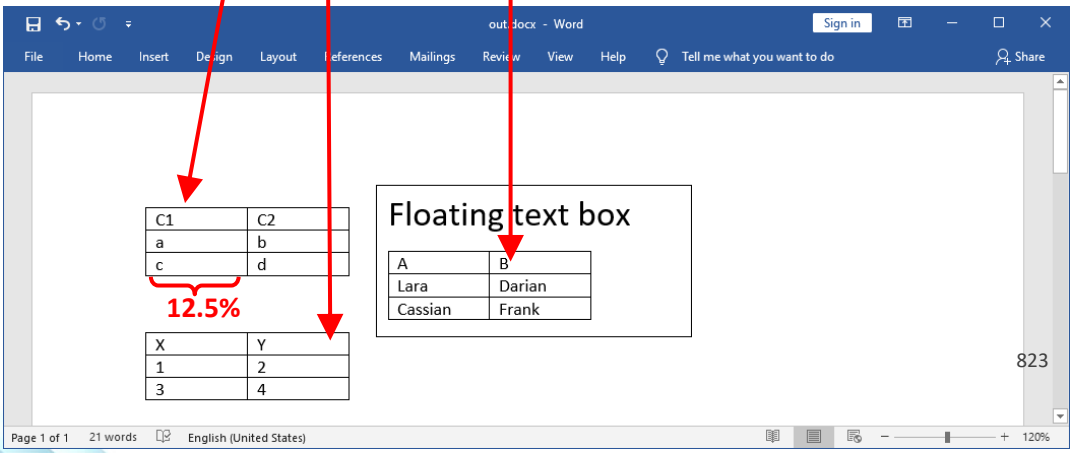
Here is an example:



The file "Template.docx" is:



The output file "out.docx" is:

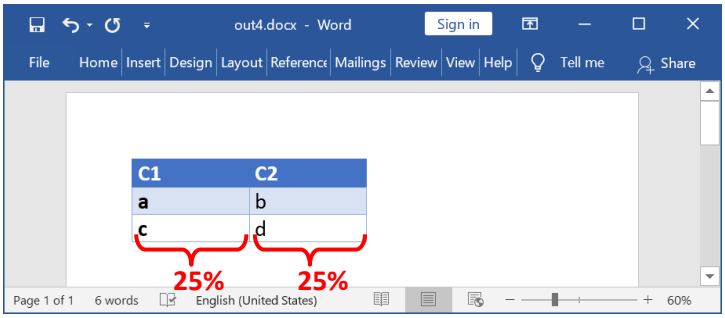


The parameter “column width” has 3 operating modes:

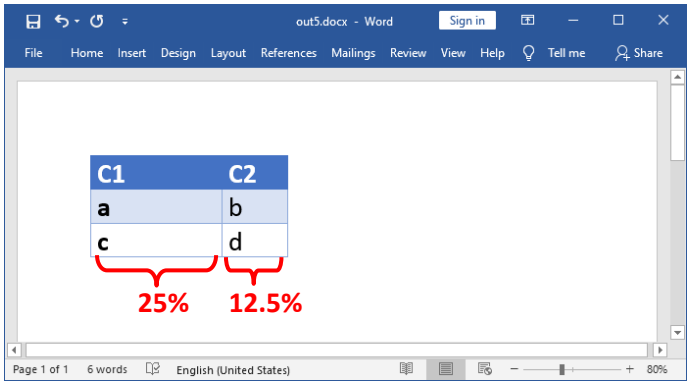
- **Operating mode 1:** The “column width” parameter is empty: the width of the injected Table inside the MS-Word document is 100% of the available width.
- **Operating mode 2:** you only give one number. This number defines the width of all of the columns inside the Table inside the MS-Word document.
- **Operating mode 3:** you give a list of comma-separated numbers. These numbers define the width of each of the columns inside the Table inside the MS-Word document.

The length-unit used inside the “column width” parameter is the *“% multiplied by 100”* (i.e. the “per-ten-thousands”). For example:

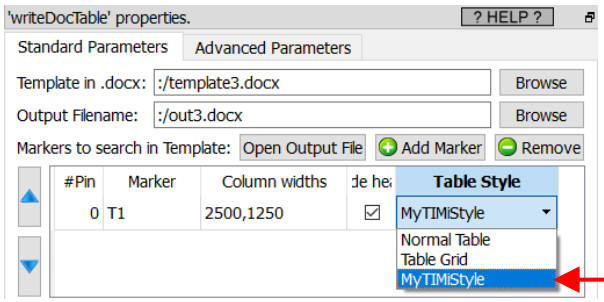
Column Width = 2500



Column Width = 2500, 1250



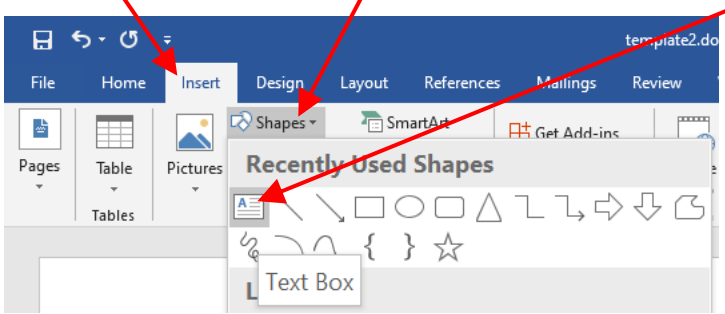
The “Table Style” parameter is a combo box that only works after you have defined the “Template” parameter. Once the “Template” parameter is defined, Anatella analyzes it and displays all the “Table Style” available inside the “Template” file: In the example below, the “template3.docx” document contains 3 “table styles”:



The “Normal Table” and “Table Grid” table-styles are (almost) always defined. The “template3.docx” document also contains one additional extra&custom table-style named “MyTIMiStyle”.

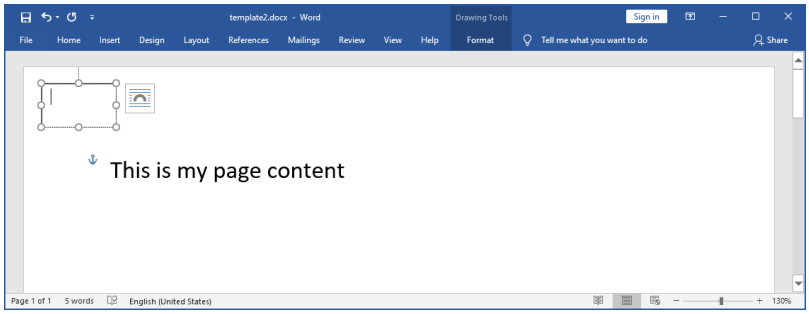
Let’s now see how we can define our own table styles! The steps are:

1. Create a 1x1 table (inside a floating text box):
   1.1. Insert a floating “text box” in the margin of the page (in this way you can hide it easily later). Open the “Insert” panel, click on “Shapes”, click on the “Text box” icon:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

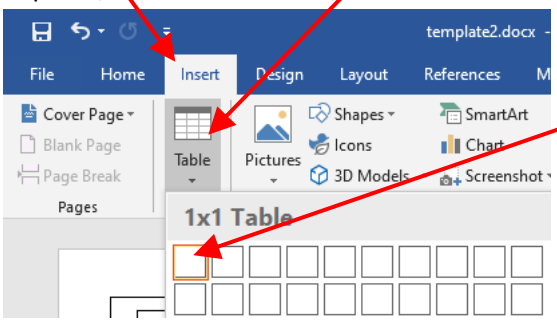
1.2. Draw your floating text box in the margin of the page:



1.3. Insert a 1x1 "table" inside your new floating text box.

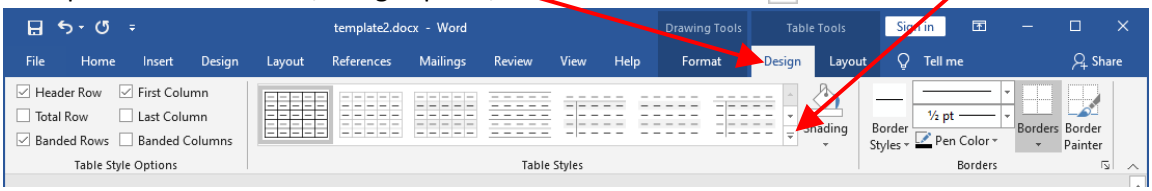
Click inside your text box.

Open the "Insert" panel, click on "Table", click on the cell that represent the "1x1" table:

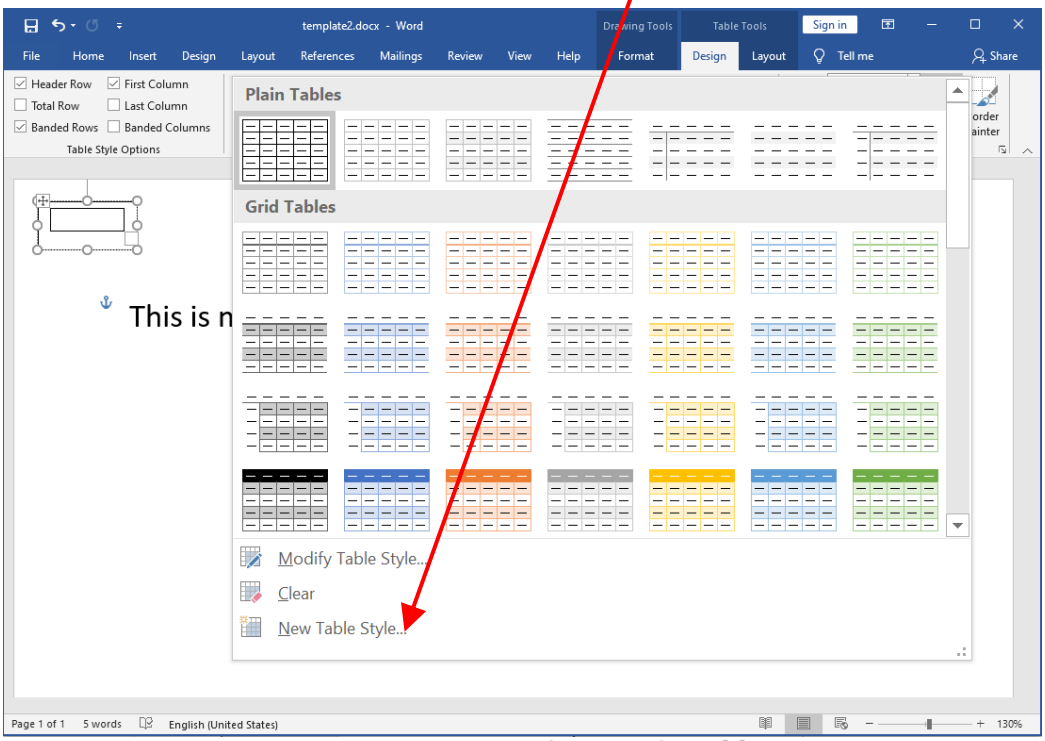


2. Define a new style for your new 1x1 table.

2.1. Open the "Table Tools/Design" panel, click on the "More" button here:

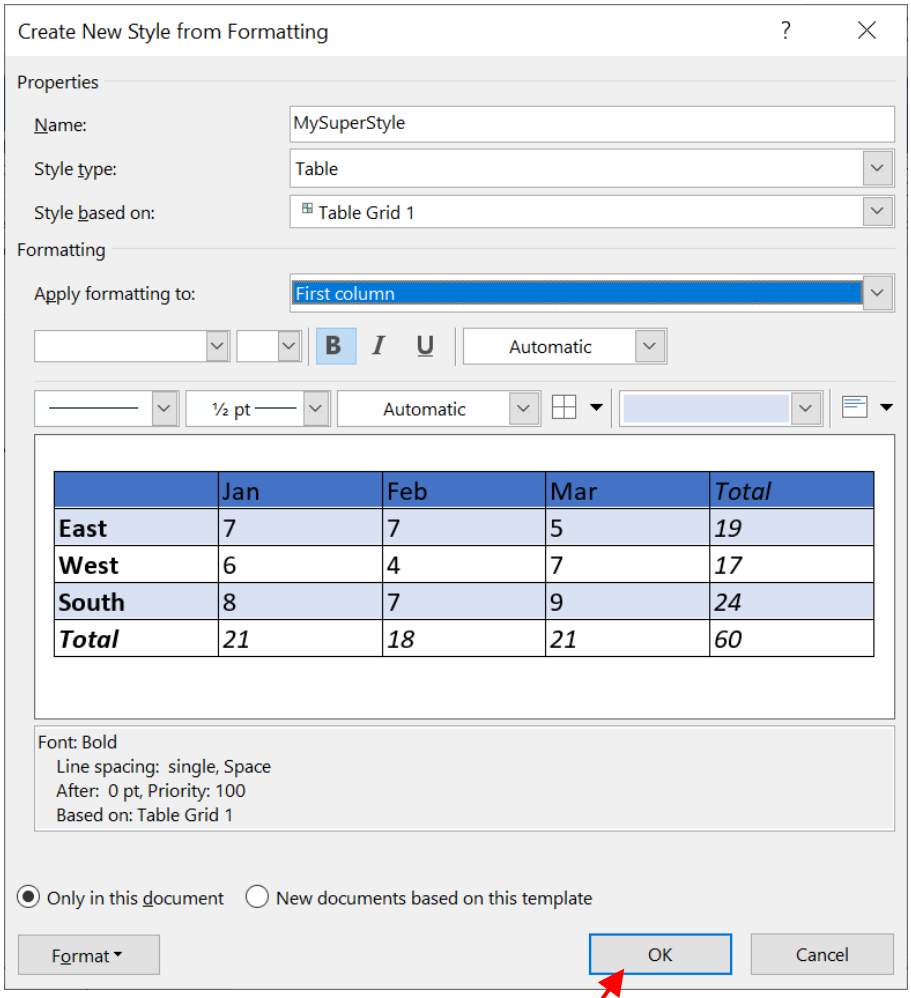


2.2. Inside the drop-down-meu, select "New Table Style":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2.3. Define a new Table Style: For example:



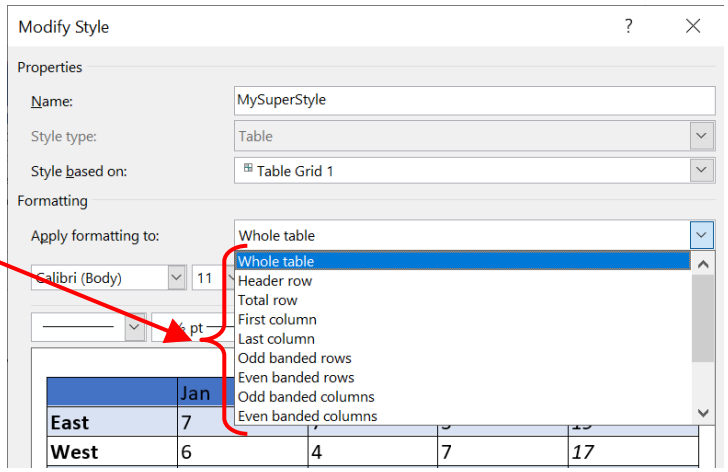
The above new Style is named "MySuperStyle", it contains the following settings:

- Header Row: with a "Dark Blue" background
- Odd banded Row: with a "Light Blue" background
- First column: font in Bold

Once finished, click the "OK" button:

You can define many different styling options for :

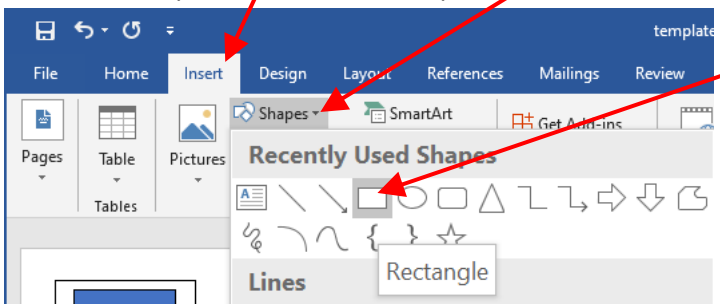
- The header row
- The total (last) row
- The First column
- The Last Column
- Odd banded rows
- Even banded rows
- Odd banded columns
- Even banded columns



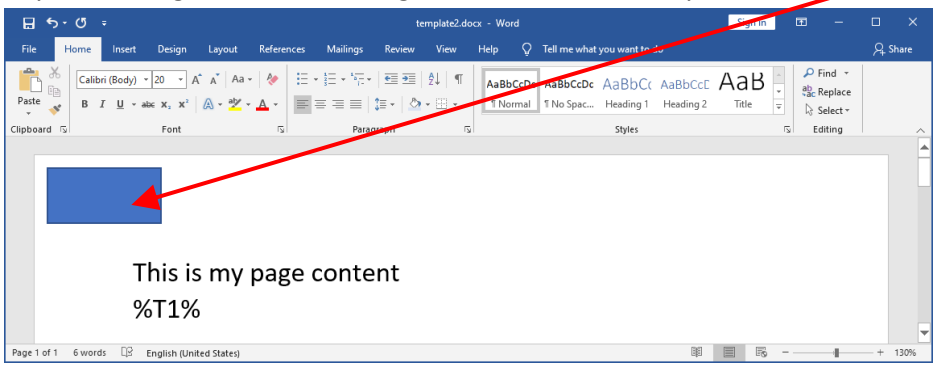
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Hide the small 1x1 table (i.e. cover it with a white rectangle)

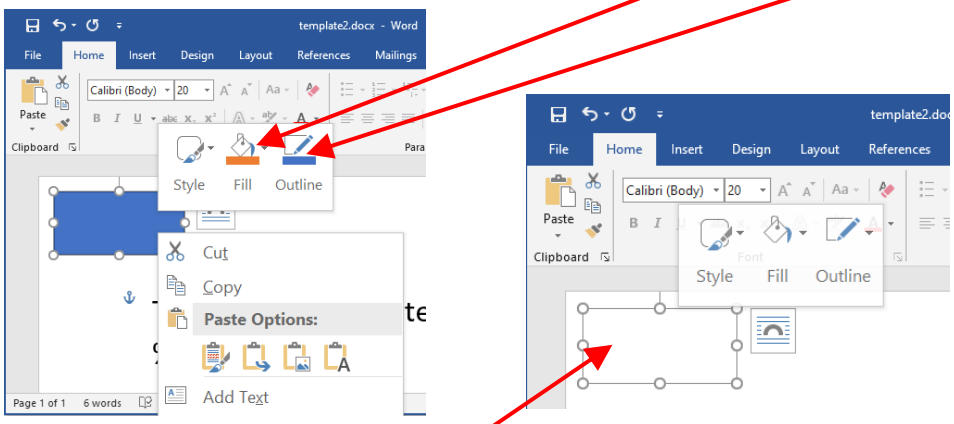
1.1. Open the “Insert” panel, click on “Shapes”, click on the “Rectangle” ▭ icon:



1.2. Draw your rectangle over the floating text box that contains your table:



1.3. Change the background color and the outline color of your new rectangle to white: Right-click your new rectangle and select the white color for the “Fill” and “Outline” option:

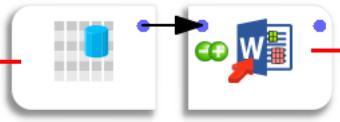


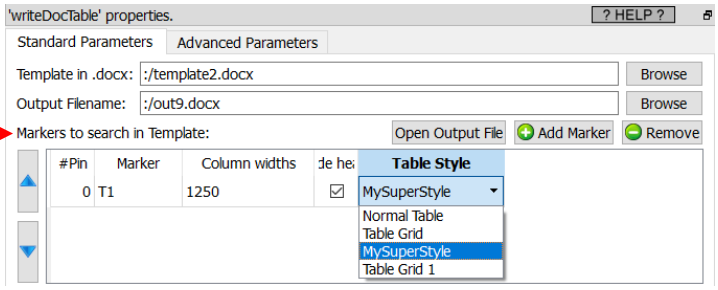
At the end, you get something like this:

2. Save your .docx document: press [CTRL]+[S].

Let’s use our new style!

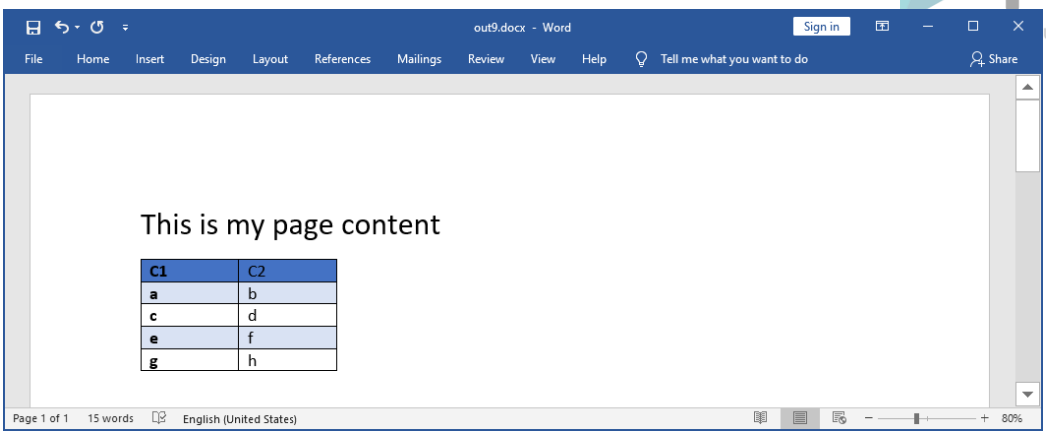
C1	C2
a	b
c	d
e	g
g	h





©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

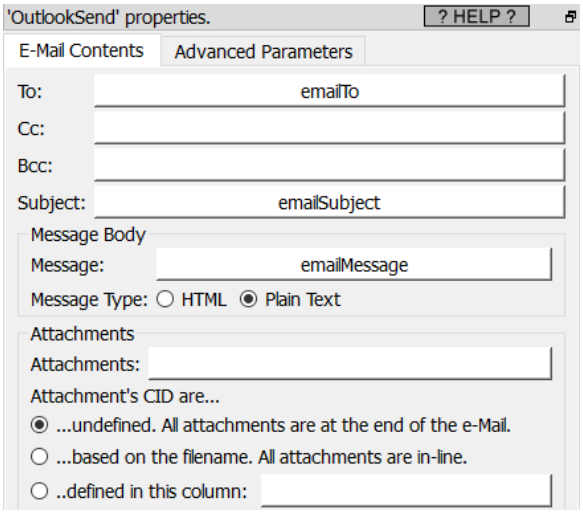
We obtain:

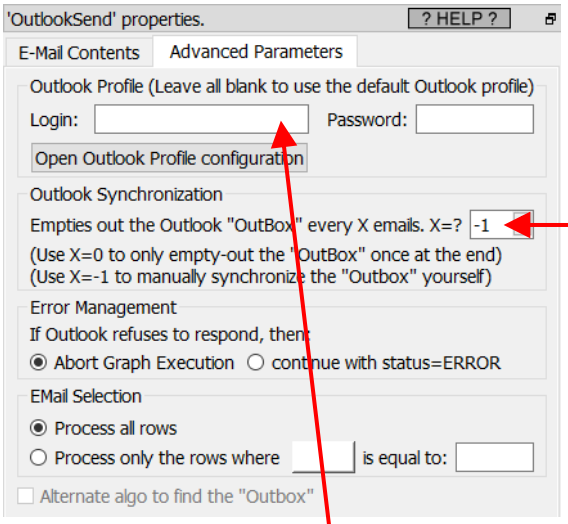


5.27.22. Outlook Send

Icon:

Property window:





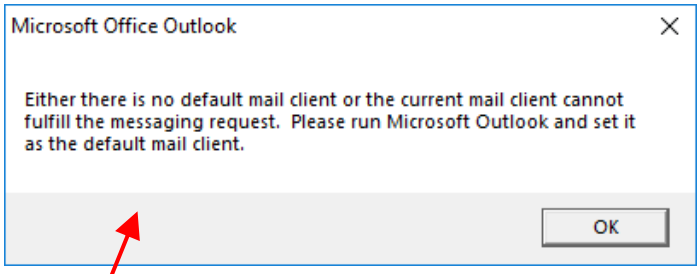
Short description:
Send Emails using Outlook.

Long Description:
All your emails and settings inside Outlook are stored inside an “Outlook Profile”. Before sending or receiving any email, you must select the right “Outlook Profile” using this parameter: see the section 5.2.24.1. for more details about how to select the right “Outlook Profile”.

To send E-mails via Outlook, Anatella simply adds emails inside the “Outbox” of MS-Outlook. To prevent that the “Outbox” becomes really (too) big, you can “empy out” the MS-Outlook “Outbox” every time that X emails are added inside the “Outbox”. The value “X” is defined here:
To “empy out” the MS-Outlook “Outbox”, Anatella runs an Outlook “Synchronization” procedure: After synchronization, the email from the “outbox” will be place inside the “sent” box. You’ll find more information about the Outlook “Synchronization” procedure inside the section 5.2.24.2.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

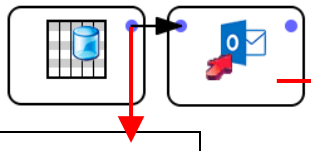
Outlook is not a very fast program: If you need to send a large quantity of e-mails, you should rather use the “Send Emails using your Office 365 subscription” action detailed inside the sections 5.23.54 or the “Send E-Mail” action detailed inside the sections 5.27.7.



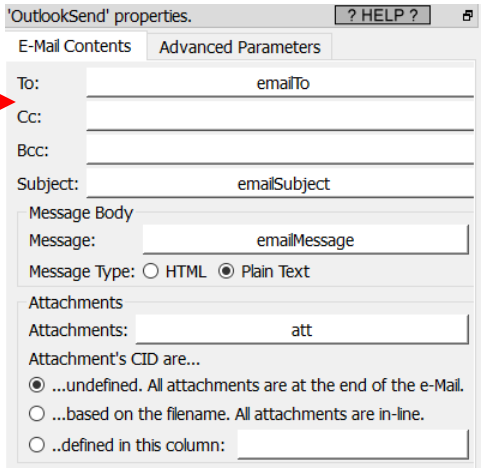
If you see this error message, it means that you need to use the 32-bit version of Anatella to communicate with Outlook (because you have the most common “32-bit version” of Outlook). Please refer to the section 5.2.24.3. to solve this issue.

There are 3 operating modes that are related to the e-mail attachments:

1. <u>Attachment CID are undefined. All attachement are at the end of the e-mail.</u>
   Here is an example:

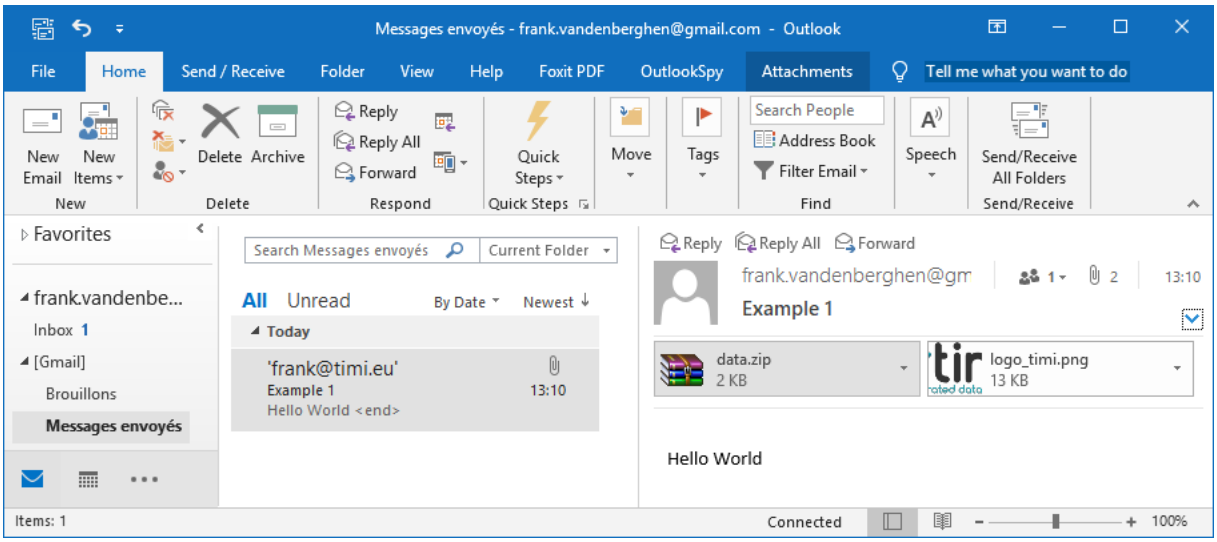


emailTo	emailSubject	emailMessage	att
frank@timi.eu	Example 1	Hello World	:/data.zip,:/logo_timi.png



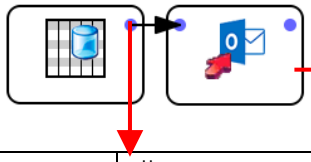
All the attachment’s filenames are separated with a comma “,”.

The resulting e-mail is:

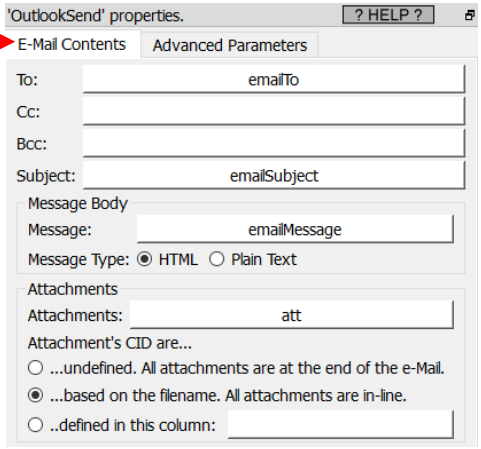


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Attachment CID are based on a filename. All attachements are in-line.

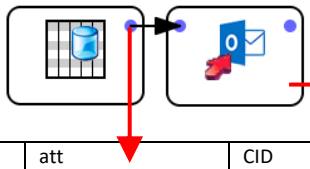


emailTo	emailSubject	emailMessage	att
frank@timi.eu	Example 2	<html><body> Hello World! </body></html>	:/logo_anatella.zip , :/logo_timi.png

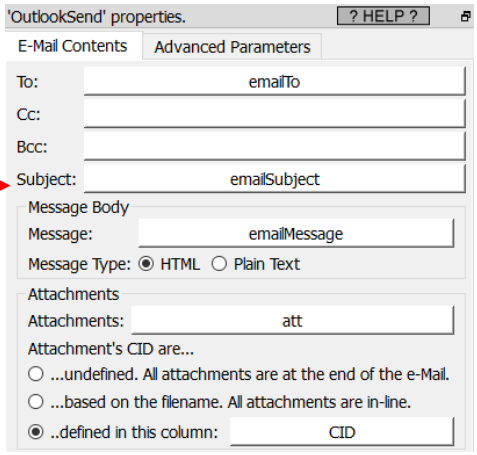


The resulting e-mail is:

3. Attachment CID are defined in an input column.



emailTo	emailSubject	emailMessage	att	CID
frank@timi.eu	Example 3	<html><body> Hello World! </body></html>	:/logo_anatella.zip , :/data.zip , :/logo_timi.png	logo_anatella , , logo_timi



The resulting e-mail is:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

5.27.22.1. Outlook Automation

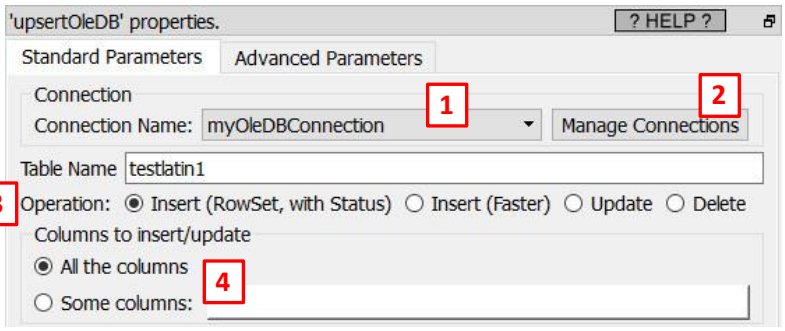
It's quite difficult to automate/schedule tasks that are using the outlookSend Action. This is why you should rather use instead the sendMailOffice action, that offer the same set of functionalities but can be easily automated with Jenkins.

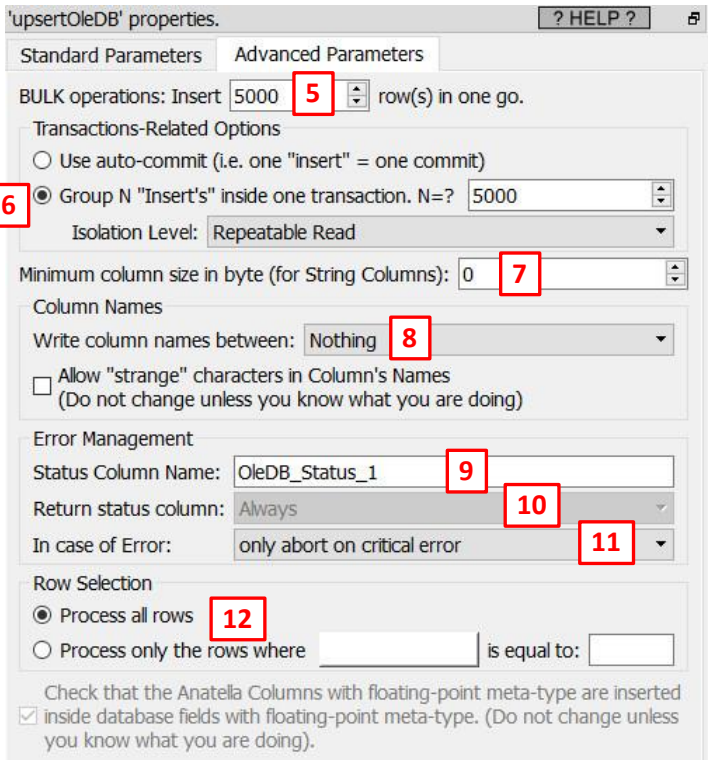
You cannot use Jenkins to run a graph containing the outlookSend Action: You must use instead the "Windows Task Scheduler" (i.e. run "taskschd.msc"). Furthermore, the "security options" of the task must be "*Run only when the user is logged on*" (and you must uncheck the "*hidden*" checkbox option).

5.27.23. Generic OleDB writer

Icon:

Property window:





Short description:
Insert/update/Delete some rows inside a relational database using the OleDB driver.

Long Description:
Please refer to the section 5.1.7. for more informations about the usage of the OleDB protocol to connect Anatella to databases.

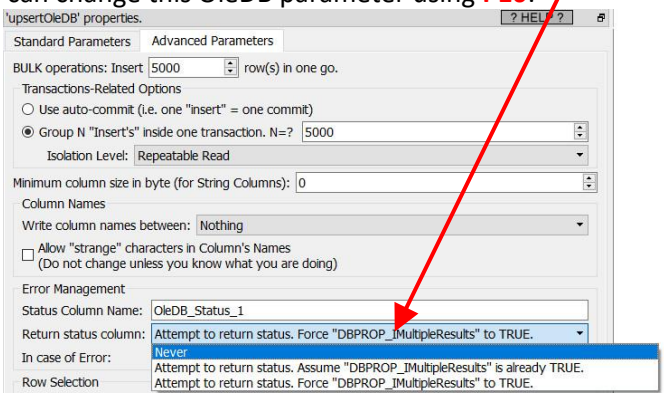
For this action to work, you need first to setup an oleDB connection string: use **P1** and **P2** to do so.

Use the parameter **P3** to choose the operating mode: these modes are self-explanatory. All these operating modes (with the exception of the first one that is named "*Insert(Row, with Status)*") are actually executing many "batch" of insert/update/delete SQL commands (the size of these "batch" is defined using parameter **P5**). When the "batch" size is **P5**="1" (i.e. when we execute all the SQL commands one-by-one), the OleDB drivers always gives us the final status after the execution of the SQL command: i.e. we know if the insert/upate/delete operation succeeded. There are many reasons why a SQL command can fail: The most common causes of failure are:

- You try to INSERT a row inside a table that contains a primary key column that has a unicity constraint. This means that the newly inserted row cannot have a primary key whose value is already present inside the table. If that's the case, the INSERT operation will fail.
- You try to write a very long string (e.g. with a length of 20) inside a column that only accepts short strings (e.g. declared as "nvarchar(10)").

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When the "batch" size is larger than one, the OleDB driver will return the final status, after the execution of the SQL command, **only if** the OleDB parameter named "DBPROP_IMultipleResults" parameter is TRUE. You can change this OleDB parameter using **P10**:



Unfortunately, as of February 2021, all the OleDB drivers that we tested (MS-SQLServer and Oracle) only accept **P10**="Never": see the sections 5.1.7.1. and 5.1.7.2. for more details on this subject.

The operating mode (selected using **P3**) that is named "*Insert(RowSet, with Status)*" is not executing any SQL commands: it's working in a totally different way: The way that it's working is the following: The upsertOleDB action reads rows from the input pin and stores them inside a small "internal" table (named a "*RowsSet*") that is located in the RAM of the Anatella server (to do so, it's using the IRowSetUpdate::InsertRow() instruction). When this small "internal" table reaches a total of **P5** rows, Anatella sends in "*one go*" the whole "internal" table to be INSERTED inside the database (using the IRowetUpdate::Update() instruction). This operating mode has the potential to be extremely fast since it does not have to parse any complex SQL query: it's just a copy/paste of some rows from Anatella to the database (it still fully supports "Transactions"). Unfortunately, the implementation of this operating mode inside the MS-SQLServer oleDB driver and inside the Oracle oleDB driver is of very poor quality and this mode is actually much slower than the "normal" INSERT mode that is based on a batch of SQL commands. So, this mode is finally quite a disappointment. The only reason to use this mode is that it always provides a "status" column, so that we can easily check which INSERT statement failed or succeeded (this "status" column is typically missing when executing some "batchs" of INSERT SQL statement).

The output table of the upsertOleDB action is an exact copy of the input table with the small exception that a "Status" column is added (more precisely, the presence of this "status" column depends on the parameter **P10**). This "Status" column contains the following value:

- OK: The insert/update/delete operation succeeded.
- ERROR: The operation failed.
- SKIPPED: The row was skipped because of the parameter **P12**.
- TOO_LONG: One of the columns to send to the database is too long to be inserted inside a "too small" column from the database. Sometime, this error can be solved very easily: i.e. Just enter a very large number inside the parameter **P7**.
- NO_STATUS: You selected **P10**="*Attempt to return status…*" but the OleDB driver did not return any status. If you see NO_STATUS inside one cell of the status column, this means that the whole content of the status column is erroneous.

When Anatella opens a connection to a database it attempts to guess the size of all the columns to insert/update inside the database. Unfortunately, this "guess" is impossible for some columns (e.g. the guess fails for all the columns declared as "*nvarchar(max)*"). In such situation, Anatella assumes that the "***un***guessable" columns have a maximum length of 5,000 unicode characters (or 10,000 bytes). When this assumption is erroneous, the graph execution stops. You can use the parameter **P7** to

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

correct this assumption. Just increase the parameter **P7** up to the point you don't get any error or abort during the execution of the graph.

The parameter **P11** has 3 possible values:

- "Always abort"
  As soon as one insert/update/delete fails, then the whole Anatella graphs stops.
- "Only abort on critical errors"
  The whole Anatella graphs stops when there is an irrecuperable errors that prevents any connection to the database to work anymore. In particular, the graph execution won't stop when Anatella receives such errors:
  - DB_E_CANTCONVERTVALUE
    The data value for one or more columns couldn't be converted for reasons other than sign mismatch or data overflow.
  - DB_E_DATAOVERFLOW
    Conversion failed because the data value for one or more columns overflowed the type used by the database. This typically happens when we try to write a "large" string in a "small" database column.
  - DB_E_INTEGRITYVIOLATION
    The data violated the integrity constraints for one or more columns of the rowset. This typically happens when we try to write a value inside a primary key column that is already present inside another row of the same table.

  For uncritical errors, the graph execution won't stop but:
  - The status column will contain "ERROR" or "TOO_LONG".
  - The Anatella log window will contain a small text that explains the nature of the error. This text is generated by the OleDB driver and it's usually not very understandable. To get a better explanation of the error, you should set **P5**=1 because, in this case, the OleDB driver usually returns more comprehensible error messages.
- "Never Abort"
  This is self-explanatory.

The parameter **P6** allows to use "Transactions" to insert/update/delete rows inside your database. Usually, it's better to avoid to use any Transactions at all: i.e. On most databases, using the "auto-commit" option (i.e. no Transaction at all) is the fastest option because it means that the database does not have to manage all the ACID properties related to the processing of TRANSACTIONS. This is not always true: See the section 5.27.4. for a more detailed discussion on this subject. When the Transactional system is enabled, you can choose between different isolation levels:

Isolation Level	Dirty read	Nonrepeatable reads	Phantom rows	Only Low Concurrency is possible
Read Uncommitted	Yes	Yes	Yes	Yes
Read Committed	No	Yes	Yes	Yes
Repeatable read	No	No	Yes	Yes
Serializable/Isolated	No	No	No	Yes
Snapshot	No	No	No	No

In the table above, the title of the columns represents the common problems that can (unfortunately) occur during the handling of a transaction. These problems are:

- **Dirty Read**
  A transaction that exhibits this phenomenon has very minimal isolation from concurrent transactions. In fact, it can see changes that are made by those concurrent transactions even before they commit.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For example, suppose that transaction T1 performs an update on a row, transaction T2 then retrieves that row, and transaction T1 then terminates with rollback. Transaction T2 has then seen a row that no longer exists.

- **Nonrepeatable reads**
  If a transaction exhibits this phenomenon, it might read a row once. Then, if the same transaction attempts to read that row again, the row might have been changed or even deleted by another concurrent transaction. Therefore, the Read is not (necessarily) repeatable.

  For example, suppose that transaction T1 retrieves a row, transaction T2 then updates that row, and transaction T1 then retrieves the same row again. Transaction T1 has now retrieved the same row twice but has seen two different values for it.

- **Phantom rows**
  When a transaction exhibits this phenomenon, a set of rows that it reads once might be a different set of rows if the transaction attempts to read them again.

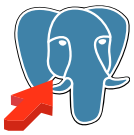
  For example, suppose that transaction T1 retrieves the set of all rows that satisfy some condition. Suppose that transaction T2 then inserts a new row that satisfies that same condition. If transaction T1 now repeats its retrieval request, it sees a row that did not previously exist, a phantom.

- **Only Low Concurrency is possible**
  The "snapshot" isolation level avoids almost all locking inside the database by using a technique named "row versioning". When there are no "locks" used, the database is able to process more operations simultaneously, providing better performances under heavy, concurrent load.

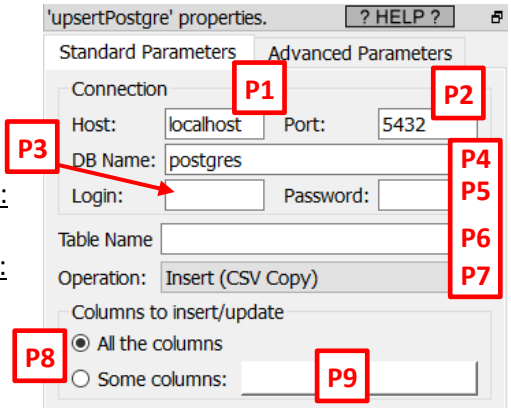
The sections 5.27.4.1. and 10.10 contains more interesting and relevant insights that are related to the management of databases with Anatella.

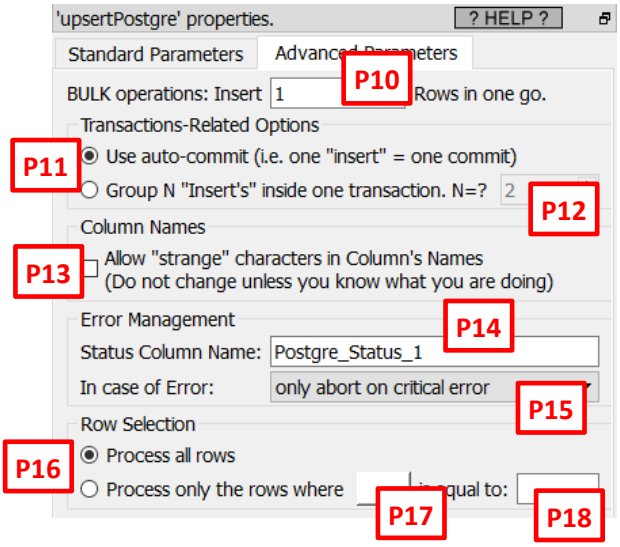
5.27.24. Postgres Writer

Icon:



Property window:





Short description:
Upload a table into PostgreSQL

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Long Description:

The parameter **P7** is the operation to perform (you can choose between insert, update, delete or upsert). Each operation can be performed using several different techniques (for example, to do some "INSERTs", there are 4 different techniques available). Here are the pros & cons of each technique:

Operation Name	Speed*1	Error & Status Reporting*2	Transfer Anatella "Float" type in High Precision*3	All PostgreSQL column's types are supported*4	Transaction supported (no autocommit)
Insert (CSV Copy)	Normal	At the block level*5	No	Yes	Yes
Insert (Binary Copy)*8	Normal	At the block level	Yes	No *7	Yes
Insert (SQL - Text)	Slow	At the row level	No	Yes	No
Insert (SQL - Binary)	Slow	At the block level	Yes	Yes	Yes
Update (SQL - Text)	Very Slow	At the row level	No	Yes	No
Update (SQL - Binary)	Very Very Slow	At the row level*6	Yes	Yes	Yes
Upsert (SQL - Text)	Very Slow	At the row level	No	Yes	No
Upsert (SQL - Binary)	Very Slow	At the block level	Yes	Yes	Yes
Delete (SQL - Text)	Very Slow	At the row level	No	Yes	No
Delete (SQL - Binary)	Very Slow	At the block level	Yes	Yes	Yes

*1: The speed of the insert, update, delete and upsert operations inside PostgreSQL is usually around from 10 to 20 times slower than MS-SQLServer.

*2: If the parameter **P11** is not set to "auto-commit" (i.e. all operations happens inside a user-defined SQL transaction), then you can ignore the content of this column because the status is simply NOT reported at all**.**

*3: The transfer is performed using a 8-byte IEEE754 binary representation of the values that guarantees no rounding errors.

*4: All PostegreSQL column's types are supported as long as they have a textual representation (which is almost always the case).

*5: The number of rows inside one block is defined by the parameter **P10.**

*6: In this mode, the parameter **P10** (i.e. the number of rows inside the buffer to send "in one go" to PostGresSQL) must be set to one.

*7: The supported column's types are: Float, Double, NVarChar, VarChar, Char, Int2, Int4, Int8, Date, DateTime.

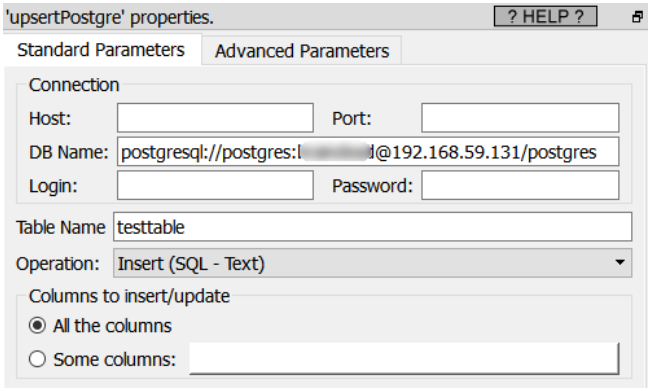
*8: Currently in development.

The parameter **P4** has two meanings: It can either be the "DB Name" or it can be a PostGreSQL connection string as documented here:

https://www.postgresql.org/docs/13/libpq-connect.html#LIBPQ-CONNSTRING

When you use a connection string to connect to PostGre you can set several additional optional parameters (this advanced functionnality is not directly accessible when using the "easy" parameters **P1** to **P5** to setup the connection). Here is an example of connection to PostGreSQL using a connection string:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

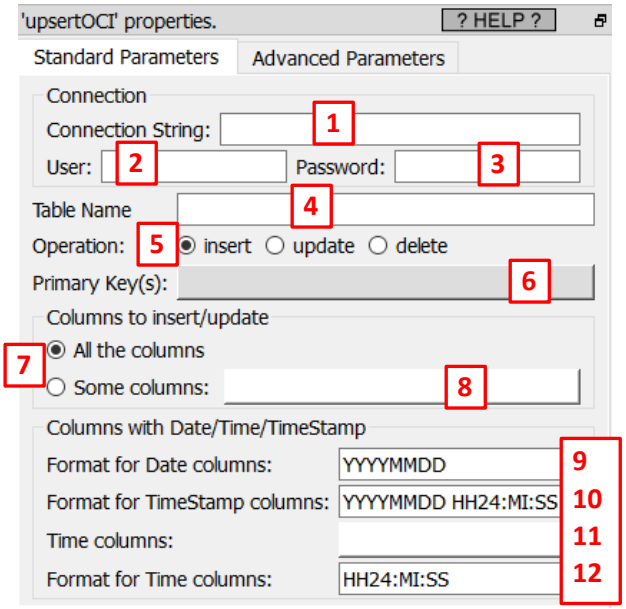


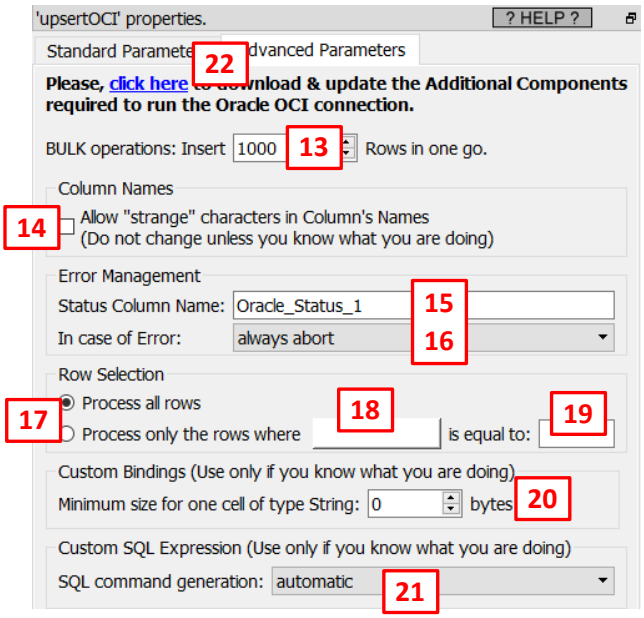
The syntax is "postgresql://<login>:<password>@<server_ip>/<database>?<option_name>=<option_value>".

5.27.25. Oracle writer

Icon:

Property window:





Short description:
Insert/update/Delete some rows inside an Oracle database using the OCI drivers.

Long Description:
Please refer to the section 5.2.26. for more informations about the usage of the OCI drivers to connect Anatella to Oracle databases.

Before using this action, you first need to download the additional Oracle OCI components (it's a 196MB download): i.e. Click the blue URL (parameter **P22**) to automatically download and install the Oracle OCI components.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The parameter **P1** (named "Connection String") can be a reference to the file "TNSNAMES.ORA", or it can be directly the contents of the file "TNSNAMES.ORA". For example, a "Connection String" such as this one is perfectly valid:

```
(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.1.1) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = XE)))
```

The parameter **P13** represents the number of rows that are injected "in one block" to the database. A higher value is better because it sends larger "blocks of rows" and thus minimizes round trips of network data transfers (and thus **the data injection runs faster**). Do not put a value too high neither, otherwise the Oracle drivers may crash (and it also consumes a little more RAM).

If you get an undescribed error with very little information about why there is an error, set the parameter **P13** to "1" and run again the upsertOCI Action. You should now have a more detailed explanation of the error. Once the error is fixed, don't forget to put back the parameter **P13** to "1000" to have a high processing speed.

If you get an error while executing the upsertOCI Action that looks like that:

```
FAILED: OCIEnvCreate()
OCI FAILED: Cannot execute the SQL code(1): □□
```

…this means that you used the readODBC action or the upsertODBC action to access Oracle. As soon as you use an ODBC-based connection to access data inside Oracle, all OCI-based connections stop working (and the other way around is also true). You cannot mix the two types of Oracle connections (ODBC and OCI) inside the same graph.

The solution is: To regain access to Oracle, close the Anatella windows (click the × button in the top-right corner) and re-open Anatella. Once Anatella is re-open you can re-start using OCI again.

Regarding the parameter P9 and the way Oracle handles Dates

A time-related column inside a table can contain:

- Some **Time** (hours+minutes+seconds)
- Some **Date** (year+month+dayOfMonth)
- Some **Time Stamp** (year+month+dayOfMonth + hours+minutes+seconds)

Inside Oracle SQL, when you create a new table, you can define a time-related column …

- ..as the SQL type "DATE" **or**
- ..as the SQL type "TIMESTAMP".

Oracle does NOT enforce any type of content inside a SQL "DATE"-type column or inside a SQL "TIMESTAMP"-type column. This means that, inside these columns, you can find any kind of content: you'll find randomly "**Time**", "**Date**" and "**Time Stamp**" informations.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Despite this terrible decision from Oracle, you can still decide on your own to declare your column's SQL types in a consistant way. If you made sure to declare a column with..

- …the SQL "DATE" type when it actually really contains some actual **Dates** or **Times**.
- ..the SQL "TIMESTAMP" type when it actually really contains some actual **Time Stamps**.

..then you can use the parameters **P9** to **P12** to easily inject all your time-related data inside Oracle. Here are more details on how to use these parameters **P9** to **P12**: When Anatella sends a new data row to Oracle, Anatella looks at the SQL type of the columns inside Oracle:

- if the SQL-type of the column X is "DATE", this means that the columns X is more susceptible to contain either a **Date** (year+month+dayOfMonth) or a **Time** (hours+minutes+seconds). To decide between the two contents (**Date** or **Time**), Anatella uses the parameter **P11**:
  - if the column X is **not** inside the list given inside parameter **P11**, then the column X is a simple **Date** and its format is specified inside the parameter **P9**.
  - if the column X is inside the list given inside parameter **P11**, then the column X is a **Time** and its format is specified inside the parameter **P12**.
- If the SQL-type of the column X is "TIMESTAMP", this means that the columns X is more susceptible to contain a **Time Stamp** (year+month+dayOfMonth + hours+minutes+seconds) and its format is specified inside the parameter **P10**.

The syntax used inside the parameters **P9**, **P10**, **P12** (to describe the time-format) originates from Oracle. It is explained on this page: https://www.techonthenet.com/oracle/functions/to_date.php

The default setting for the Parameter **P9** is "YYYYMMDD".

If the dataset to inject into Oracle comes from the OracleOCI action (with all the default settings) then, you should change the Parameter **P9** to "YYYYMMDD HH24:MI:SS". If you forgot to do that change, you will get an error from Oracle telling your that your date-formatting is erroneous.

If the dataset to inject into Oracle comes from the OracleOCI action **AND,** inside the OracleOCI action, you also changed the parameter "*Extract DATE column as*" to the value "YYYYMMDD" (which is the recommended setting if you decide to be consistent in the declaration of your SQL columns), then you can keep the default value for the Parameter **P9** (that is "YYYYMMDD") inside the upsertOCI Action.

To summarize:

* if you are **consistent** in the declaration of your SQL columns, set the parameter "*Extract DATE column as*" to the value "YYYYMMDD" inside the OracleOCI action.

* if you are **inconsistent** in the declaration of your SQL columns, set the parameter Parameter **P9** to "YYYYMMDD HH24:MI:SS" inside the upsertOCI action.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6. Distributed/Cloud Storage

6.1. Introduction

Anatella is compatible with many remote/distributed/cloud storage solutions.
Using Anatella, you can manage (read, write, delete and list) files inside:

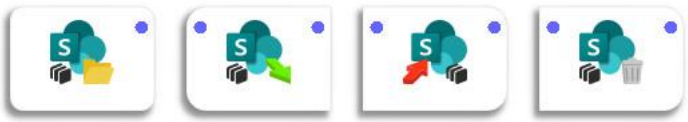
S3 Storage (AWS and many others)

Google Drive

Google Cloud Storage

OneDrive

Azure Storage


Sharepoint drives (libraries)

OpenStack Swift

Huawei Object Storage

Dropbox

Mailbox

The above Actions are all working following the same principles: For example:

- Let's assume we want to read an Excel file initially stored inside your google drive.
  This is done in two steps:
  - **Step 1:** Use the GDriveDownload Action to download the excel file from our GoogleDrive and save it inside a temporary file on the local drive.
  - **Step 2:** Use the readExcel Action to read the content of the temporary Excel file stored on the local drive (that we just create on step 1).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Let's assume we want to write some data into an Excel file that is stored inside your OneDrive. This is done in two steps:
  - **Step 1:** Use the writeExcel Action to write the data inside a temporary Excel file stored on the local drive.
  - **Step 2:** Use the oneDriveUploadFiles Action to upload the excel file that we just create on step 1 into our OneDrive.

Inside both of the two examples hereabove, we are using a temporary file. These two examples work fine but they are not very efficient. Indeed, if we want to read a large .gel_anatella files (e.g. a 100 GB file), we'll have first to download the file and save it on the local SSD, before reading it again with the readGel Action. This is inefficient: this consumes too much time and too much SSD space. It would be nicer if we could directly "stream" row-by-row the content of the .gel_anatella file from the remote/distant/cloud drive directly inside Anatella without going through a temporary file.

Due to the "hype" around the HDFS technology, in June 2017, we produced a new version of Anatella (Anatella v1.65) that adds the possibility to read&write data from/to an HDFS drive in "streaming" (without requiring any temporary files, exactly as explained in the paragraph here above). This was a disaster. HDFS is terrible: very slow, very unreliable. At our customer sites, we heared many horrible stories about (1) unrecoverable HDFS drives (when all the data and files from the customer are lost because of a crash of HDFS) and (2) "security holes" that allowed unethical hackers to perform ransom-ware attacks. We must now admit and acknowledge that, despite all our best efforts, we were not able to help our customers to use HDFS in a reliable way. So, we scratched all the HDFS code from Anatella (that was painfully crafted during nearly 2 years! So much suffering to create this code, all for nothing!) and we went searching for a better solution. This was not an easy decision to make. We dwleve in more details on the reasons behind this decision in the section 10.13 of this document.

The alternative, better solution arrived a few years later in the form of a general concensus from all the large players in the "big data" industry around a new protocol: the S3 protocol. Initially, the S3 protocol was invented by Amazon to access the "*AWS S3 Objec Storage*". The full documentation of the S3 protocol is here:
https://docs.aws.amazon.com/AmazonS3/latest/API/API_Operations_Amazon_Simple_Storage_Service.html
The S3 protocol allows to access a drive (i.e. a "bucket" in Amazon terms) to read and write files (i.e. the "files" are named "objects" in Amazon terms). This is a complete protocol that also allows to copy, delete, and list files (currently, you cannot rename files/objects, but this should be added soon since it's a much-requested feature).

The S3 protocol is much more cleverly designed, has better security features (all headers are "signed") and allow for a much higher throughput than HDFS. Furthermore, there are now dozens of mature projects implementing the S3 protocol and all of them are faster and more robust than HDFS ever was. Here are the currently most prominent projects that offers an S3-interface to manage you own remote/cloud/distributed drive on your own machines:

- MinIO (coded in Go)
- OpenIO (coded in python and C)
- SeaWeed (coded in Go)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- OpenStack Swift (through their S3 adapter – Coded in Python)
- Ceph (coded in C)
- ..and many more other projects here: https://github.com/topics/s3-storage

There are also many s3-compatible commercially available cloud/distributed drives that are 100% managed for you: Here are some examples:

- The "original" Amazon AWS S3 object storage: https://aws.amazon.com/s3/
- https://www.scaleway.com/en/object-storage/
- https://www.exoscale.com/object-storage/
- https://fuga.cloud/services/s3-objectstorage/
- https://www.idrive.com/s3-storage-e2/
- ..and many others.

During the course of the development of the S3-related code inside Anatella, we used for testing the "original" Amazon AWS S3 object storage and MinIO. MinIO was so easy to setup and install that it made sense to use it for a little "testing needs". Ceph could also be an interesting alternative since it's coded in C and thus it might be faster (but installing Ceph seems really cumbersome).

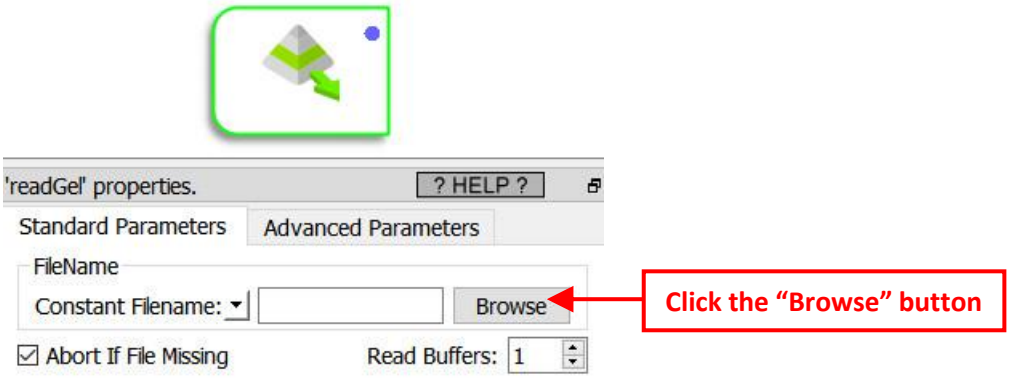
6.2. Examples of S3 Integration with Anatella

Inside Anatella, the S3 drive is no different than a "standard, local" drive (such as the "C:" drive): Here are some examples of the S3 Integration within Anatella:

6.2.1. Example 1: Browsing S3 drives

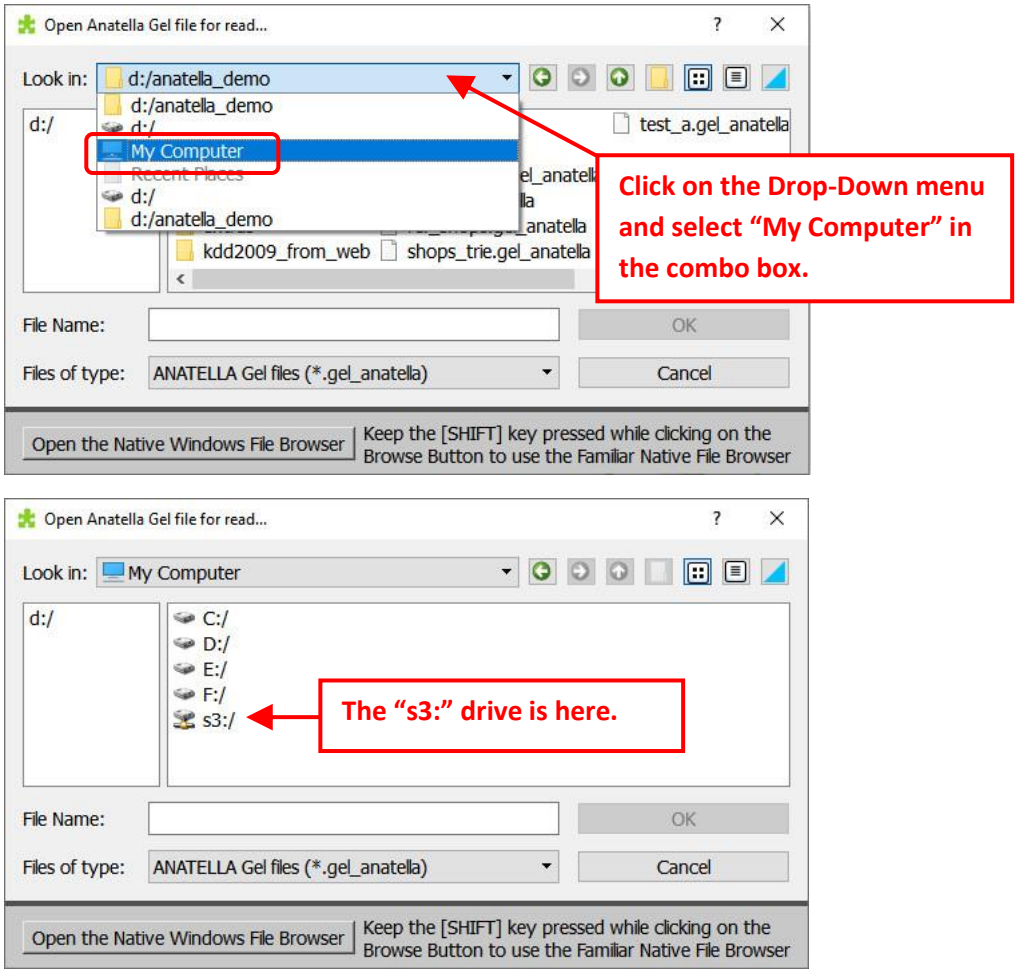
You can browse your "s3:/" drive using the extended "file browser" from Anatella. In this example, I am browsing for a .gel_anatella file to read using the readGel Action.

Step 1: Click the "Browse" button inside the property window of the readGel Action:

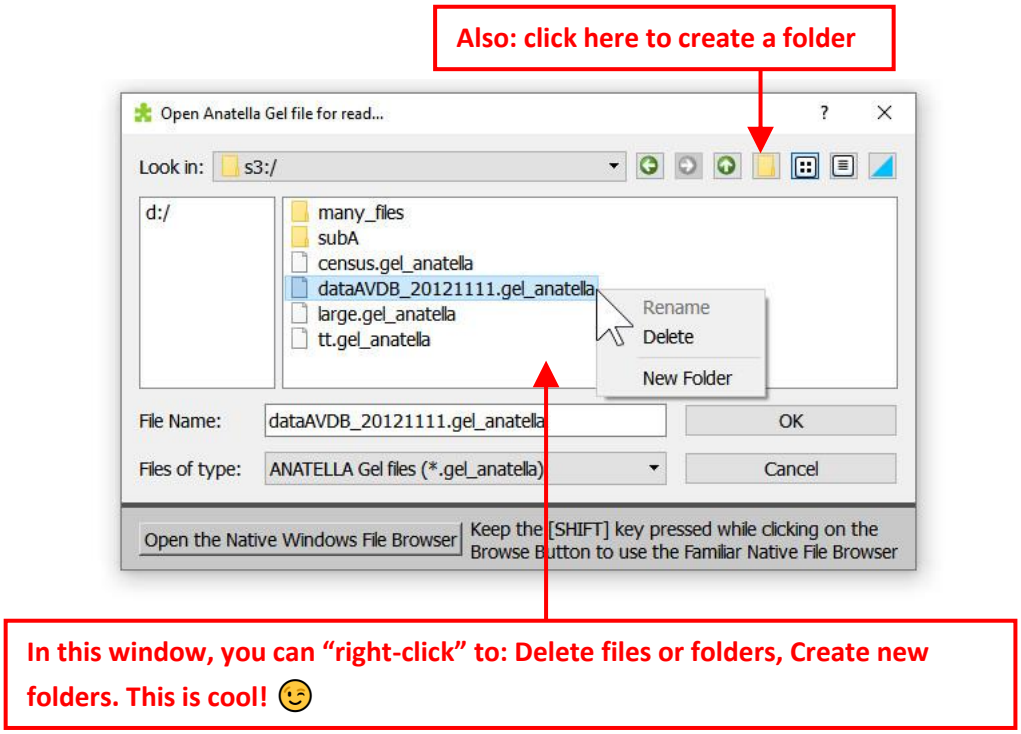


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Step 2: Navigate to the Top of the directory Structure to see the "s3:/" drive:

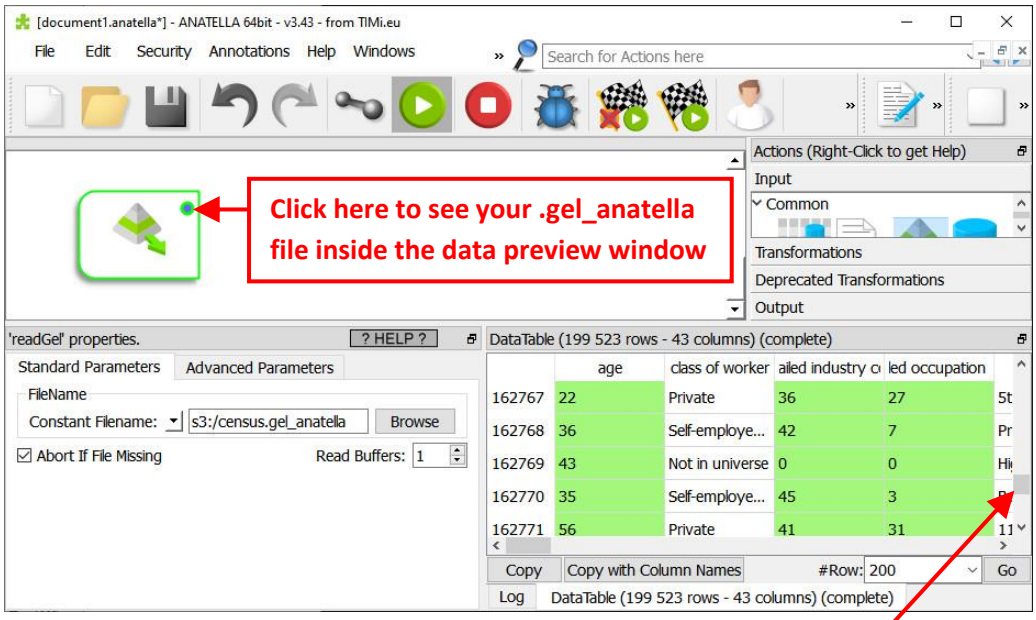


Step 3: Navigate down the directory Structure inside the "s3:/" drive to reach the required .gel_anatella file:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

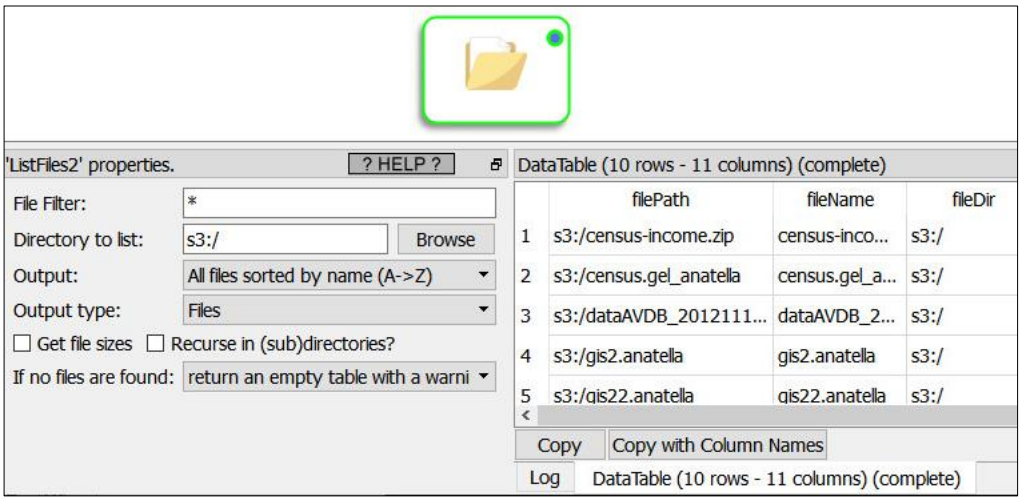
Step 4: After clicking the "OK" button inside the "File Browser Window", you get:



At that point, you can see the content of the .gel_anatella file contained on the S3 drive "as if" this file is stored on the local drive. This means that you can move freely the slider here: to move freely in the data preview window, even on .gel_anatella files that have a size of several terabytes. When you move the slider, the data-preview-display is refreshed instantaneously (whatever the file size or the file location: "s3:" or "c:"): This is cool! 😉

6.2.2. Example 2: Listing Files/Directories on S3

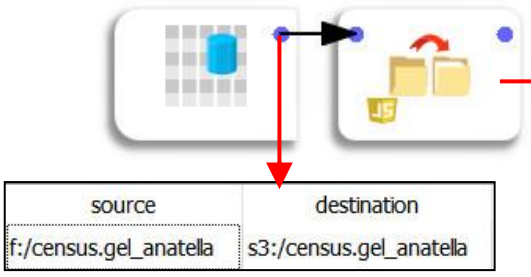
You can use the listFile2 Action (see section 5.21.4 for more details about this action) to list the content of the "S3" drive: For example, here is the content of the "s3:/" directory:

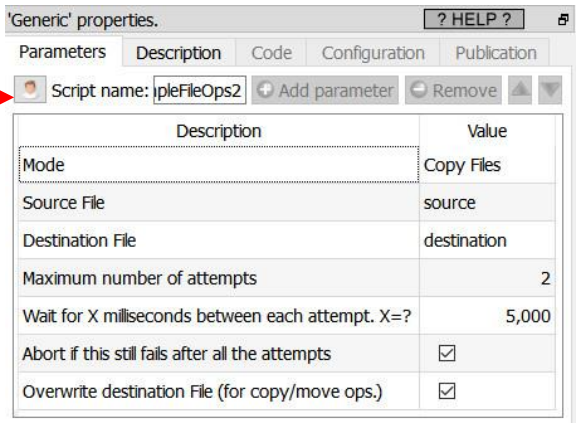


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6.2.3. Example 3: Copying Files from/to the S3 drive

You can use the simpleFileOps2 Action to manage (copy/delete) files on the "s3:" drive: For example, this copies the (local) file "f:/census.gel_anatella" inside the "s3:" drive:



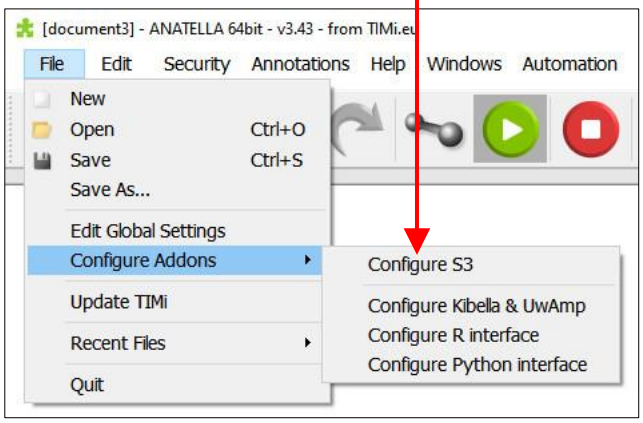


The simpleFileOps2 Action allows to:

- copy files between the S3 drive and the local drive (this usually takes a little bit of time since it involves a large network traffic to upload/download all the data).
- copy files from the S3 drive to another directory on the S3 drive (this is usually almost instantaneous because it runs without any network traffic from/to your local machine: everything is handled for you by the remote S3 server)

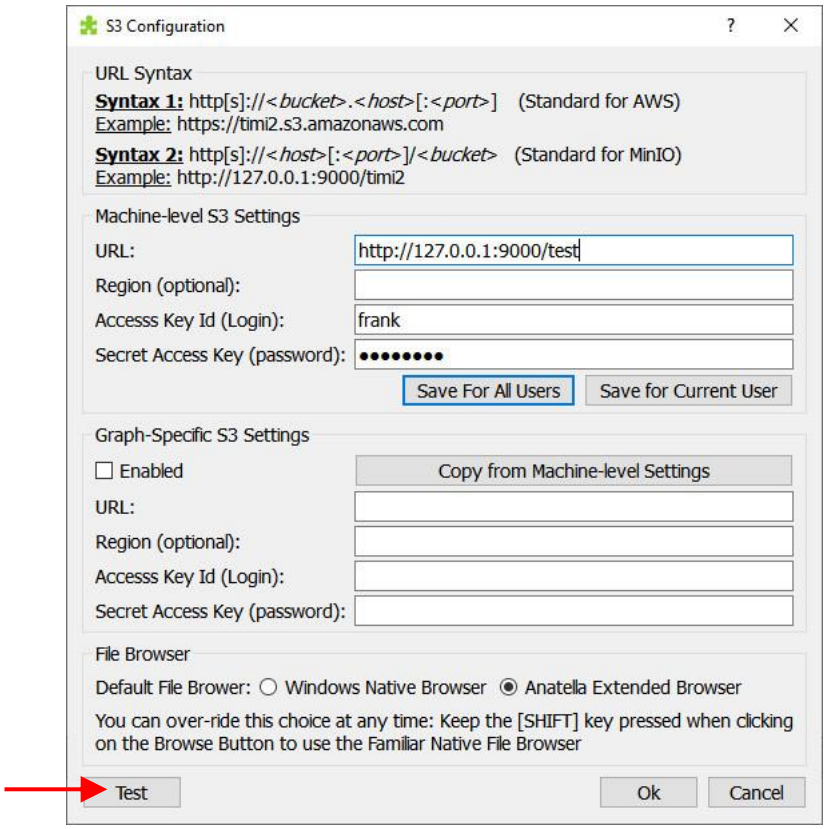
6.3. First time setup

1. Open the S3 configuration window: Click here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Enter the URL address, the Access Key ID, the Secret Access Key (and, optionally, the Region) of your "S3 drive" inside the S3 configration window:



For the "URL address" parameter of the S3 drives, there are two different syntax:

- **Syntax 1:**
  The standard AWS syntax where the bucket name is a sub-domain of the host.
  http[s]://<bucket>.<host>[:<port>]
  Example: https://timi2.s3.amazonaws.com

- **Syntax 2:**
  The MinIO syntax where the bucket name is at the end, part of the path on the server:
  http[s]://<host>[:<port>]/<bucket>
  Example: http://127.0.0.1:9000/timi2

3. Click the "Test" Button in the bottom left corner of the S3 configuration window: Anatella will try to access the S3 drive. If the test succeeds, you will see a file browser window that shows you the content of your S3 drive. If the test fails, you'll get an explanation about why it failed inside the standard Anatella Log Window.

4. ***At any time*,** you can also change which file browser you are using **by default** inside this configuration window (at the bottom of the window): i.e. There is no need to have a live S3 connection to change this setting. If you don't want to use anymore the "Anatella Extended Browser" (because you prefer the native Windows File Browser and you won't be using S3 anymore for some time), just click at any time on the "Windows Native Browser" option.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6.3.1. Manual First Time Setup

You can use the setting-window described inside the previous section 6.3. to edit and save all the parameters required to access to your S3 drive. All these settings are actually saved inside a single DOS-environmental variable named “TIMiS3FS” (that’s an acronym that stands for “TIMi S3 File System”) that you can manually edit using any other tool. The syntax of this “TIMiS3FS” DOS-environmental variable is:

<url>;<region>;<Access Key ID>;1<encrypted Secret Access Key>

…or :

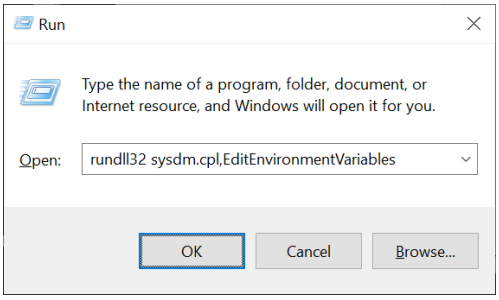
<url>;<region>;<Access Key ID>;0<Secret Access Key in plain sight (not encrypted)>

Here are two example screenshots that shows the “TIMiS3FS” DOS-environmental variable:

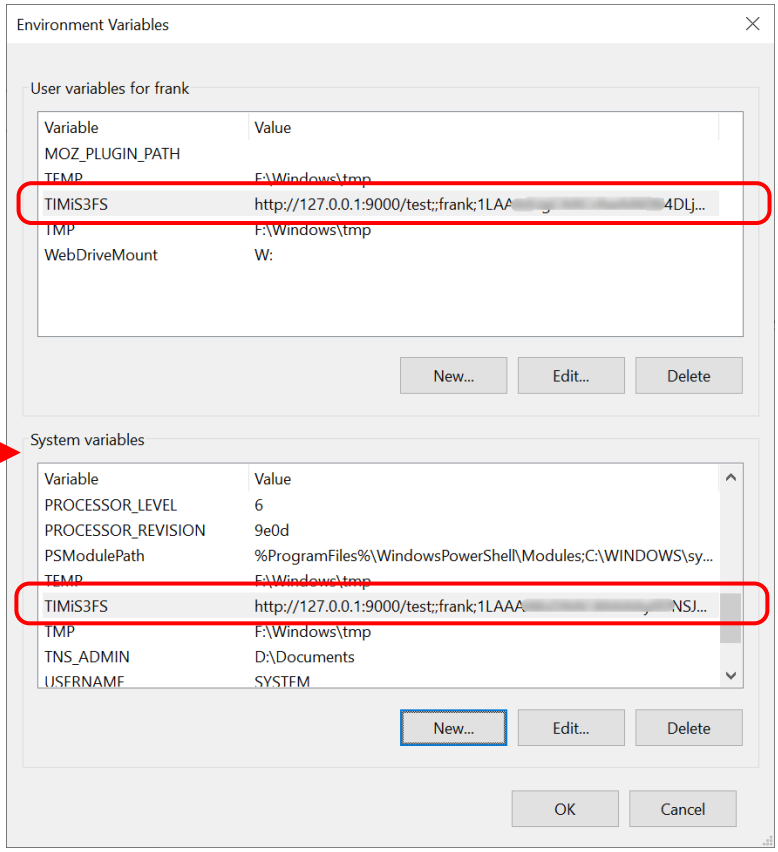
Alternativily, you can press [WIN]+[R] and run inside this window the instruction:

```
rundll32 sysdm.cpl,EditEnvironmentVariables
```

Here is a screenshot:



..and then you get the classical window from win10 to edit the all the DOS-environment variables:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

6.4. Do I need an S3 drive?

The advantages of an S3 drive are:

- It's a really cheap storage solution **compared to a traditional database system** because:
  - It only uses common grade hardware (i.e. low costs PC's) that is much cheaper than the dedicated/specialized hardware (such as the "InfiniBand network cards") found in Teradata, Exadata, etc. databases.
  - it's open source and free software: There are no licensing costs (if you use MinIO, Ceph or other "open source" tools).
- It's a solution that is easily extensible: If you need more storage, simply add more servers.

The disadvantages of S3 are:

- It's an **expensive storage solution** compared to a traditional SSD/NAS/SAN/RAID6 drive.
  It's more expensive because you need to hire specialized staff for the maintenance and support of the software stack (MinIO, Cepth, etc.) that manages the S3/cloud drive.
- The data access speed of an S3 drive is slower than a local SSD/RAID6 drive (that runs between 500 MByte/sec and 2000Mbyte/sec). With S3, you can expect a read speed around 100 MByte/sec (it mainly depends on your network cards).

Despite these two major defaults, the S3 drive is one of the only solution that offers (theoretically) an unlimited data capacity (just add more nodes to get more capacity) and it can thus make sense to use an S3 drive if you need to store really large volumes of **cold** data that do not fit inside a SSD/RAID6 drive (despite the very efficient data compression algorithms used in the .gel_anatella files and in the .cgel_anatella files used in Anatella). Needless to say that a situation where the usage of an S3 drive is really justified by technical constraints is a situation that occurs almost never (at least, not when you are using the highly compressed file formats available inside Anatella).

6.5. Expectations

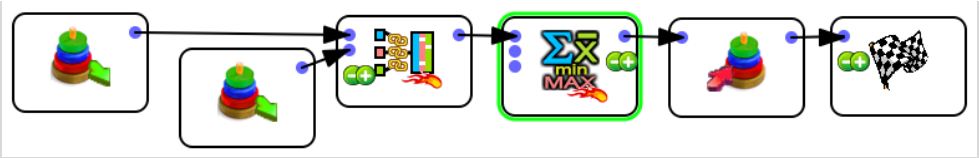
One good thing to keep in mind when using an S3 drive and Anatella together is the following: The slower I/O speed of the S3 drive is (usually) not really a problem when you manipulate your data using Anatella because, when running an Anatella-Data-Transformation-Graph, the I/O speed is usually not the "bottleneck" (i.e. when using Anatella, the only bottleneck is usually the CPU).

Within Anatella, the only bottleneck is usually the CPU. How is that so?

The Anatella engine has two important properties that make it mostly insensible to I/O speed:
1. The Anatella engine works "in streaming" (meaning that it processes data row-by-row: see sections 5.3.2.1. to 5.3.2.5 for more details about this subject).
2. Reading (and writing) a data file inside Anatella is done using an asynchronous algorithm (see section 5.2.6.2. for more details about this subject)

This meaning that, most of the time, Anatella is able to simultaneously "*read the required data file*" and "*process the data*". In this example:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



… we simultaneously:
* read the data
* compute the join and the aggregate.

Let's now assume the following: In the above example:
* the I/O read speed is around 100MB/sec
* the computation speed (to compute the join and the aggregate) is around 50 MB/sec.

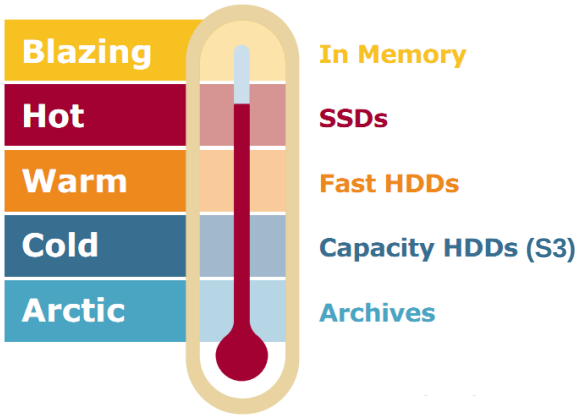
Looking at the above assumptions, we can see that the "bottleneck", when runnng this Anatella graph, is not the I/O speed: It's the computing/CPU speed (as it is nearly always the case with Anatella): i.e. The global speed of the whole Anatella graph is limited to 50 MB/sec because of the limited computational speed of the join and the aggregate.

In practice, one quickly realizes that the I/O speed is almost never the "bottleneck element" that decides of the overall execution time of an Anatella-data-transformation-graph.

This "*insensibility to the I/O speed*" is also due to the nature of the usual tasks that are performed using Anatella. Since Anatella is built for analytics and predictive analytics tasks in mind, a typical "workload" requires complex, CPU-intensive computations (to create refined KPI's or to do "feature engineering"). These CPU-intensive operations usually represent 95% of the computation time (i.e. inside Anatella, the CPU is usually the bottleneck). So, if we can read the data faster, we will (maybe!) just gain a few percent out of the 5% of the time that Anatella devotes to reading the data.

On the other hand, it's true that, for a very simple "Anatella-data-transformation-graphs" (e.g. for example, a graph that contains only one simple aggregate to compute), it's worth reading the data faster.

To alleviate the slightly slower speed of S3, we can also choose how the data is stored using a methodology based on "data temperature". We use the term "temperature" as a metaphor for "frequency of access":



- Seldom-used data is "cold data" that should be kept on low-cost, high-capacity disk storage, such as S3.

  Cold data may not be the most popular, but it is used daily to support vital strategic analytics and decision-making. The cold data is often last year's financials, government risk and compliance data, or consumer behaviours that is used by the marketing department for loyalty or churn analysis.

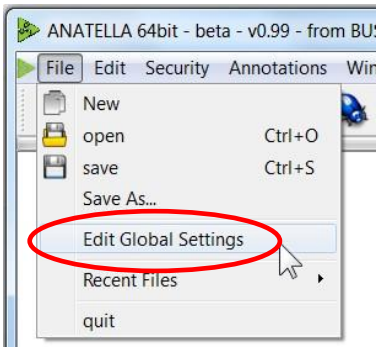
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- High-use data is “hot data” and it should be placed on high-performance SSD, SAN/NAS or RAID6 drives.

  Hot data typically includes all recently collected data tables. It also includes all the “heavy used” datasets that are automatically prepared every night and used during the day by the data scientists to complete their daily tasks.

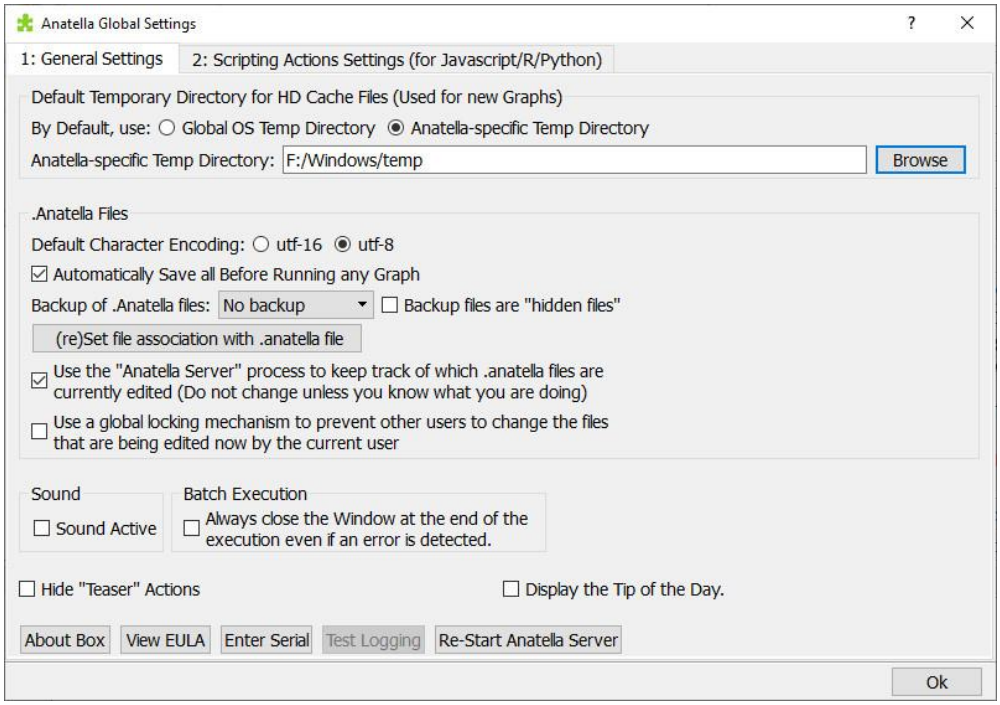
7. Anatella Global settings

To get the “Anatella Global Settings” windows, select the “*Edit Global Settings*” option inside the “*File*” drop-down menu:



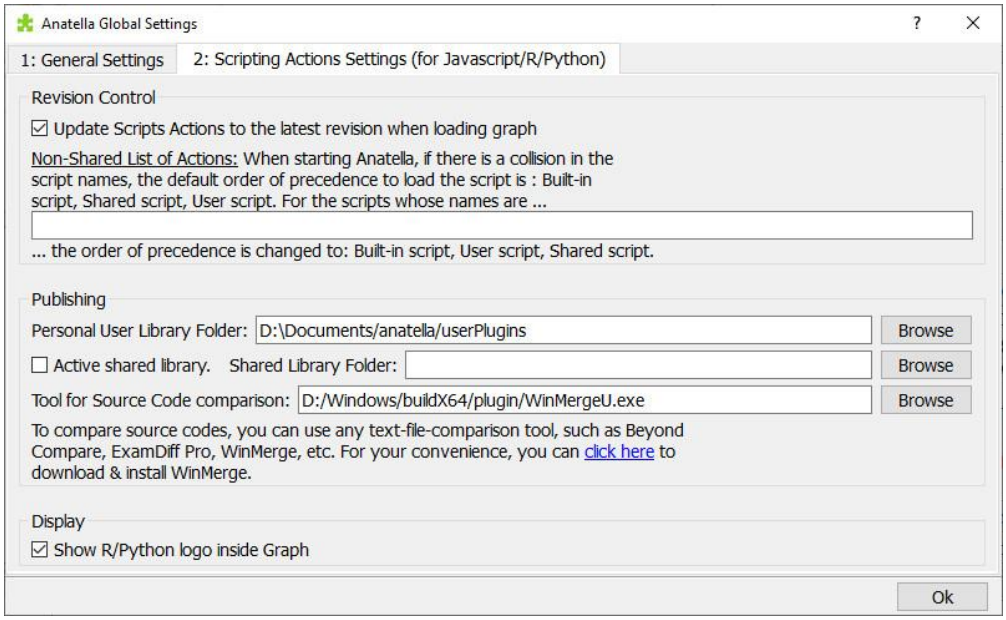
The “Global Settings” windows opens: It contains 2 tabs:

- **Tab 1: “General Settings”**



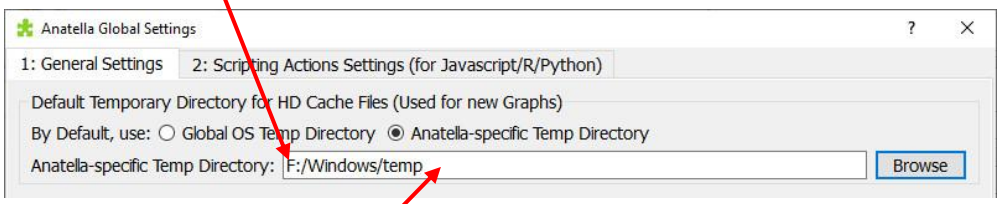
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- **Tab 2 : "Scripting Actions Settings"**



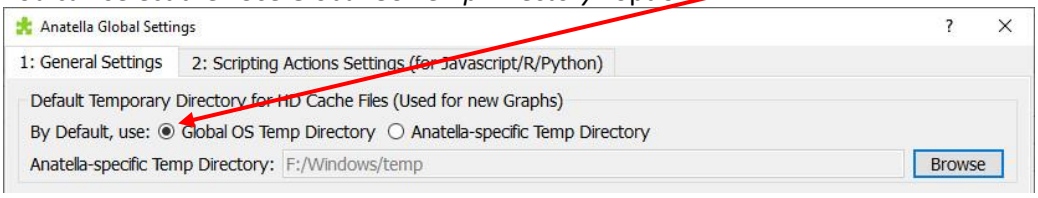
7.1. Panel 1: Default Temporary Directory for HD Cache Files

Inside any newly created Anatella Graph, the location of the "*Working directory for cache files*" is the location stored here:



There are two different ways to change this location:

1. You can simply edit this text field
2. You can select the "*Use Global OS Temp Directory*" option:



Once you have selected the "*Use Global OS Temp Directory*" option, all the **new**ly created Anatella graphs will use the "*Global OS Temp Directory*" setting: i.e. you still need to manually change all your **old** Anatella graphs, to set them to use the option "*Global OS Temp Directory*" inside the "Graph Global Parameters" window (if you have many graphs, you can automate this change using the GraphEncrypt Action. See the section 10.12 for an example on how to use the GraphEncrypt Action).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

It's a good idea that all the concurrent users of Anatella in a "compute server" use different working directories (possibly on different drives).

By default, the "*Working directory for cache files*" location is set to the value "*Use Anatella-Specific Temp Directory*". The location of the "*Anatella-Specific Temp directory*" parameter is different for every user and it's initialized to be the same location as the location that was stored inside the "%TEMP%" MS-Windows Global Environmental Variable when you ran Anatella for the very first time. You don't need administrative rights to change this location.

You can also temporarly change the "*Working directory for cache files*" parameter using the command-line parameter "-c": see section 4.7.5.

If you look inside the "HD Cache" directory, you will see many files with the extension "*.gel_Anatella". These files are the different Gel files (or "cached tables") of your Anatella-Transformation-Graph. Each time you create a new "HD Cache", a new "*.gel_Anatella" file is created.

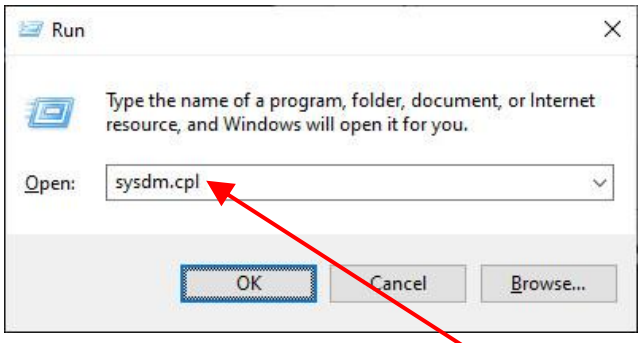
When you are running several Anatella graphs in parallel, Anatella automatically selects the filename of all the temporary "Gel Files" so that no "collisions" will ever occur (a "collision" is when two Anatella un-related processes attempt to write in the same temporary "Gel File").

7.1.1. Changing the value of the "Global OS Temp Directory" parameter

When using the "*Global OS Temp Directory*" option, the location of the "*Working directory for cache files*" parameter is contained inside the "%TEMP%" MS-Windows Environment Variable (you can still change this default behavior for a particular graph using the "Anatella Global Settings" windows).

To change the "%TEMP%" Environment Variable:

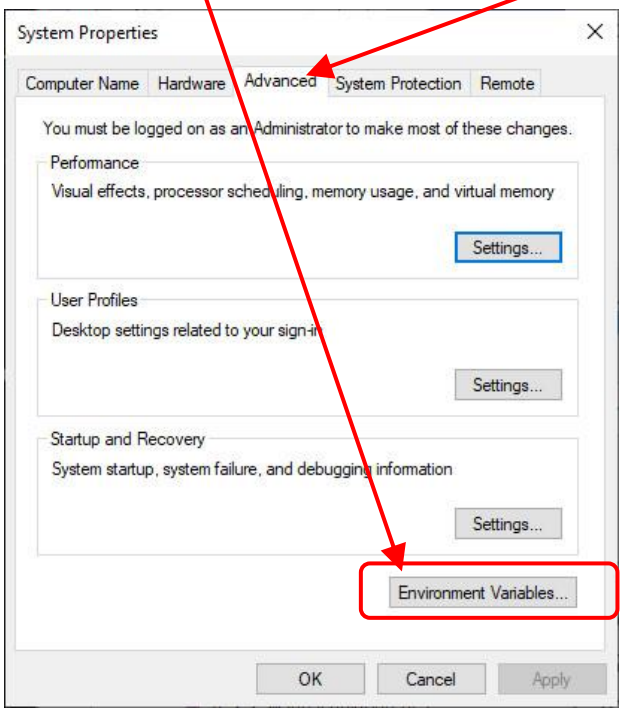
1. Press the [Win]+[R] keys simultaneously. This opens the "Run" window:



2. Inside the "Run" window, enter sysdm.cpl here: and press [Enter]. At the UAC prompt, respond "ok".

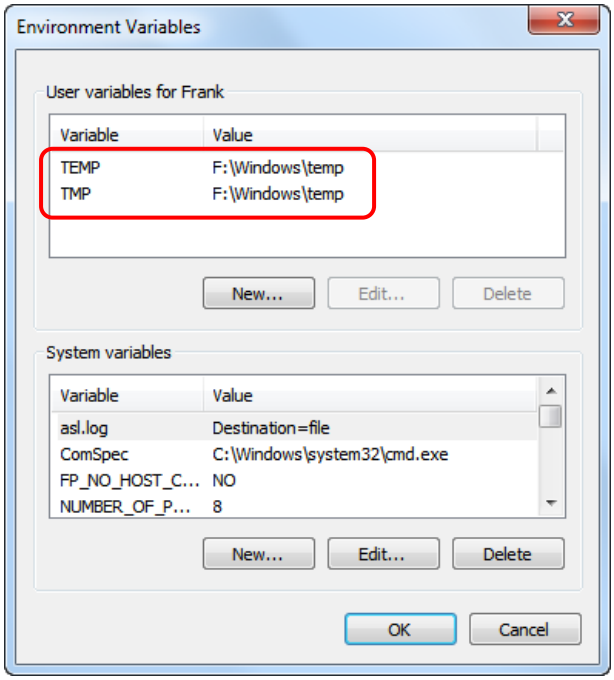
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Inside the "System properties" window, inside the "Advanced" panel , click on the "Environment Variable" button



4. Inside the "Environment Variable" window, edit both the "TEMP" and the "TMP" variables to set the default value for the "*Working directory for cache files*".

   For example: Currently, my "Cache files" are stored inside the "F:\Windows\temp" directory:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

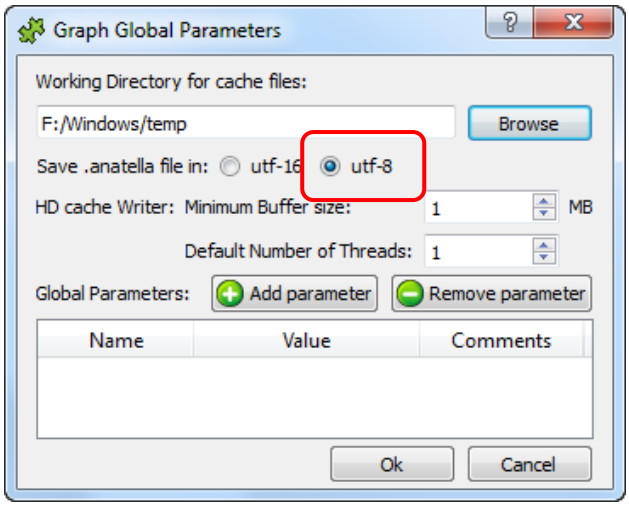
7.2. Panel 1: .anatella files

7.2.1. Character Encoding

The "*Character Encoding*" option is mainly useful for Git integration: By default, all .anatella files are saved as Unicode UTF-16 files (because, internally, Anatella handles all characters as UTF-16, so that using UTF-16 is the fastest, most efficient option). Unfortunately, Git does not recognize UTF-16 text files (i.e. it only see them as binary files). To enable Git to properly handle your .anatella files, you should select the "UTF-8" option (to save all your **new** .Anatella files in UTF-8).

You can also convert your **old** .anatella files to UTF-8, using two different procedures:

1. Open the .anatalla file to convert to UTF-8 and click on the "utf-8" option inside the "Graph Global Parameter" window:



…and save your file (i.e. press [CTRL]+[S]).

2. Use the GraphEncrypt Action to convert "in bulk" hundreds of .anatella files to UTF-8.

7.2.2. .anatella file Automatic Save

The option "*Automatically Save all Before Running any graph*" is self-explanatory. This option is interesting because it ensures that you never lose the last modifications done inside your graphs. For more information about this subject: see section 5.4 (and also section 5.3.).

7.2.3. Backup of Anatella files.

The option "*backup of Anatella files*" is self-explanatory. Backups allows you to keep track of all the changes that you recently made to your .anatella files. "Backups" are quite handy if you want to "undo" the latests changes made to a graph: Simply "go back" to a previous backup. To keep track of the changes made on your .anatella file during a longer time period, use a versioning system (such as Git): see section 5.9. about this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

7.2.4. Global locking mechanism

The option "*use a global locking mechanism*" is usefull when there are several data analysts working simultaneously (and collaboratively) on the same server. This option prevents two analysts to simultaneously edit the same .anatella graph file. Several analysts or processes can still run the same .anatella graph file simultaneously but the simultaneous edition is prevented/blocked.

This setting is defined at the level of the server (it's not user-specific).
You need administrative rights to change this setting.

More details:

When this option is enabled, each time an analyst opens an .anatella graph file for edition, Anatella directly create a small "locking" file (with the same filename and the same directory as the edited ".anatella" graph file but with the extention ".lock_anatella". The "locking" file has the "hidden" attribute).

Then, if another analyst attempts to open the same ".anatella" graph file, Anatella stops with an error message saying that the file is already currently being edited/locked by another analyst (and it also gives the name of the other analyst that is currently "locking" the file, so that you can contact him to request that he terminates his edition).

To unlock a .anatella file prematurely, just delete the ".lock_anatella" file.

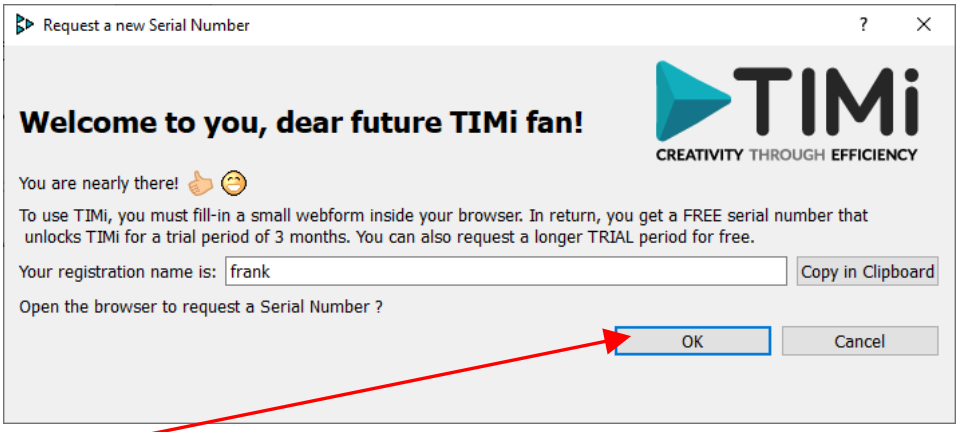
7.3. Panel 1: Sound

When your data-transformation is complete, Anatella can emit some "alert" sounds to let you know that you can resume your work.

7.4. Panel 1: Serial Number

7.4.1. Get your serial Number at the first execution of Anatella

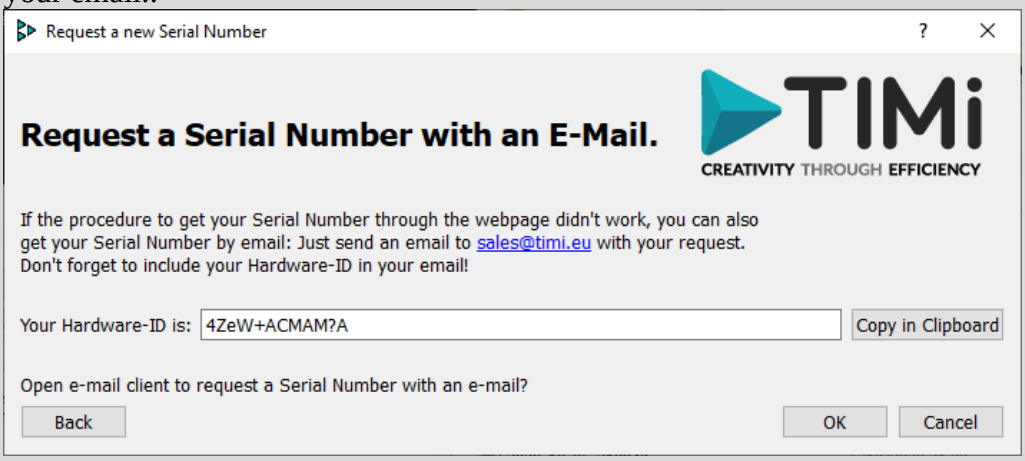
The first time that you run TIMi on a machine without any Serial Number, you are directly redirected to a Wizard that guides you through the procedure to get your Serial Number. The first screen of this Wizard is :



Click OK to open your web browser to fill out a web form to request your serial number.

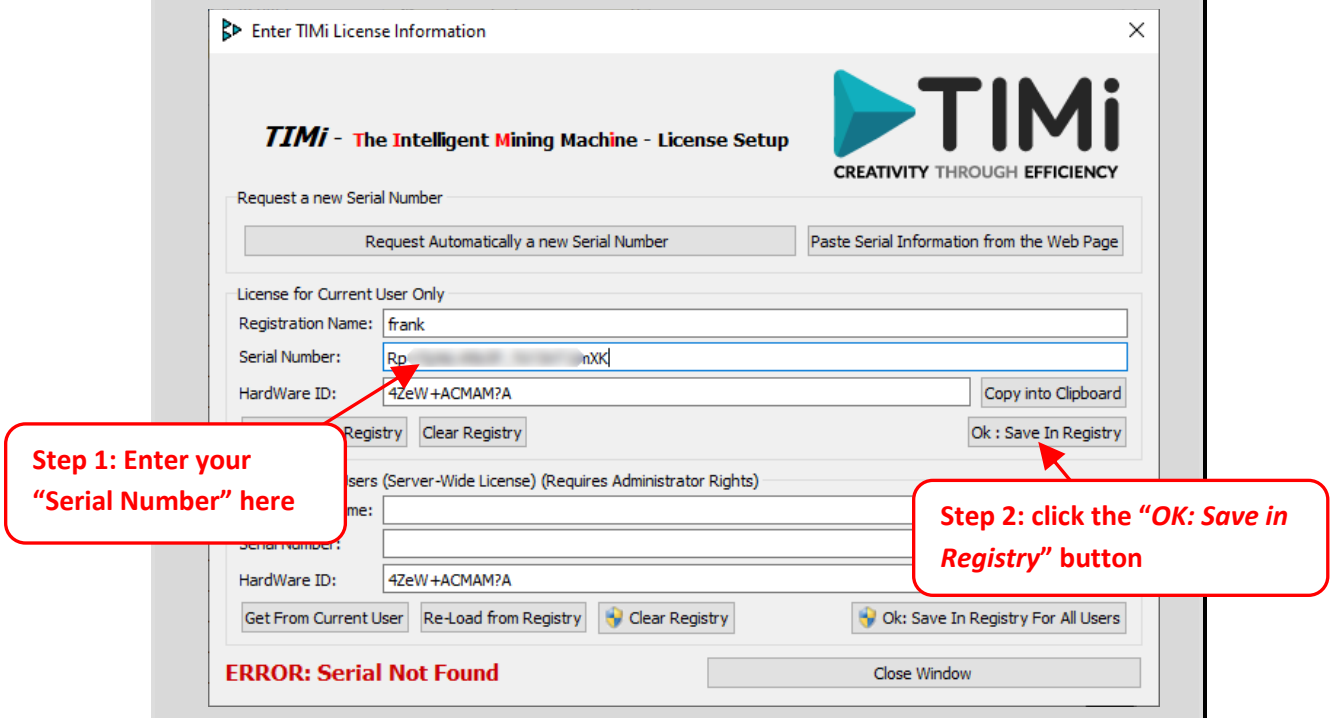
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If you don't have any active internet connection on your server, don't worry: Click on the "Cancel" button to request your "Serial Number" by email. You will arrive on this page which explains what information you need to include in your email.:



Make sure that your email contains your Hardware-ID.
Without your Hardware-ID, we cannot create a "Serial Number".

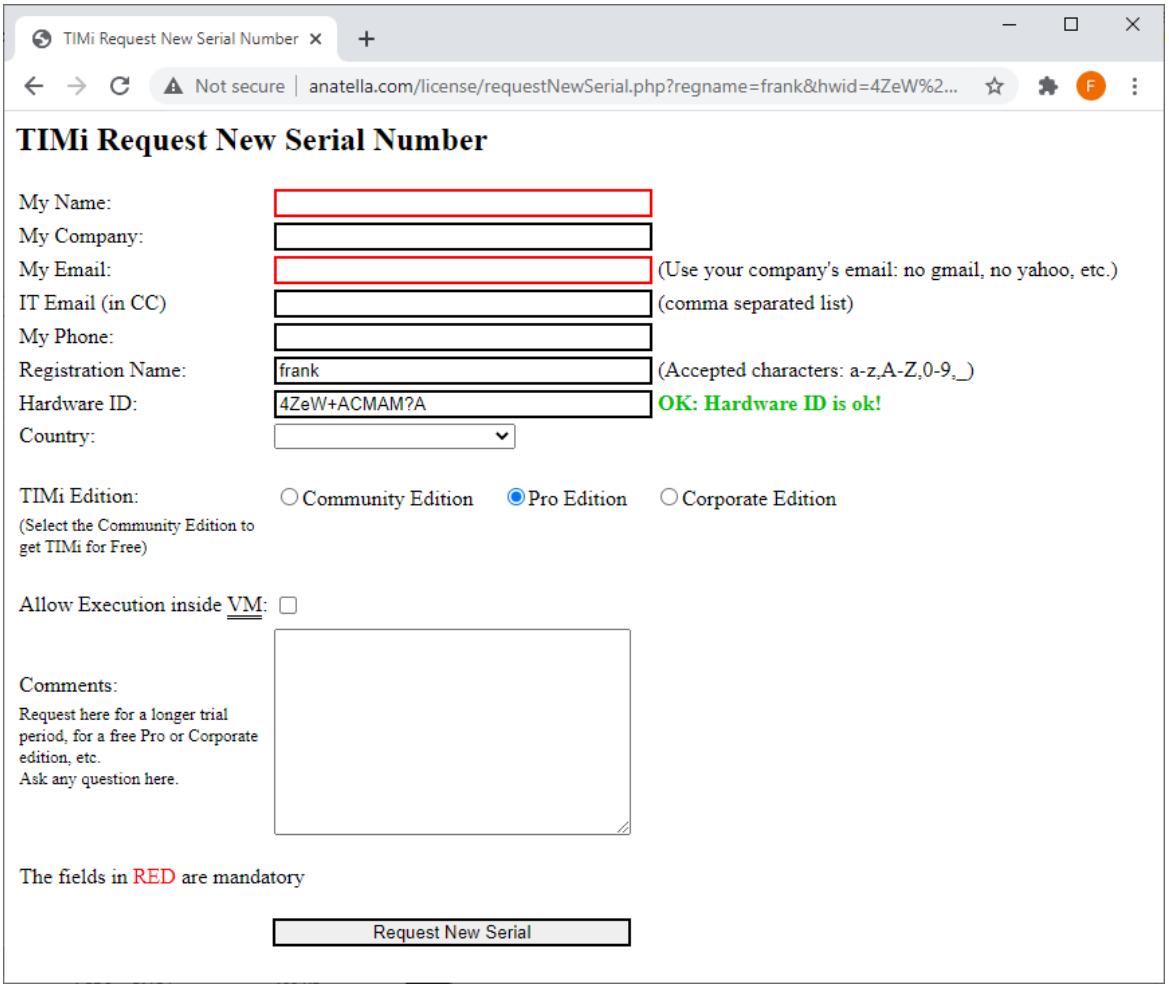
In return for your email, the TIMi Suite representative will send you an email containing your license information (i.e. your "Serial Number").
At this point, please go back to the TIMi “Serial Number Manager” application and enter your "Serial Number" as indicated in the e-mail you received. See the illustration below :



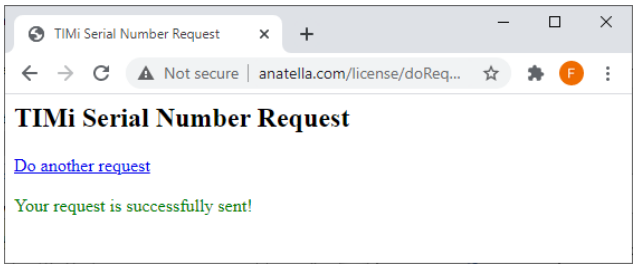
When you have finished encoding your "Serial Number", click the "Ok: Save in Registry" button.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You arrive on the web form to fill in:



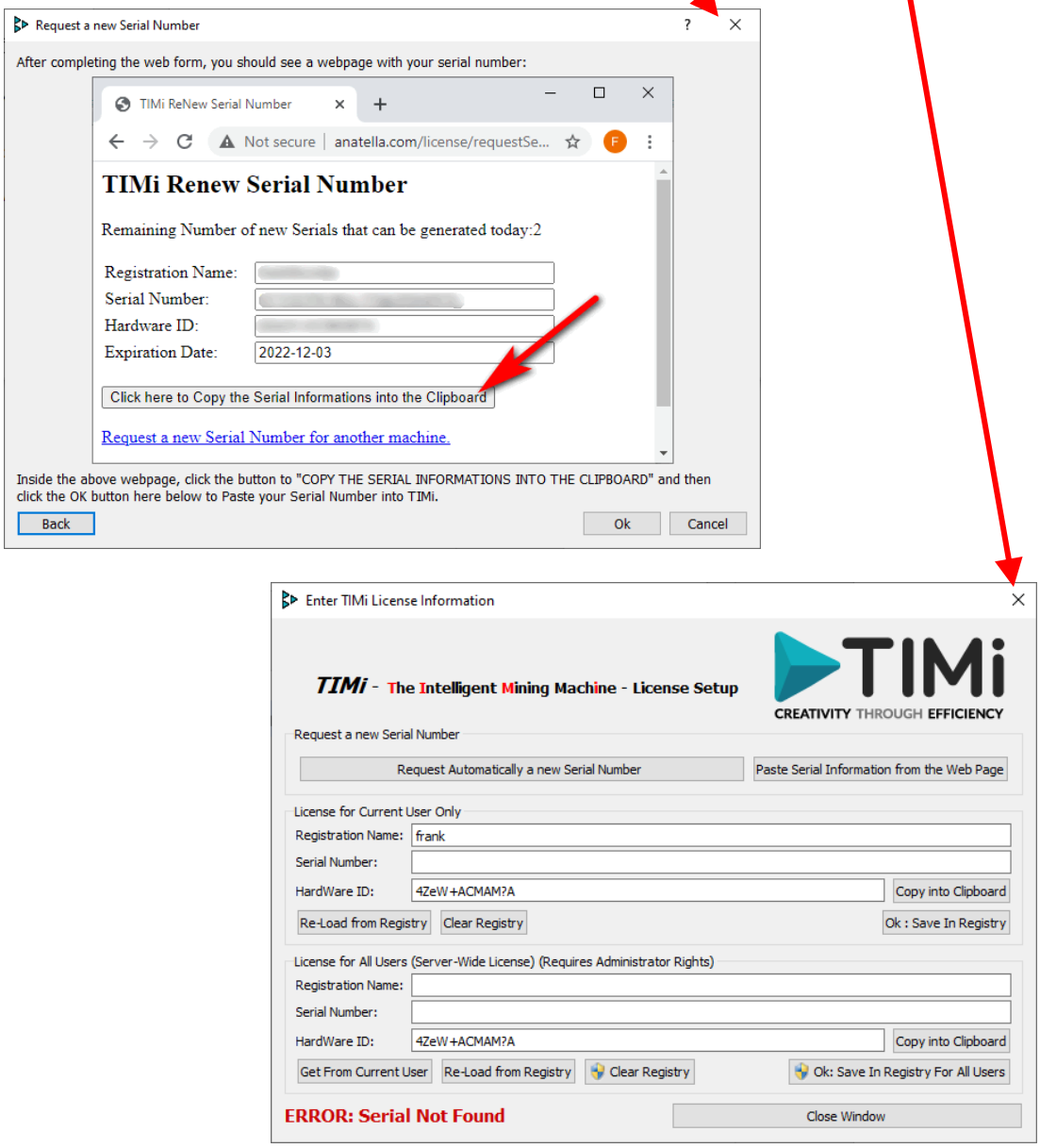
After filling out the form, you arrive on this confirmation page :



Your request for a "Serial Number" has been taken into account. Usually we process these requests within the hour and at most within 24 hours. You will receive an email notification when your Serial Number is available.

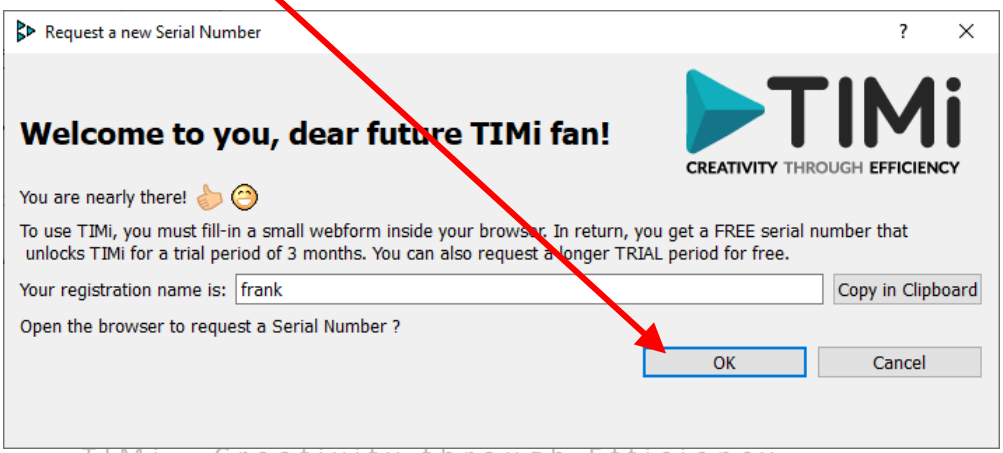
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For the time being, you can close the License Manager: Click here : and then here :



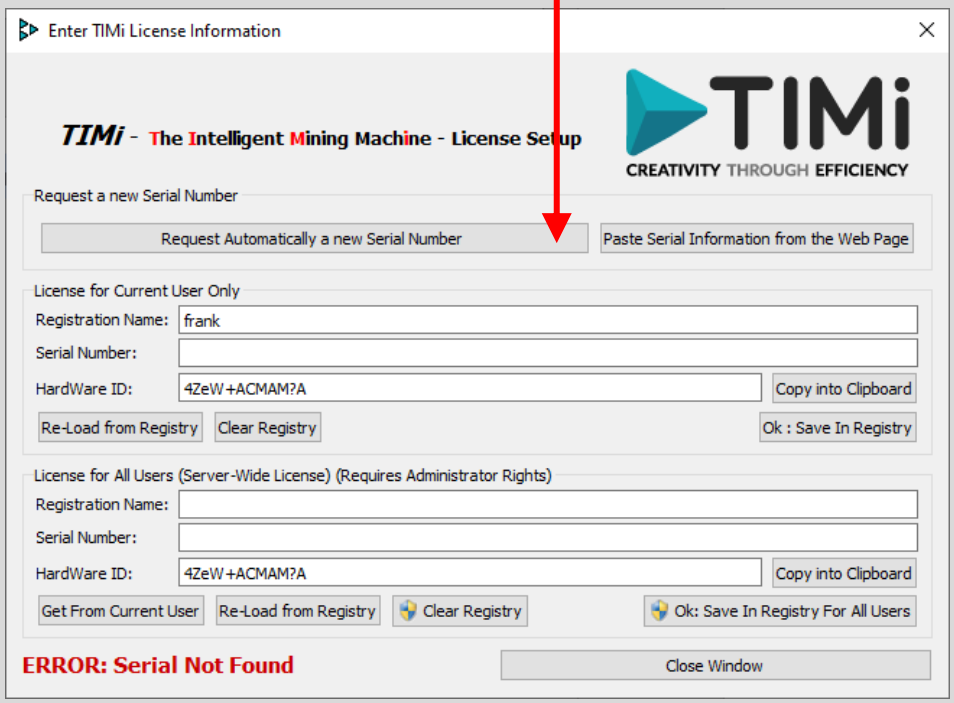
A few minutes have passed and your "Serial Number" is now available.

Open TIMi "*Serial Number Manager*" application again (see the next section 7.4.2 to know how to do that) and click the "OK" button here

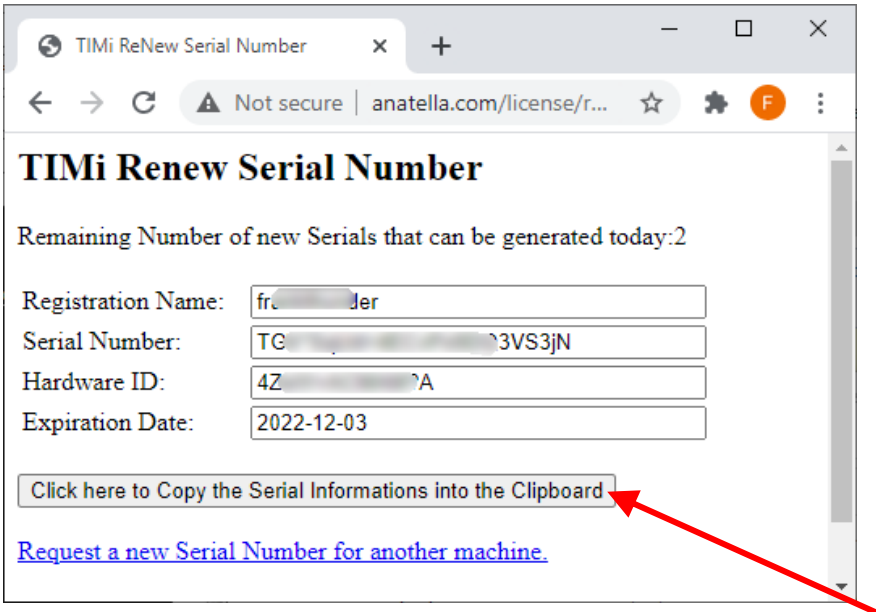


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Alternatively, instead of clicking the "OK" button above, you can also click the "Request Automatically a new Serial Number" button in the main window of the “Serial Number Manager” application:



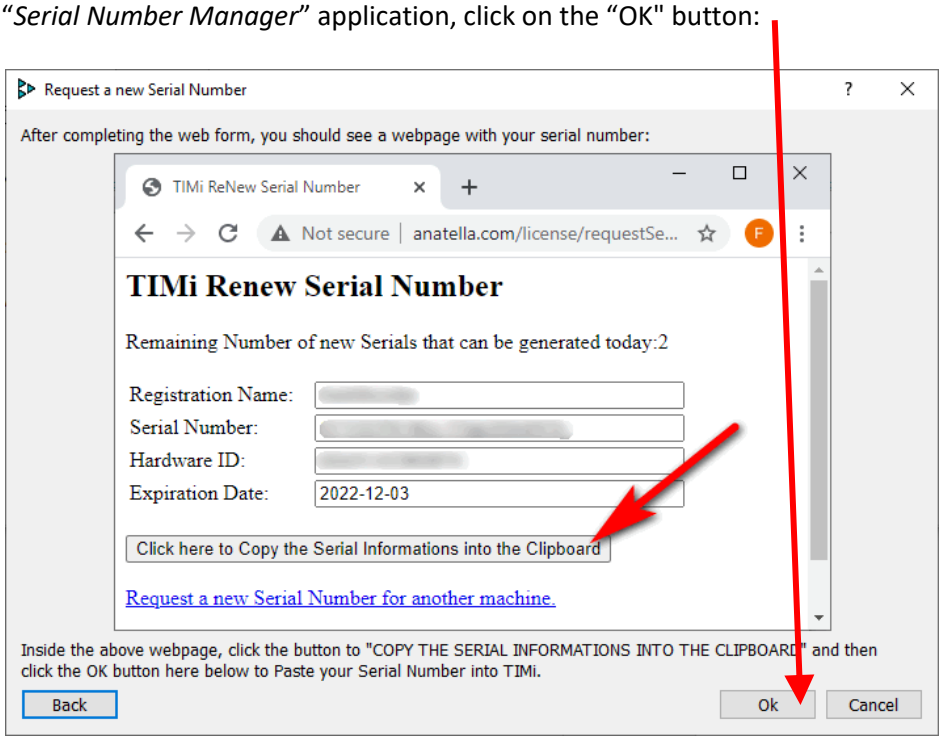
Your internet browser opens again but this time you directly receive your "Serial Number":



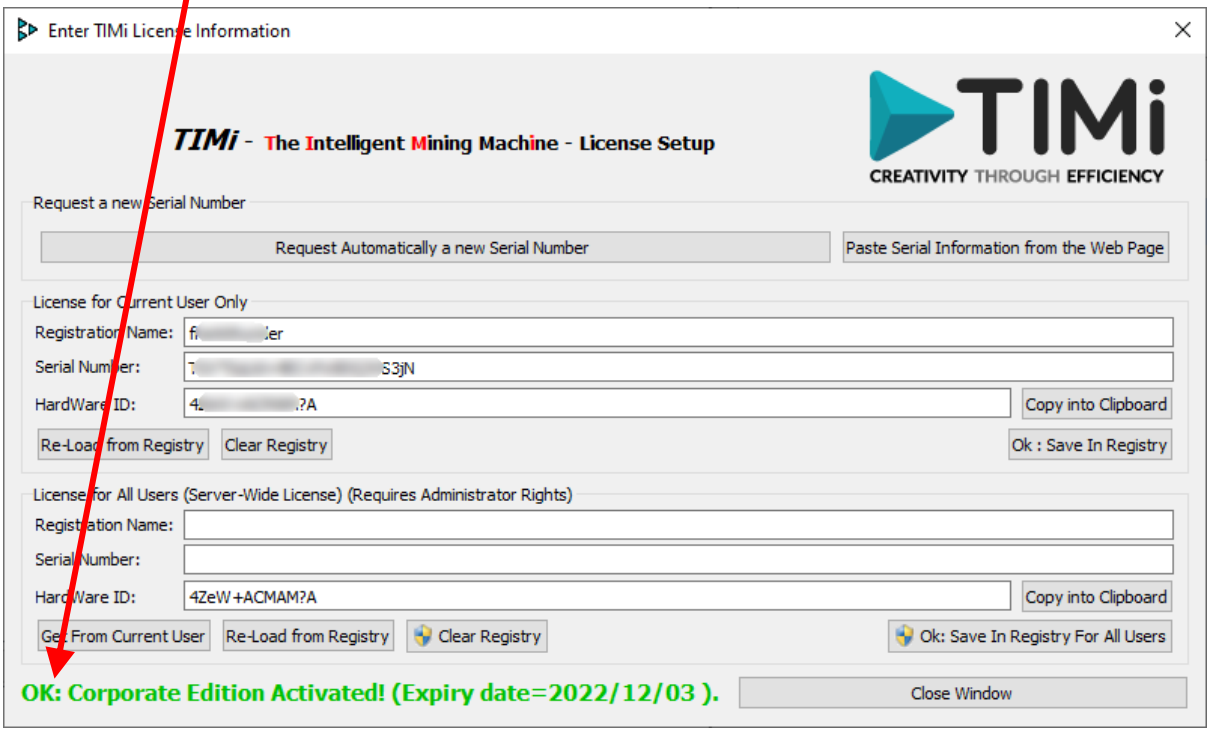
In your internet browser, click on the button "***Copy** the Serial Number into the Clipboard*" here:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Inside the TIMi "*Serial Number Manager*" application, click on the "OK" button:



If your "Serial Number" has been correctly encoded on your machine, you should now see the "OK" message in green here:



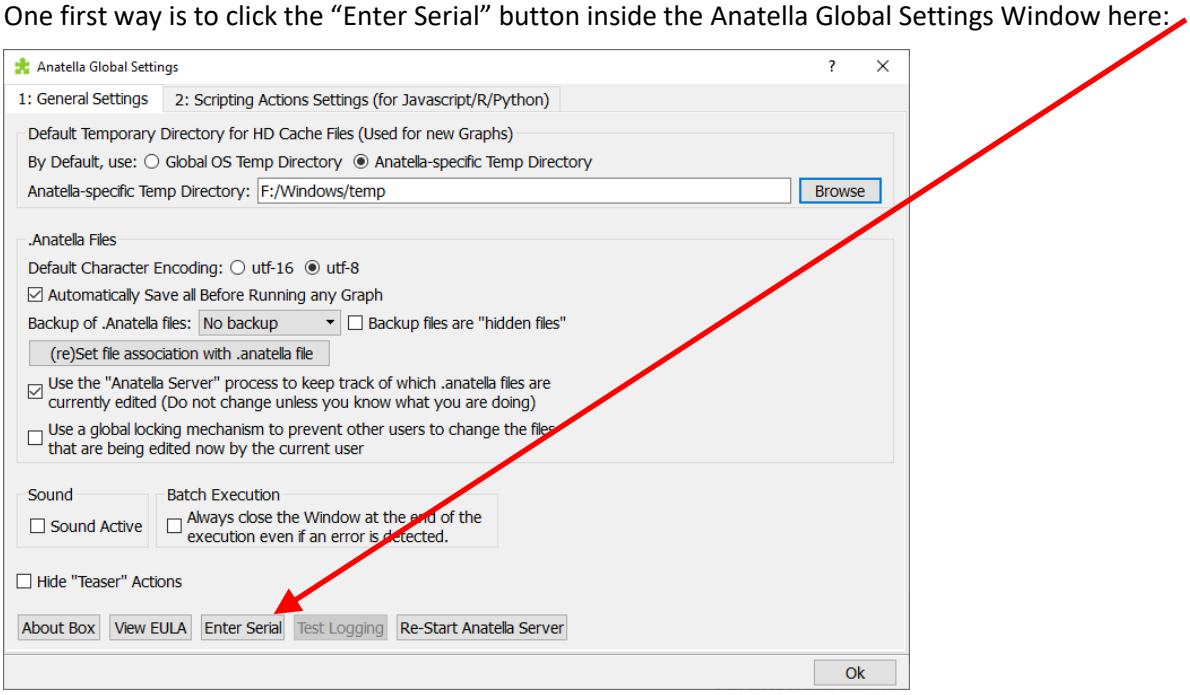
Congratulations !
You now have access to all the features offered by TIMi and Anatella! 😄 👍

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

7.4.2. (re)Open the "Serial Number Manager" application

There are several ways to open the "Serial Number Manager" application.

One first way is to click the "Enter Serial" button inside the Anatella Global Settings Window here:

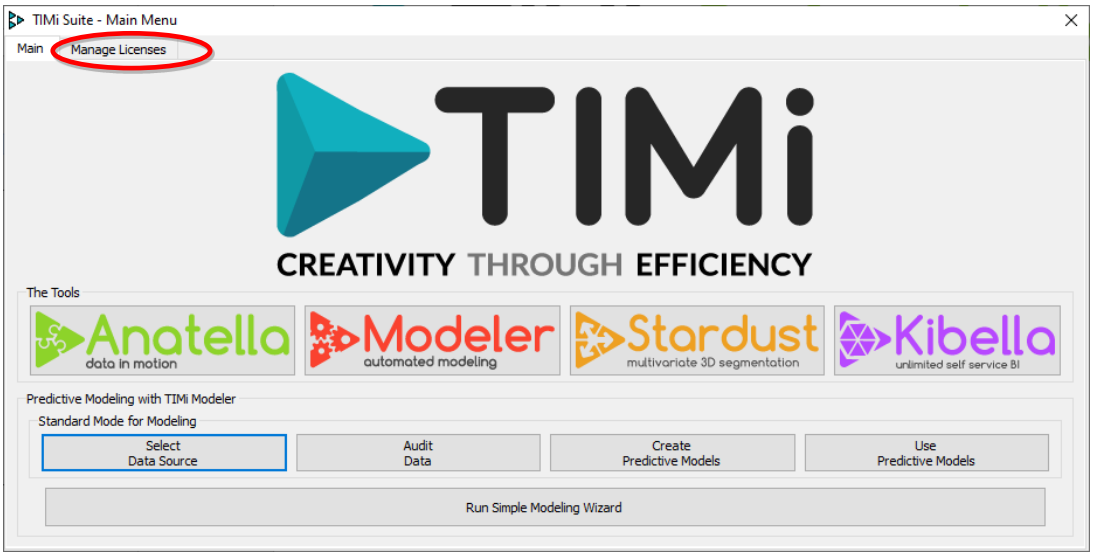


Another way to open the "Serial Number Manager" application is the following:

1. Run the TIMi Main Menu: i.e. double double-click on the TIMi Suite icon:

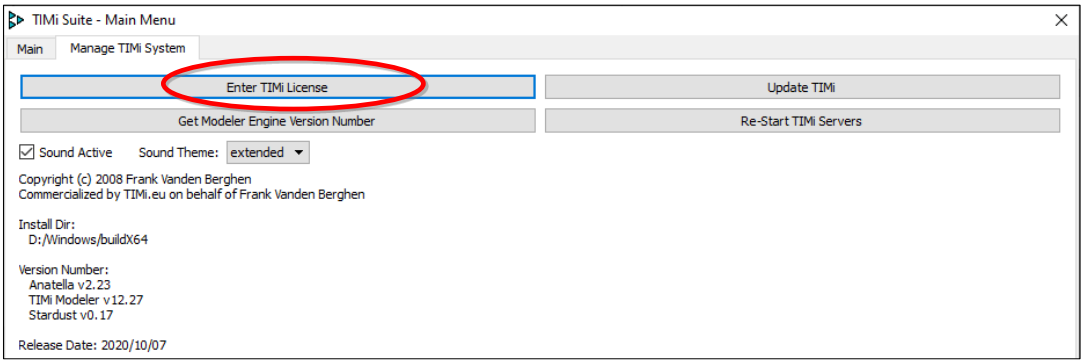
   The "TIMi Suite – Main Window" appears.

2. Inside the "TIMi Suite Main Window", click on the "Manage Licenses" Tab, as illustrated below:



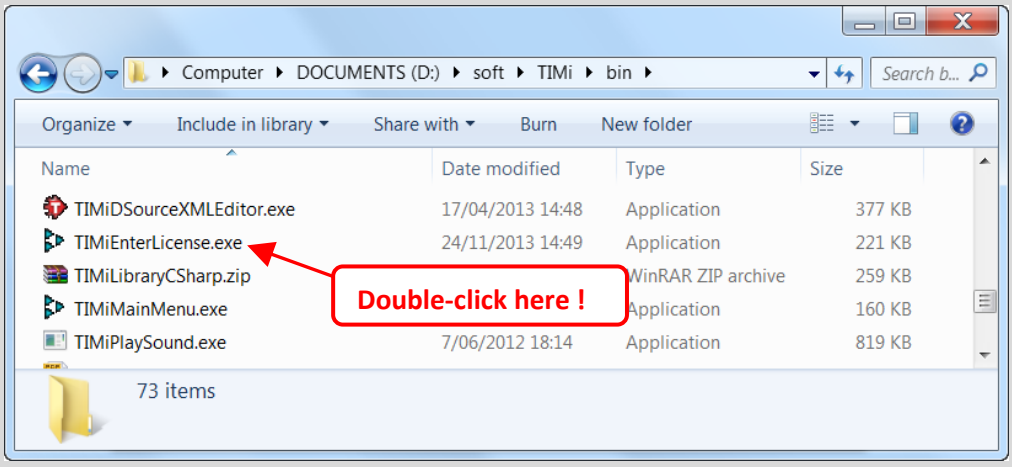
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Click on the "Enter License" button, as illustrated below:



4. The TIMi "*Serial Number Manager*" application should now appear.

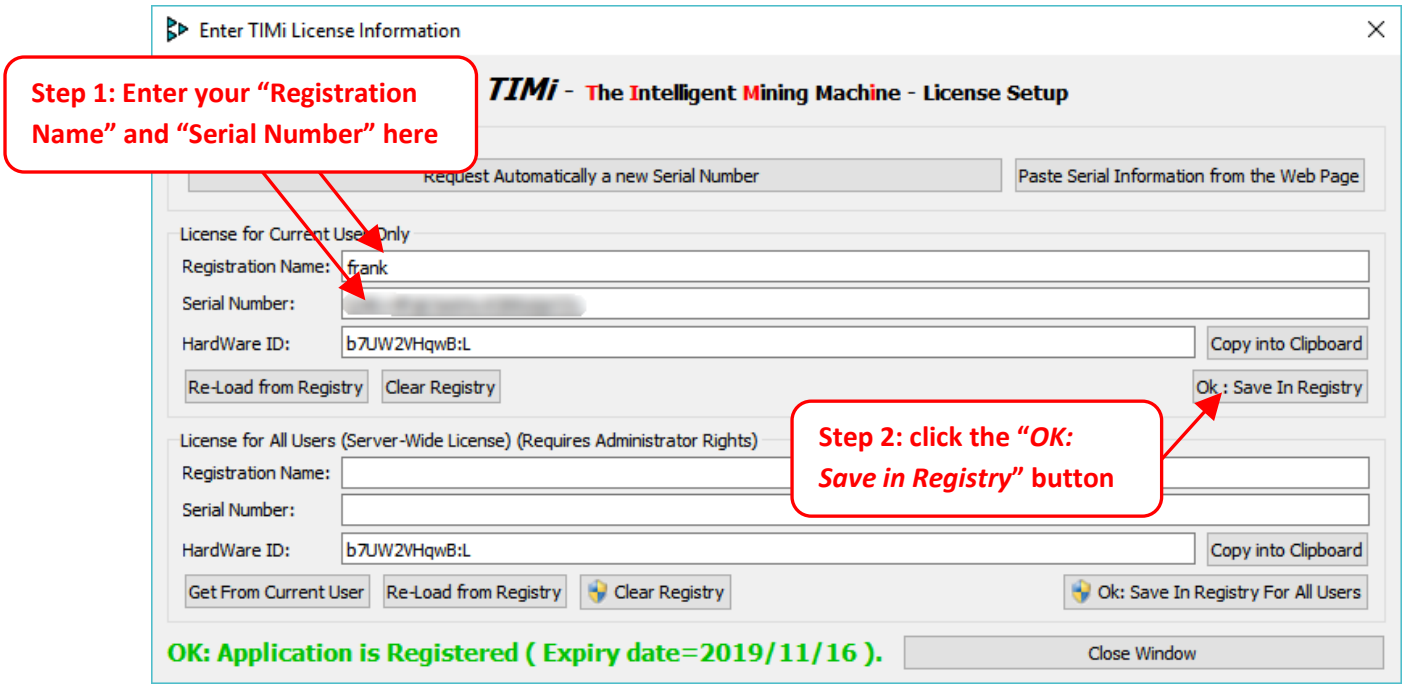
Still, another, alternative, method to open "Serial Number Manager" application is to run the "TIMiEnterLicense.exe" executable located inside the subdirectory "bin" inside the TIMi installation directory (i.e. Typically, you'll run the file
"C:\soft\TIMi\bin\TIMiEnterLicense.exe" or
"C:\Program Files\TIMi\bin\TIMiEnterLicense.exe"):



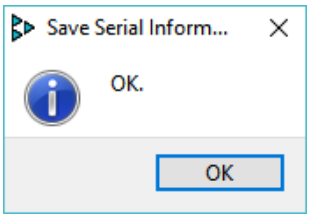
7.4.3. Enter your License Informations (for the Current User Only)

To enter your "Registration Name" and "Serial Number", open the "Serial Number Manager" application (see the previous section about this step) and then:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Once the information is entered, click the "***Ok: Save in Registry***" button. You should see a small confirmation window that pops-up:



7.4.4. Enter a Server-Wide License

Each different Anatella/TIMi user on the server has to enter the "Registration Name" and the "Serial Number" (Everybody can use the same "Registration Name" and "Serial Number" but it must be encoded one time by each user).

If there are many different users on the server, it might be simpler to enter only **one time** the "Registration Name" and the "Serial Number" **for all the users**. This option is unfortunately only accessible if you have administrative rights on your machine.

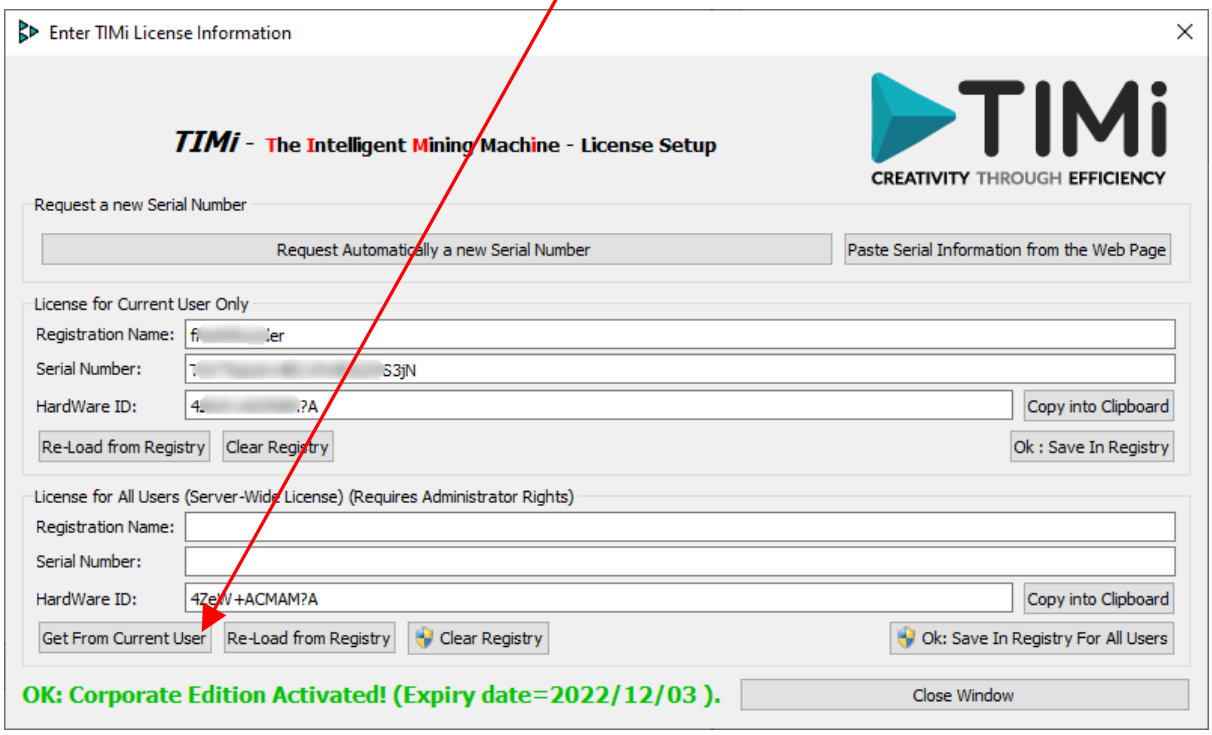
Configuring a Server-Wide License is also required when using Jenkins: See the section 4.8.3.1. for more information about this subject.

To enter the "Registration Name" and the "Serial Number" for all the users of the Server: Go back to the "License Information Window" and follow these steps:

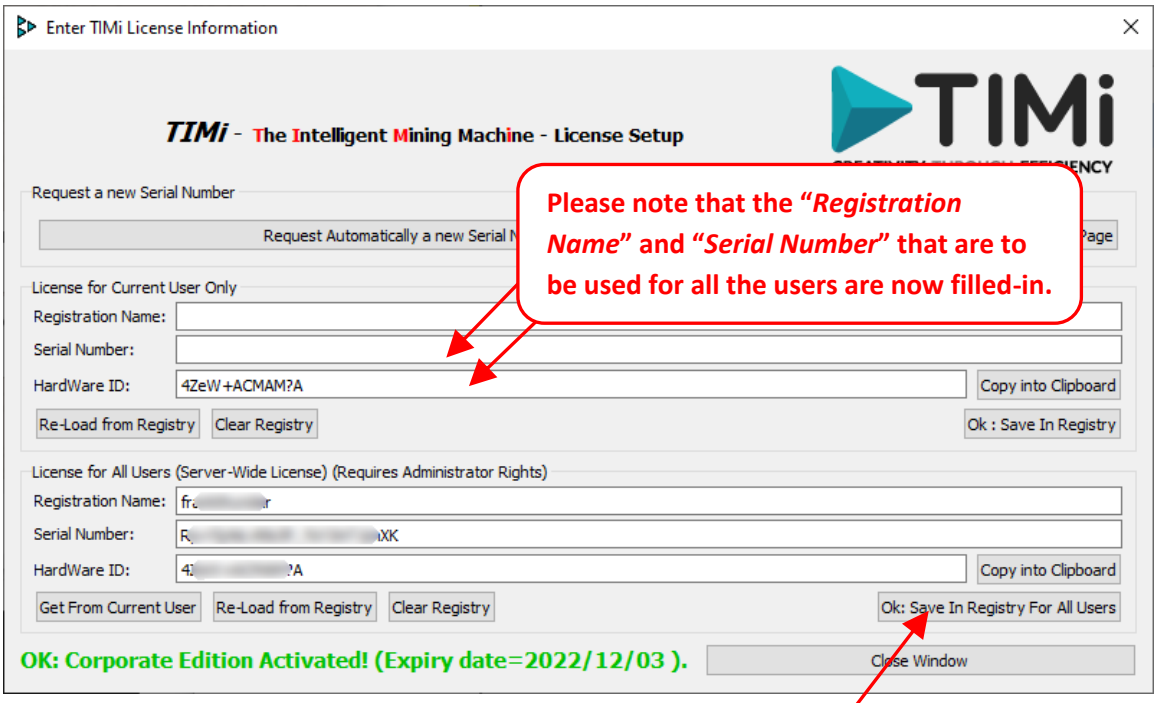
Step 1: Follow the steps described in the section 7.4.1. here above to obtain a functional "Serial Number" for the current user.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Step 2: Click the "Get from Current User" button here:



Step 3: At this point, you should have something that looks like that:



Step 4: Click the "Ok: Save in Registry for All Users" button here: (…when Windows asks for administrative privileges, say "yes").

Congratulations! You have now activated TIMi for all the server users ! 😄 👍

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

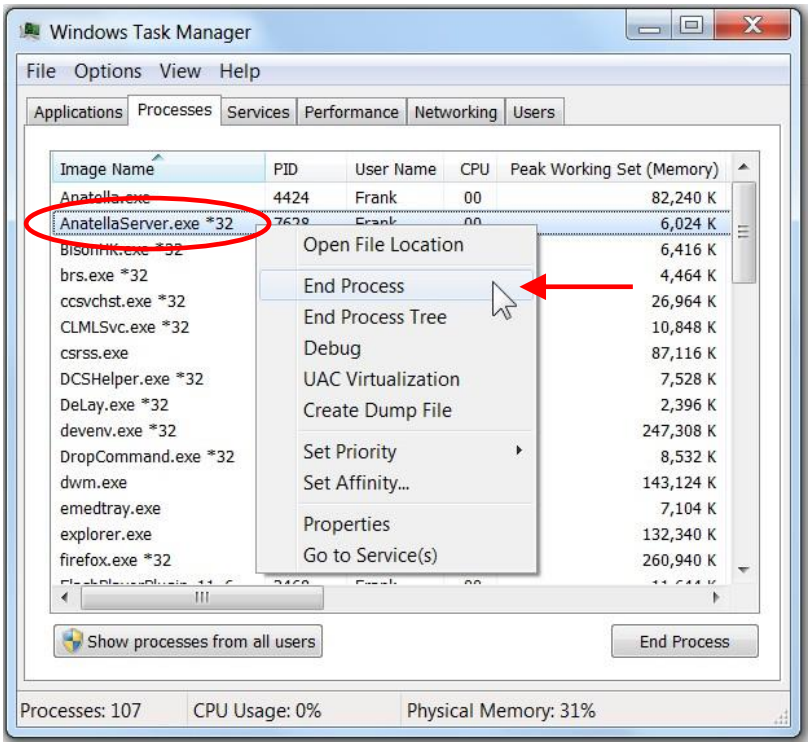
7.5. Panel 1: Restart the Anatella server

When you double-click on an .anatella file that is already opened inside an Anatella window "X", your operating system (i.e. MS-Windows) won't open a new Anatella window but it will rather put "top-most" the Anatella window "X" (This prevents a complete mess where you would be editing the same graph in different windows).

The above mechanism is provided by the "Anatella Server". The "Anatella Server" is a very small executable that runs in the background. It keeps track of which process opened which .anatella file. It also sets the required MS-Windows file associations (e.g. all .anatella files must be associated with the "AnatellaServer.exe" process).

If the "Anatella Server" is frozen, you won't be able to open any new .anatella file for edition. In such situation, close all opened .anatella files and kill the "Anatella Server".

Alternatively, you can also "manually" kill the "AnatellaServer.exe" process using the "MS-Windows Task Manager":



If no "AnatellaServer.exe" process is running, MS-Windows will transparently (re-)launch a new "AnatellaServer.exe" process as soon as you open an .anatella file. Usually, you don't have to worry about the "Anatella Server": everything is automated and transparent to the user.

7.6. Panel 1: Anatella SQL Logging

When your data scientists are managing sensitive data stored inside a database, you need to closely monitor their actions to prevent any catastrophic event that might jeopardize the confidentiality of the data (or even damage your database!). To handle this delicate situation properly, you might be interested in logging all the SQL commands that were sent to your databases. These "SQL logs" are

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

usefull to make data scientist accountable for all their actions that are related to your sensitive databases. This section explains how to activate the "SQL logging" mechanism included inside Anatella.

A typical sequence found inside these logs is the following 4 rows:

ID	ProgramID	PID	Host	User	Graph	Date	SeverityCode	ActionName	ActionID	MessageText	Optional
17	0	13092	COGITE	COGITE\Frank	F:/testLogging.anatella	20200405 17:58:33	3	readODBC	1	Starting (pin=0)	
18	0	13092	COGITE	COGITE\Frank	F:/testLogging.anatella	20200405 17:58:33	3	readODBC	1	43 Columns:select * from income	
19	0	13092	COGITE	COGITE\Frank	F:/testLogging.anatella	20200405 17:58:33	3	readODBC	1	10 Rows Processed	
20	0	13092	COGITE	COGITE\Frank	F:/testLogging.anatella	20200405 17:58:33	3		0	Stopped	

Here is a small explanation on the meaning of each of the 4 above rows:
Row 1: The user clicked on the Pin 0 of the "readODBC" action (with id=1) to start the execution of the Anatella graph named "f:/testLogging.anatella".
Row 2: The "readODBC" action (with ActionID=1) started the execution of the SQL command "*select * from income*". This returned a table with 43 columns.
Row 3: The "readODBC" action (with id=1) extracted 10 rows out of the database.
Row 4: The execution of the Anatella graphed stopped at 17:58:33 on the 2020/4/5.

The columns of the "AnatellaLogs" table are:

- **ID:** A primary key that uniquely identifies the row
- **ProgramID:** The ID of the tool that initially created the row.
  The ID's are: Anatella=0, Modeler=1.
- **PID:** This is the "Process ID": it's an identifier that uniquely identifies the Anatella process that created the row (in the possibly case when there are several Anatella instances running simultenously on the same server/host).
- **Host:** This is the (Windows) name of the server that generated the row inside the AnatellaLogs table.
- **User:** This is the (Windows) name of the ysed that generated the row inside the AnatellaLogs table.
- **Graph:** This is the filename of the .anatella graph file that generated the row inside the AnatellaLogs table.
- **Date:** Self-explaining.
- **SeverityCode:** This is a number that refers to this table:

SeverityCode	Meaning
0	SUCCESS
1	ERROR
2	WARNING
3	INFO
4	DEBUG

- **ActionName:** The name of the Action inside your .anatella graph file that generated the row inside the AnatellaLogs table.
- **ActionID:** The ID of the Action inside your .anatella graph file that generated the row inside the AnatellaLogs table.
- **MessageText:** A description of the event represented by the current row inside the AnatellaLogs table.
- **Optional:** empty/null (possible usage in future extension).

Inside Anatella, all the SQL commands are logged. This means that Anatella is logging each execution

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

of the readDOBC action, the upsertODBC action, the createTable action, the teradataWrite action and the OracleOCI action.

The steps required to activate the logging mechanism included inside Anatella are:

1. Request a serial number (i.e. a license) where the "SQL Logging" option is enabled. Please contact your TIMi sales representative to request your new serial number with this special option. The sections 7.4.2., 7.4.3 and 7.4.4 explain how to enter your new serial number.

2. Create inside your database a table named "AnatellaLogs": i.e. run the following SQL command inside your database:

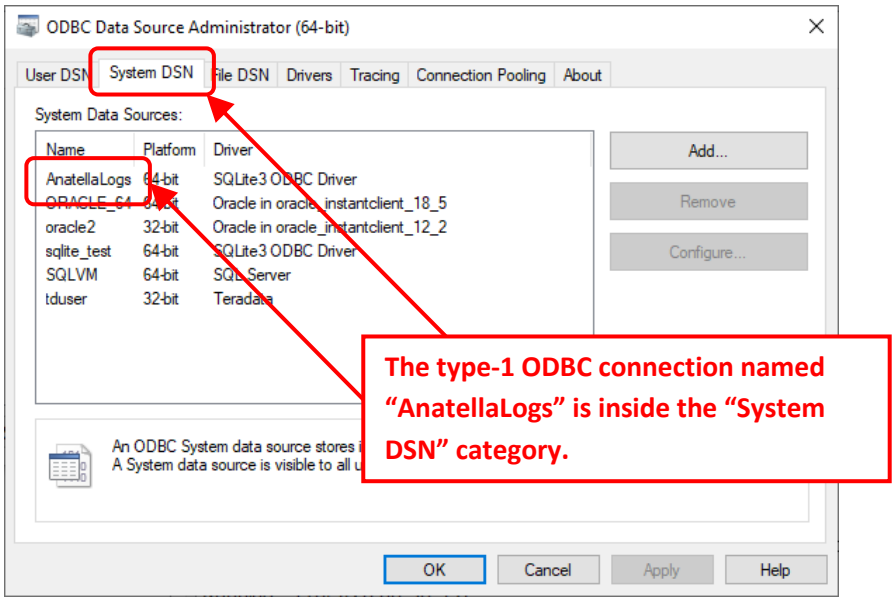
```
CREATE TABLE AnatellaLogs (
      ID             INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
      ProgramID      INTEGER NOT NULL,
      PID            INTEGER NOT NULL,
      Host           nvarchar(100),
      User           nvarchar(100),
      Graph          nvarchar(300),
      Date           nvarchar(20),
      SeverityCode   INTEGER NOT NULL,
      ActionName     nvarchar(20),
      ActionID       INTEGER NOT NULL,
      MessageText    TEXT NOT NULL,
      Optional       TEXT
);
```

Inside the above "CREATE TABLE' statement, the columns "ID" and the column "Optional" are not required: you can omit/remove them.

3. Create inside your database a database-user named "AnatellaLogs" that can only "INSERT" new rows inside the "AnatellaLogs" table. In particular, the "AnatellaLogs" user should not be authorized to DELETE ROWS or UPDATE ROWS from the "AnatellaLogs" table (otherwise an ill-intentioned user might be tempted to corrupt the "AnatellaLogs" table to "*cover its traces*").

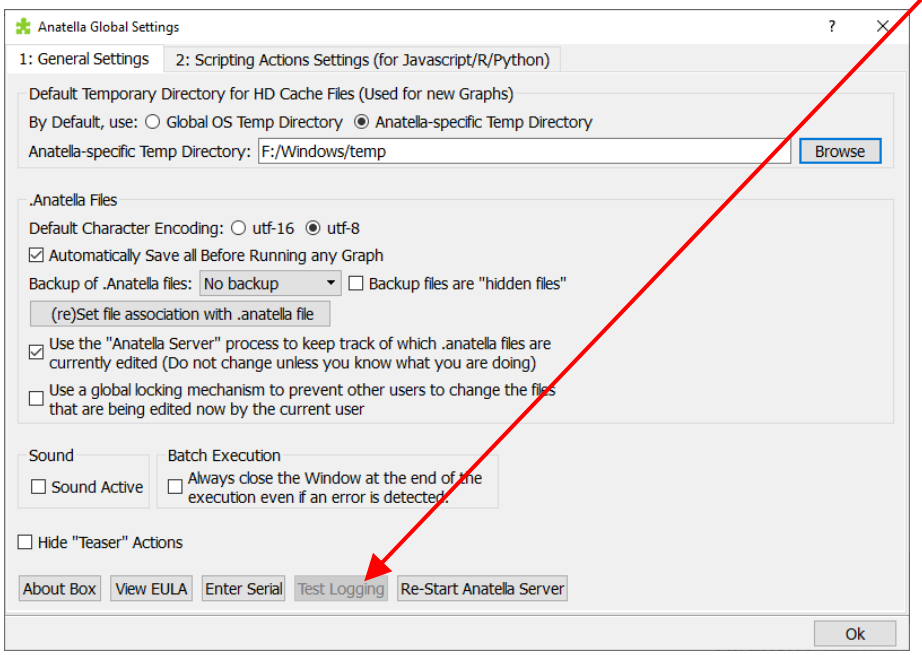
4. Create on the local machine a type-1 ODBC connection to your database that is named "AnatellaLogs". This is important that this new type-1 ODBC connection is inside the "**System DSN**" category because this prevents a non-adminstrative user to temper with the ODBC connection:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Please, pay attention to the ODBC manager: If you are using a 64-bit version of Anatella, you need to use the 64-bit ODBC manager. Likewise, if you are using a 32-bit version of Anatella, you need to use the 32-bit ODBC manager. You'll find more details on this subject inside the section 5.1.6.1.

5. Test your setup: To run a simple quick test, you can click the "Test Loggin" button here:



If the "Test Logging" button is disabled (i.e. grayed out, as in the example above), it means that you forgot to request a serial number with the "SQL Logging" option that is enabled (see step 1).

After clicking the "Test Logging" button, you should see a new row inside the "AnatellaLogs" table inside your database. This new row looks like this:

ID	ProgramID	PID	Host	User	Graph	Date	SeverityCode	ActionName	ActionID	MessageText	Optional
21	0	17676	COGITE	COGITE\Frank	Test	20200405 18:25:10	3		0	Test Logging	

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For your convenience, to run a quick test, you'll find a sqlite database with a proper "AnatellaLogs" table already pre-configured inside the file "<TIMi_install_dir>/bin/AnatellaLogs.sqlite". This sqlite database is only useful for a quick test: i.e. it should NEVER be used in a production setting because of one limitation of the sqlite database engine: i.e. The sqlite database engine does not handle user rights: i.e. All sqlite users can DELETE ROWS from a sqlite database (and, in our case, this should be forbidden: For more details, see the step 3, here above). A good database engine to store the "AnatellaLogs" table is MS-SQLServer, Oracle, Postgres, etc. You'll find the ODBC drivers for SQLite databases here: http://download.timi.eu/ODBC/ODBC_drivers_SQLite/

7.7. Panel 2: Javascript/R/Python Revision Control

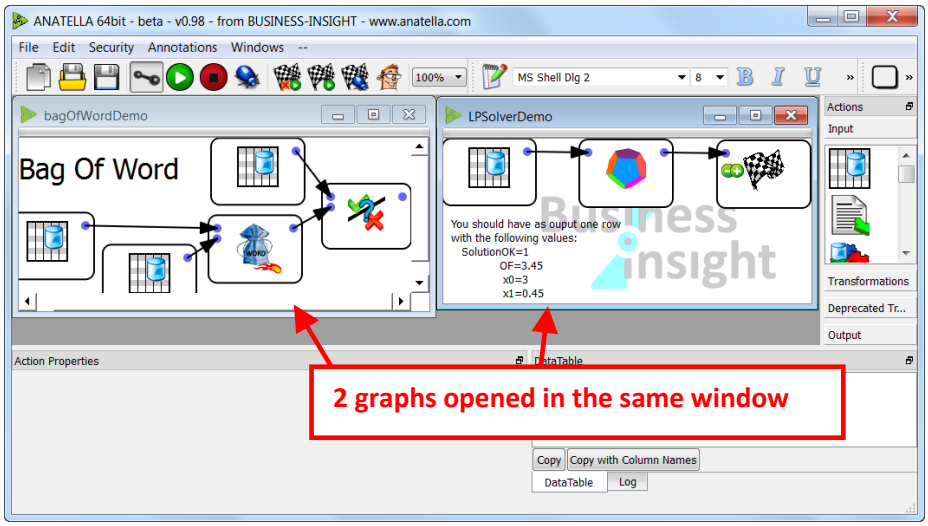
See section 9.8. (and 9.7) for more information about this subject.

8. Anatella best practices

This section details the good practices that improve your experience using Anatella. It's strongly advised to follow the "best practices" given in this section for the most efficient Anatella usage.

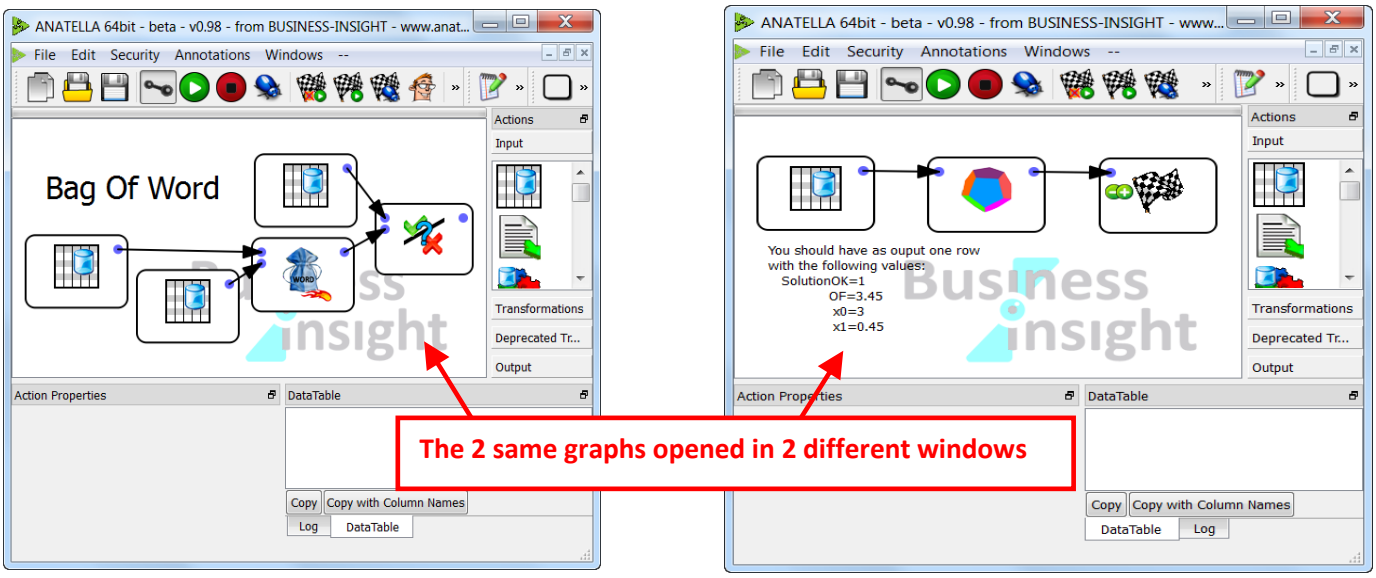
8.1. One graph per Anatella window

You should avoid opening many different Anatella graphs inside the same window: For example: This is BAD:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This is GOOD:



8.2. Run each graph inside its own window

You can run several Anatella graphs at the same time, but each graph must be inside its own window (i.e. in its own process).

8.3. Quickly stop/abort a graph execution

To stop the execution of an Anatella graph, you use the button on the toolbar. When you click the button, Anatella instantaneously writes inside the Log window the following text:

WARNING: User Abort.

…but this does not mean that the execution of your graph stopped! This only means that Anatella noticed that you **requested** stopping the graph execution. The execution is properly stopped only when you see the following message:

Run has been stopped unexpectedly! (finished at 29/1/13 09:59:15 after 0.12 minutes - Peak Memory Consumption: 1523 MB)

It's quite common to be forced to wait for a quite long time before Anatella completely stops the graph execution and before seeing the final message "***Run has been stopped***".

To avoid waiting, you can simply click the button in the top-right corner of the window: This will **instantaneously** close the window and you can thereafter **instantaneously** re-open it. You'll notice that all your settings have been saved (Anatella automatically saves your graphs before every run) and you can directly continue to work on your graph.

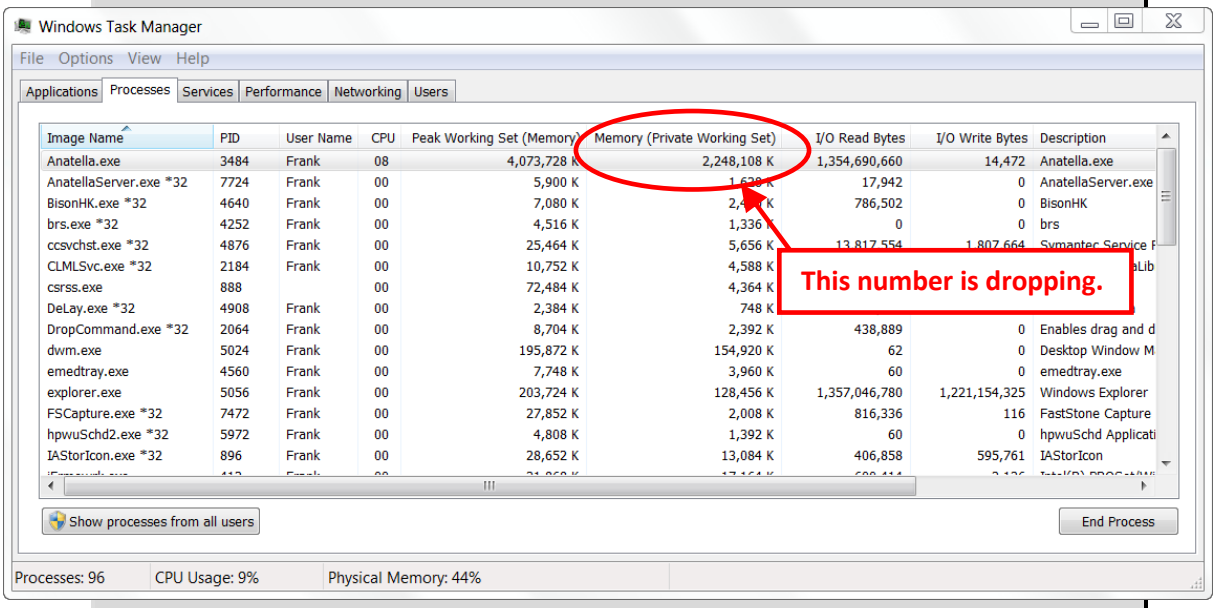
If you try to edit your graph when it's still executing (i.e. before seeing the final message "***Run has been stopped***"), unexpected results & errors might occur.

Always wait until complete stop before editing your graph.

How is it possible that you can stop the execution of some Anatella graphs instantaneously and, for some other graphs, it can take nearly 1 minute?

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When you stop the execution of an Anatella graph, Anatella asks to MSWindows to "free" all the RAM memory that it is currently using. More precisely, when stopping a graph, you can see the memory consumption of the "anatella.exe" process that is dropping:



When the memory consumption returns back to a value around 180 MB, it means that all the memory used for the graph-execution is completely "free'd" and Anatella can safely display the message "***Run has been stopped***" (i.e. The memory consumption of the "anatella.exe" process when idle is around 160-180 MB).

Some Actions require a large amount of RAM memory to run (see section 5.3.2.7 about this subject). An Anatella graph that contains these Actions requires a longer time to stop.

When you click the button, MSWindows free's all the memory consumed by the "Anatella.exe" process: it's a lot faster because MSWindows can reclaim "in bulk" the complete memory used by the process (In opposition to the "normal" stop when there are always 60-80MB of memory that remains in use after a "stop").

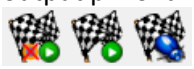
8.4. Avoid losing time when freezing

Anatella is a pretty stable software but it can happen that it "freezes" when starting a very complex data transformation. Hopefully, this only happens when (re-)starting a graph execution in "interactive mode" (aborting a graph and running it again several times might let the engine in an incoherent state). Before executing any graph, Anatella always saves transparently on the hard drive your data transformation graphs so that you can quickly re-open them without losing any of your work, should something wrong happen.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This means that, if you see that something is not working as expected (i.e. Anatella is unexpectedly "freezing"), you can always click the button and re-open quickly your data transformation graph without losing time.

There are several ways of launching a graph execution: You can:

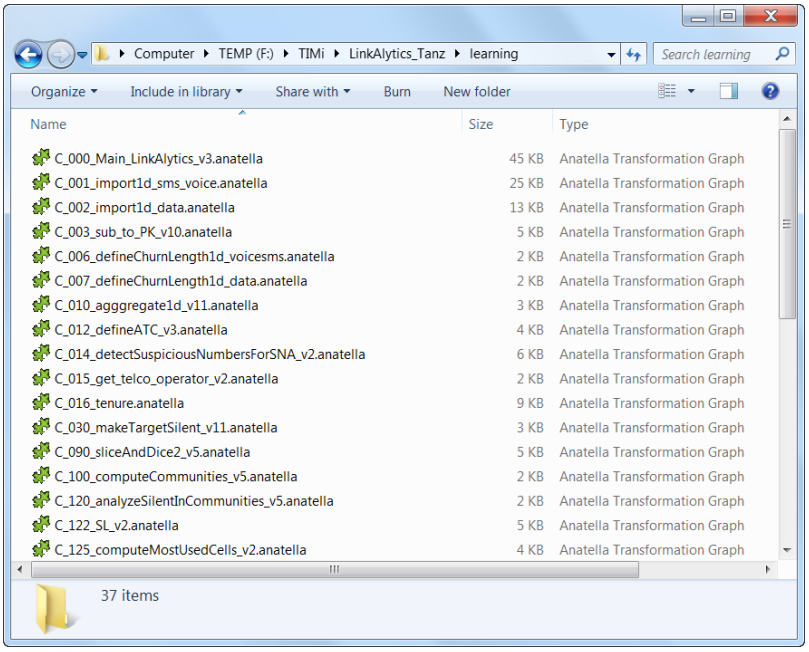
- Click the output pin of an Action.
- Click the  buttons inside the toolbar.
- Click the "Refresh from pin" button inside the "Column chooser" window.
- Click the "dry-run" button of the CreateTable Action.

If Anatella freezes when launching a graph execution, click the button and re-open (instantaneously) your graph.

8.5. Quickly close and re-open Anatella graphs

To be able to close an Anatella Graph (using the button) and re-open it (very) quickly, it's better to always have a "File Explorer Window" opened on the directory containing all the Anatella files that are composing your project.

Closing and re-opening an Anatella graph very quickly is very useful when you want to stop very quickly the execution of your graph (or when you noticed that Anatella is "freezing").
For example, I am always keeping open in the background this "File Explorer Window":



...that contains all the Anatella graphs that I am currently working on. I only have to double-click an .anatella file to (re-)open it instantaneously.

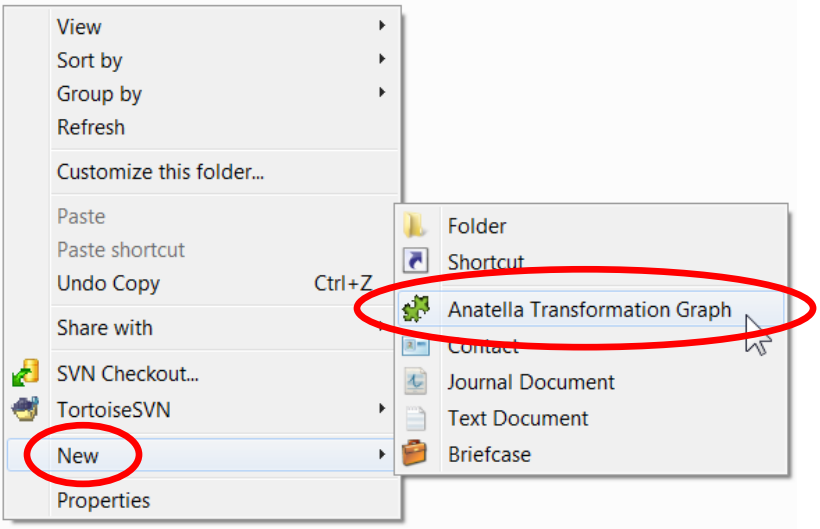
Anatella is closely integrated with the Windows file Explorer:

- If you double-click on an .Anatella file that is already opened inside an Anatella window named "X", MSWindows won't open a new Anatella window but it will rather put "top-most" the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella window named "X" (This prevents a complete mess when you are editing the same graph in different windows).

- You can manage all your Anatella graphs from the Windows file Explorer: For example: You can rename, copy and even, create new graphs directly inside the Windows file Explorer:



- You can drag & drop a (compressed) CSV/Text file from the MSWindow File Explorer into an Anatella Window: This will automatically add a new "CSV File Reader" Action to the current Anatella Graph. Anatella automatically adapts the settings of this new "CSV File Reader" Action (i.e. the filename parameter) so that you can directly see the content of your (compressed) CSV/Text file inside the "Data preview" window.

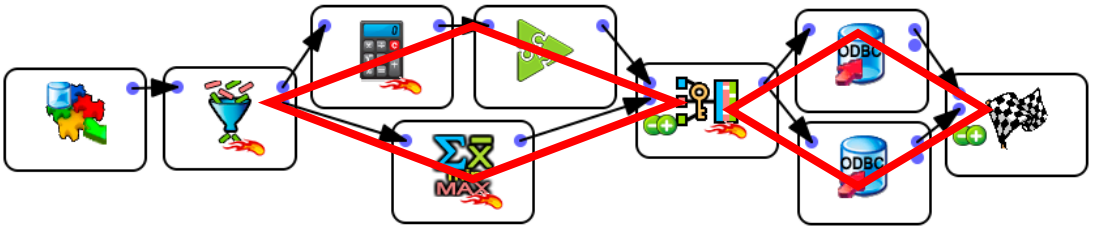
- You can drag & drop a .gel_anatella file from the MSWindow File Explorer into an Anatella Window: This will automatically add a new "Anatella GelFile Reader" Action to the current Anatella Graph. Anatella automatically adapts the settings of this new "Gel File Reader" Action (i.e. the filename parameter) so that you can directly see the content of your .gel_anatella file inside the "Data preview" window.

- You can drag & drop many types of files from the MSWindow File Explorer into an Anatella Window: You can drag & drop: Columnar Gel files (.cgel_anatella), XML files, HTML files, JSON files, SQLite files (.sqlite or .db3), SAS files (.sas7bdat), SPSS files (.sav or .por), EDI files, X12 files, Antenna Files (.196), Coda files (.cod or .coda). This will automatically add the corresponding action.

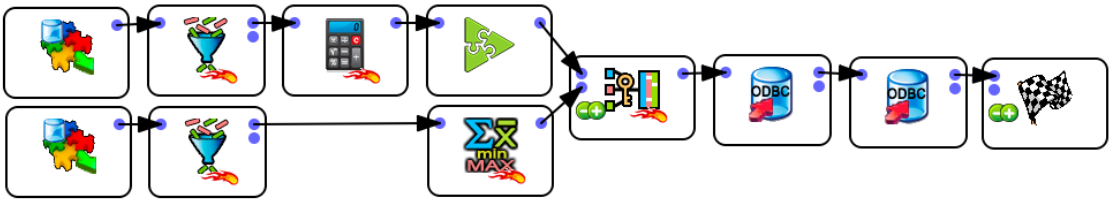
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

8.6. Avoid "diamond shapes" in graphs

Avoid designing Anatella graphs with diamond shape in them: For example:
This is BAD:

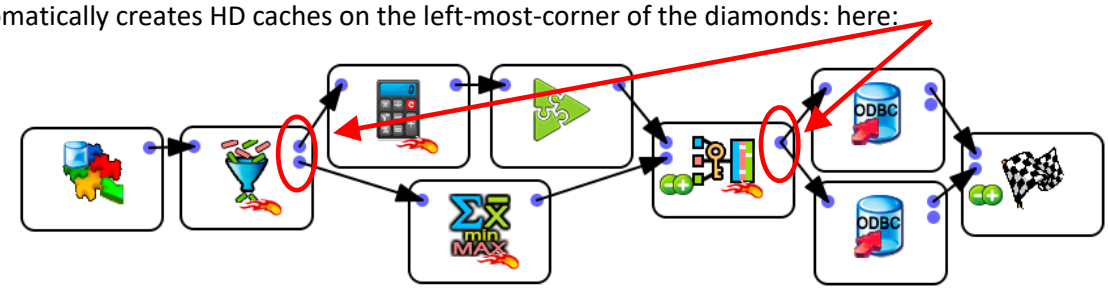


This is GOOD (and it's equivalent to the above BAD graph):



(Please note that we had to duplicate the FilterRows Action in order to remove the left diamond shape).

When there are some diamond shapes inside a graph, Anatella is forced to create a temporary HD cache containing all the data. The extra I/O's performed to create (and thereafter read) this HD cache cost a large amount of time and disk space (and should therefore be avoided). More precisely, Anatella automatically creates HD caches on the left-most-corner of the diamonds: here:



8.7. Use sampling wisely

Most of the time, to have a more responsive interface, you should use sampling when designing new Anatella graphs that are manipulating large datasets (especially when your graph contains some

Sort Action). See section 5.5.15. for more information about this subject.

8.8. Forcing the "Refresh from PIN" button to return some columns

Sometime, when you click the "Refresh from PIN" button of the "Column Chooser" window, Anatella fails to display the required columns. This can happen for several reasons:

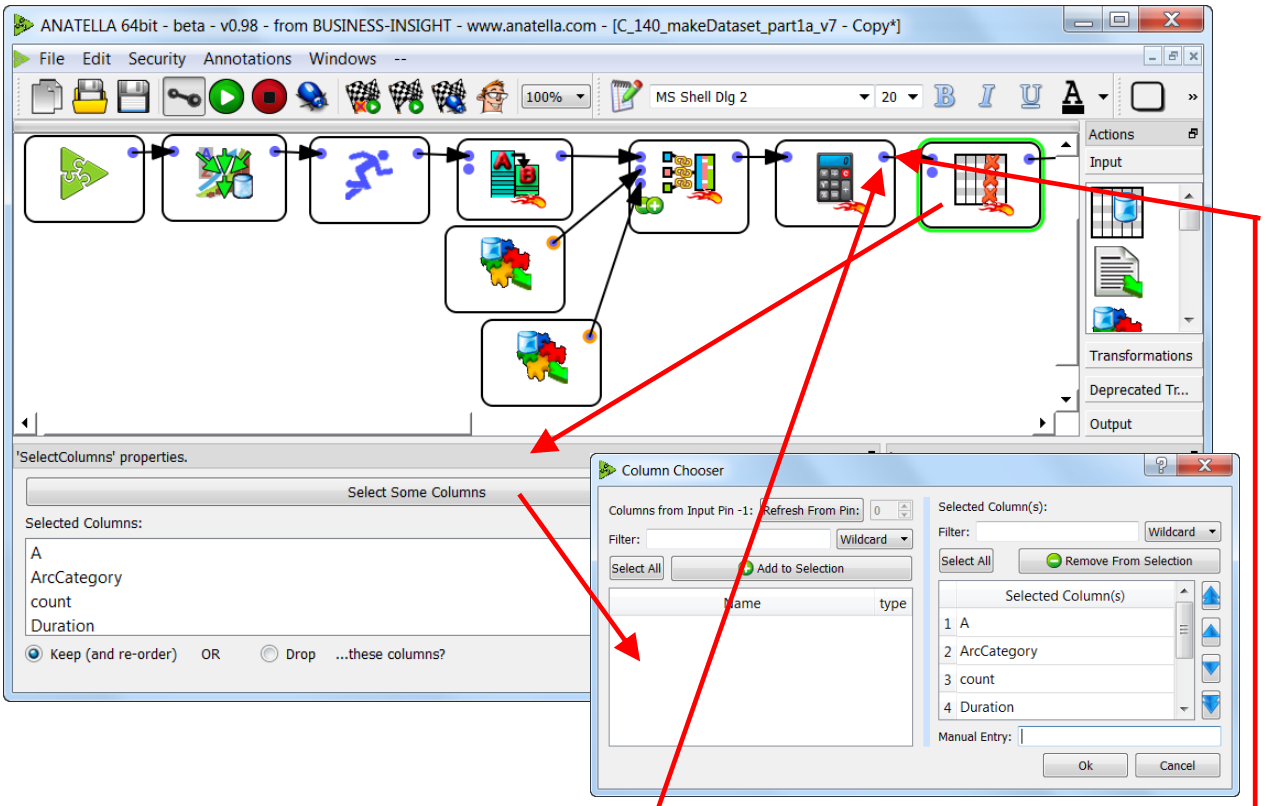
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- The graph execution failed and no table ever "arrived" inside the currently edited Action (and since there is no input table, there are no column names to display neither). In such situation, simply check the Log window for error messages that explain you how to debug your Anatella transformation graph.

- The graph execution is taking too much time. In such situation, you should use sampling.

- The graph execution is taking too much time (even when using sampling). In such situation, you should create a HD cache just before the currently edited Action: Anatella will use this HD cache to display instantaneously the required column names inside the "Column Chooser" window.

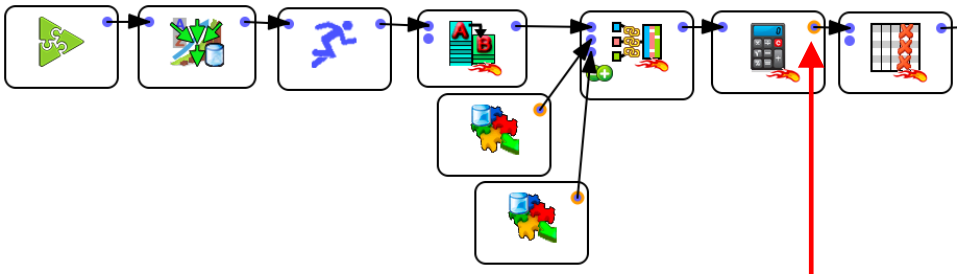
  For example: Let's assume that we are using the "Column chooser" window to select which columns "to keep" inside a SelectColumn Action:



… when we click the "Refresh from Pin" button (from the "Column chooser" window), Anatella executes the graph up to this point:
This execution is quite long (because of the 2 large slaves tables used inside the MultiJoin Action) and Anatella might fail to display the required columns. The solution is simple: Click this pin (in "run" mode) to create a HD cache:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We don't need to "cache" the complete table: The first row is enough: As soon as one row is visible inside the data preview, click the abort button in the toolbar to stop Anatella. We now have:



Please note the HD cache that is now available here (i.e. there is a yellow or green circle around the output pin here). This HD cache is used by the "Select Column" window to instantaneously display the column list when clicking the "Refresh from Pin" button.

Most of the time, instead of focusing on getting the "Refresh from Pin" to return a list of columns, it's simpler (and quicker) to copy/paste the required column-list from another "Column Chooser" window: See section 5.1.4. for more information about this subject.

8.9. Collaborative Work

Anatella fully supports collaborative work.

For example, one of the largest Anatella-based project contains about 500 Anatella-Graph-Files. This project has a dozen analysts that are simultaneously & collaboratively working full-time on it.

Collaborative work can happen at different levels:

1. The Analysts can work on the different Anatella graph files and share their work using a versioning system (or using a simpler synchronization software such as "Google Drive" or "DropBox").

   The most established versioning system is currently "Git". One limitation of Git is that it only supports text files saved in the "utf-8" character encoding (other text file encodings only appear as "binary" files). Please refer to the section 7.1. to know how to save your .anatella files in an optimized format (UTF-8) to get a perfect Git integration.

   A Git project nearly always include these 2 files (or some files with an equivalent usage):
   - default_config.cfg
   - config.cfg

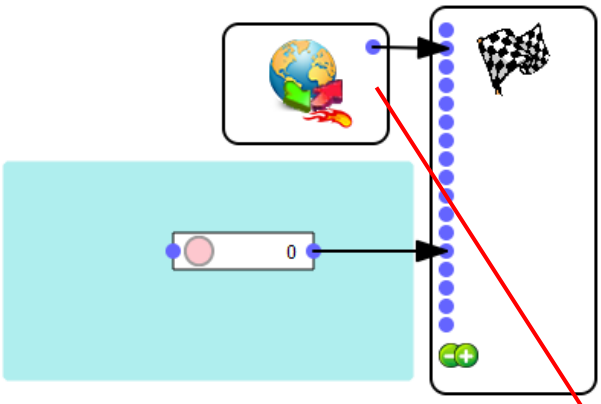
   …with the "config.cfg" file inside the .gitignore file.
   Typically, the program first load all the settings inside the "default_config.cfg" file and thereafter it loads the settings from the "config.cfg" files (possibly over-writing some settings from the "default_config.cfg" file). Thus, every developer can "tune" all the project's settings

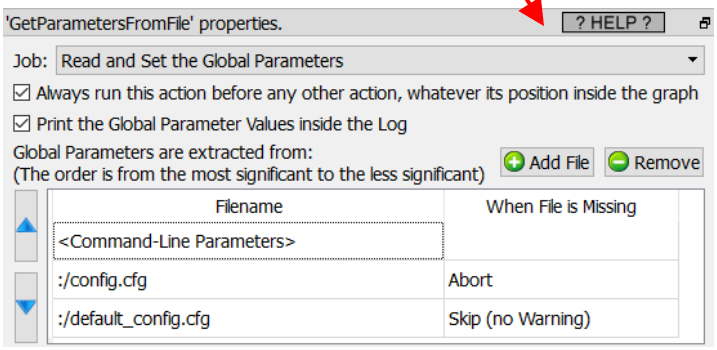
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

using its personal “config.cfg” file, to adapt them to the local machine. To follow this idea inside Anatella, you can include inside your graph:

Initialize the Global Parameters from the two Cfg files

Actually do the real transformation after setting the Global Parameters





2. Use the “AnatellaGraph” Javascript Object to automatically modify “in bulk” hundreds of .anatella files to:
   - Manage the users that are allowed to edit/run the graphs
   - Save the .anatella file in a format optimized for a version control system (such as git). This optimized format has the following properties:
     - UTF-8 encoding
     - No file encryption
     - Various Cleaning of the XML file: No “keyHD” attributes, No “forceHDBuffering” attributes, No “WorkingSpace” XML tag, no breakpoints inside the Javascript code.
     - (optional): No special users for the .anatella file

3. The Analysts can work collaboratively on the code of some tailor-made Javascript/R/Python Actions.

   One tailor-made Javascript/R/Python Action can potentially be used in many project and in hundreds of different .anatella-graph-files and you want a “centralized” way of managing your tailor-made JS/R/python Actions. The Analysts can work collaboratively on the “central” repository containing all your tailor-made JS/R/Python Actions. Once an analyst upgrades (or fix) a tailor-made JS/R/Python Action, the change is automagically propagated to all the .anatella graph files that uses the Action. A good idea is (again) to use a versioning system to keep track of the scripts inside your “central” repository.

   See the section 9.7. and the section 9.8. to know more about this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

4. The deployment and monitoring of the correct execution of your .anatella graphs can also be done in a collaborative way. More precisely: Anatella integrates with Slack: https://slack.com/ using the

Action. This means that you can easily send to Slack some monitoring messages that are therafter dispatched (by Slack) to the proper members of your team (to their email, cell phone, etc.).

   If you don't want to work with Slack, you can also use a simpler "email"-based approach for monitoring: For example, this sends a simple "coucou message" to "bob@timi.eu":

```
var sm=new SendMail("ssl0.ovh.net", "anatella@timi.eu",SendMail_SSL, 465);
sm.setLoginPassword("bob@business-insight.com",password);
sm.send("bob@timi.eu","test","coucou message");
sm.close();
```

5. You might also want to activate the option named "*use a global locking mechanism*". This option is usefull when there are several data analysts working simultaneously (and collaboratively) on the same server. This option prevents two analysts to simultaneously change the same .anatella graph file. You'll find more details about this option inside the section 7.2.4.

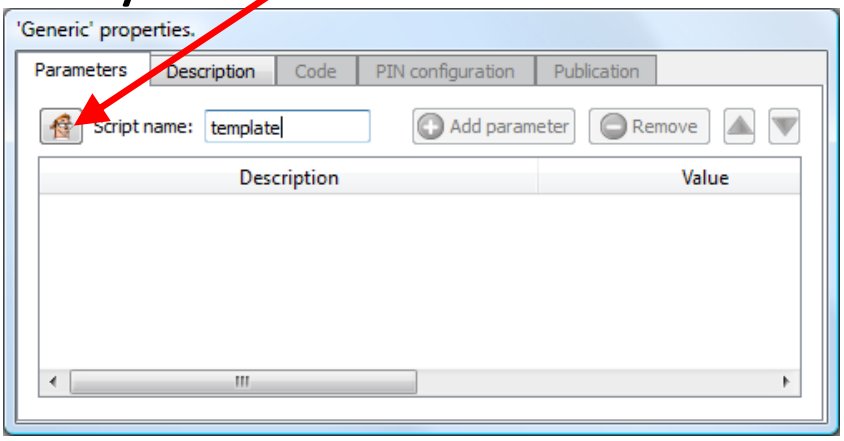
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9. Anatella for the Expert Users

Only the Anatella "expert users" have access to the Anatella "expert features". The "expert features" allow you to develop rapidly brand new Actions. New Anatella Actions can be coded in either Javascript, R, Python or C/C++.

To become an Anatella "expert user" (i.e. to switch between the normal and expert user mode), you can:

1. Press CTRL-U anytime ('U' stands for User mode)
2. Click the button in the main toolbar of the application.
3. Click the very small button in the property window of any Script-based action:



To change the Key file inside the Encrypt Action, you also need to switch to Expert-User-Mode.

9.1. Developing new Script-Based Anatella Actions

9.1.1. Selecting the Right Language to create your new Actions

To create new Actions, you can choose between different (scripting) languages: R, Python, JavaScript or C/C++.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here are the Pro&Con of each of these languages:

Criteria	JavaScript	R	Python	C/C++
Runtime Speed	★★★	★½	★	★ x 30
Scalability (i.e. Ability to process large tables)	★★★★★★	★½	★★	★ x 30
General-Purpose code available from the Internet, directly ready to "Copy-Paste"	★★	★★★★	★★★★★★	★★★
Availability of Machine Learning Algorithm (i.e. will you find the ML algorithm that you are searching for?)	No	★★★★	★★	★★★★
Short Development Time Easy to rapidly get some results	★★★★★½	★★★	★★★★	No
Directly offers some facilities to easily do Matrix/Vector computations	No	Yes	Yes	No
The Syntax of the Language & the Libraries is agreable/coherent.	★★★★	½	★★	★½

What's the difference between JavaScript and Java?

Actually, the 2 languages have almost nothing in common except for the name. Java is coded in a similar fashion to C++. It is powerful enough to write major applications. Java has been generating a lot of excitement because of its unique ability to run the same program on IBM, Mac and Unix computers. Java is not considered an easy-to-use language for non-programmers.

JavaScript is much simpler to use than Java. No compilation, no applets, just a simple sequence.

Some Comments about the above table:

- C/C++ is the fastest, the most scalable and the most universal langage. Actually, the (R/Python/JS) languages are all developed in (C/C++) and thus, most of the functionalities from these languages are also directly available in C/C++. Unfortunately, C/C++ is also the most difficult language to master. The time required to develop a new functionality using C/C++ is usually (at least) 20 times greater than with any of the other languages. So, unless you need extreme speed, I would suggest you to avoid C/C++.
- In terms of Speed and Scalability, JavaScript is clearly the winner (after the C/C++). So, if you need to do some simple data processing on some large datasets, avoid R and Python: Use JavaScript.
  The JavaScript engine included inside Anatella is more scalable than the R/Ptyhon engine because it processes data "row-by-row" in "streaming mode". By default, the R/Python engines cannot process data "row-by-row" (in streaming mode) and are thus limited: They can only handle smaller data tables (where the whole table must fit inside the limited RAM of the computer). Inside Anatella, there is a special mode that allows to manipulate large tables (i.e. larger than the RAM) using the R&Python engines (i.e. when you partition the input table) but this mode is usually not as efficient (in terms of speed) as the normal "streaming mode" that is always used in JavaScript.
  The JavaScript engine included inside Anatella contains an advanced debugger: You can place breakpoints, look at the variables inside the watch window, etc. This means that the debugging

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

of Javascript-based Actions is really easy (compared to R/Python actions that do not have any debugger).

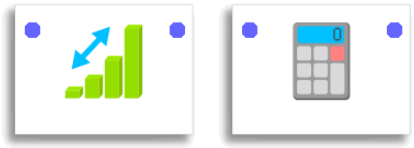
- Sometime, you won't be able to use Javascript to code your new Action because:
  - You need to use a specific library that only exists in R or Python.
    The required library can be of two different types:
    - General Purpose Library: e.g. To access some REST service, to read some specific file format, to post some results inside a non-SQL database, etc. Your best option is then to use Python.
    - Machine Learning Library: Your best option is to use R because R has still the largest Machine Learning Library and the largest Algorithmic Library.
  - You need to do many Matrix/Vector computations: You can use R or Python indeferently. R is usually slightly faster during runtime but Python is easier/faster to write. If you only have vector computations to perform, you might also be interested in using the simple&fast Vectorized Calculator Action (see section 5.7.10).

The Appendix A of this document contains an introduction to the Javascript Language. After reading this short introduction (4 pages), you should be able to code (nearly) any Action using Javascript.

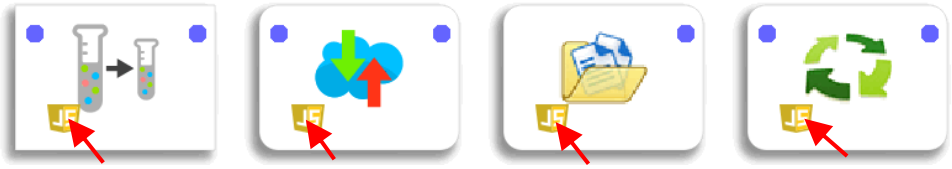
Although Javascript is not very popular (yet!) in the particular field of the "data science", it still is a language that worth learning, just because of its vast popularity in every other fields of the "computer sciences": See appendix C (section 11.3) for more details about the popularity of Javascript.

Here are some more comments about each language:

- **C/C++.**
  You can recognize these Actions because they don't have any Javascript, R or Python logo:



- **Javascript.**
  You can recognize these Actions because they have a Javascript logo here:



  When developing a new Action in Anatella, the Javascript language is usually the best solution because:
  - The syntax of the language is cleaner,
  - It's faster than R or Python
  - It can easily run in "streaming" mode inside Anatella (i.e. with a very low RAM-memory-consumption).
  - There is already a large quantity of Javascript-based Actions inside Anatella that you can use as "starting point" to create your own Actions.

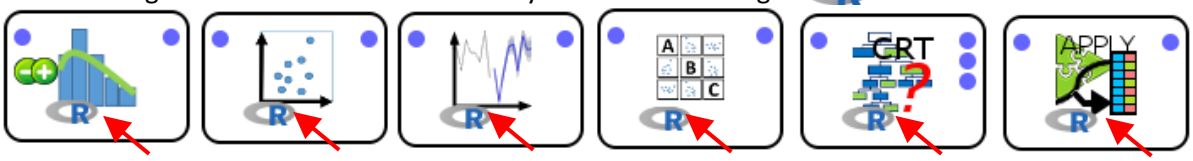
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- You won't be limited by the Anatella Javascript engine because it fully supports the latest version of the Javascript/ECMAScript language (see the sections 11.1, 11.2, 11.5, 11.6 for more details about Javascript/ECMAScript)

To know how to create new Actions in Javascript, see the section 9.2.1.

- **R**

You can recognize these Actions because they have a small "R logo" here:



The integration between R and Anatella gives access to a very large library of algorithms because the R language is the language that offers the largest library of algorithms. These algorithms cover many different usages: machine learning, time series, clustering, etc. If you need a specific algorithm, there are good chances that it already exists inside R.

To know how to create new Actions in R, see the section 9.2.2.

- **Python**

You can recognize these Actions because they have a small "Python logo" here:

Most developpers like Python because of its easy syntax and the ability to usually arrive quickly to a working solution. However, this comes at a cost: i.e. The Python language is very (very!) slow, it consumes a large quantity of RAM memory and it offers a limited set of Machine Learning algorithms compared to the R language.

Many coders, unaware of Anatella, are using Python to develop simple ETL scripts (e.g. using Panda dataframes). This is an error because ETL script that are developped in Python are:

- …several orders of magnitude slower than the same ETL created purely using Anatella Standard Actions.
- …using a very high quantity of RAM memory because, in Python, all data-transformations are 100% "in RAM" (although, if you are using the Python engine included inside an Anatella data-flow, Anatella can manage to "stream" your datasets, so that your python scripts only uses a reduced quantity of RAM).
- …not scalable to large data volumes (e.g. for Big Data applications). Since all data-transformations in Python are "in RAM", you are limited to the RAM of you server/laptop. In opposition, Anatella has no size limitation because it processes data in "streaming" mode.
- …slower to develop&create than a simple Anatella graph. This usually translates to larger bugdets and more delays for the final end-users.
- …difficult to maintain & support compared to the "easy to understand" Anatella graphs. This usually translates to much larger "support&maintenance" costs for the final end-users (that needs specific&costly man power to maintain their operational system).

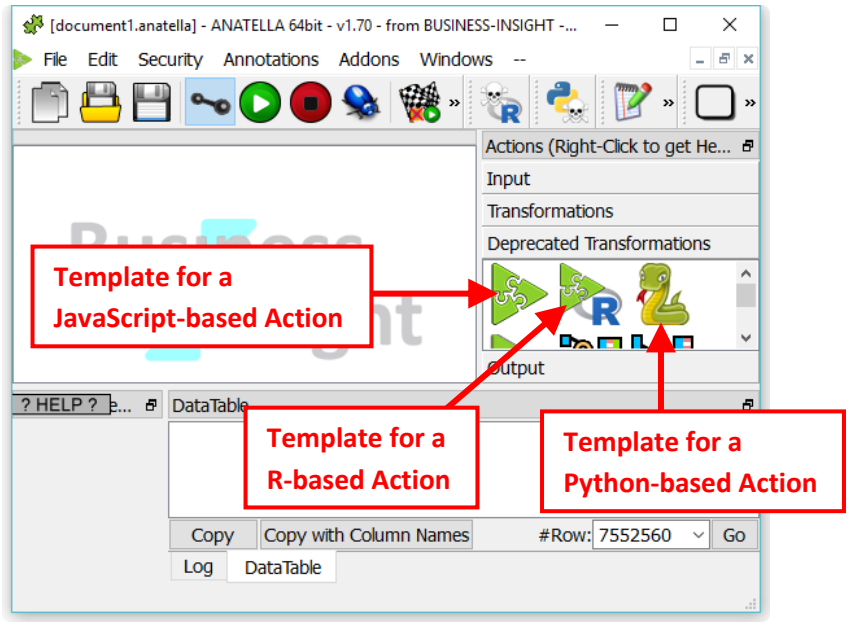
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Because of the limitations of Python listed here above, we advise you to only use Python when you have a very specific and very complex (and also very rare) data transformation to create that could not easily be coded using simple Anatella Actions.

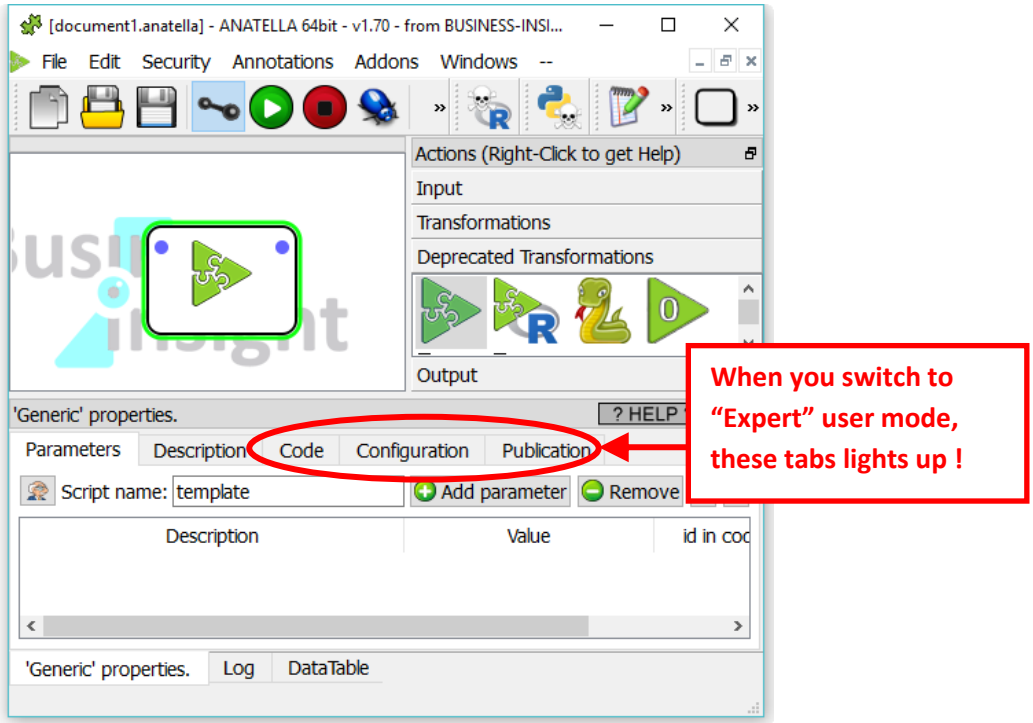
To know how to create new Actions in Python, see the section 9.2.3.

9.1.2. How to edit the Script-code of an Action?

The easiest way to develop new Script-based Anatella Actions is to edit & modify an existing one. Here are the pre-existing templates that are a good "starting points" to create a new Action coded in Javascript, R or Python:

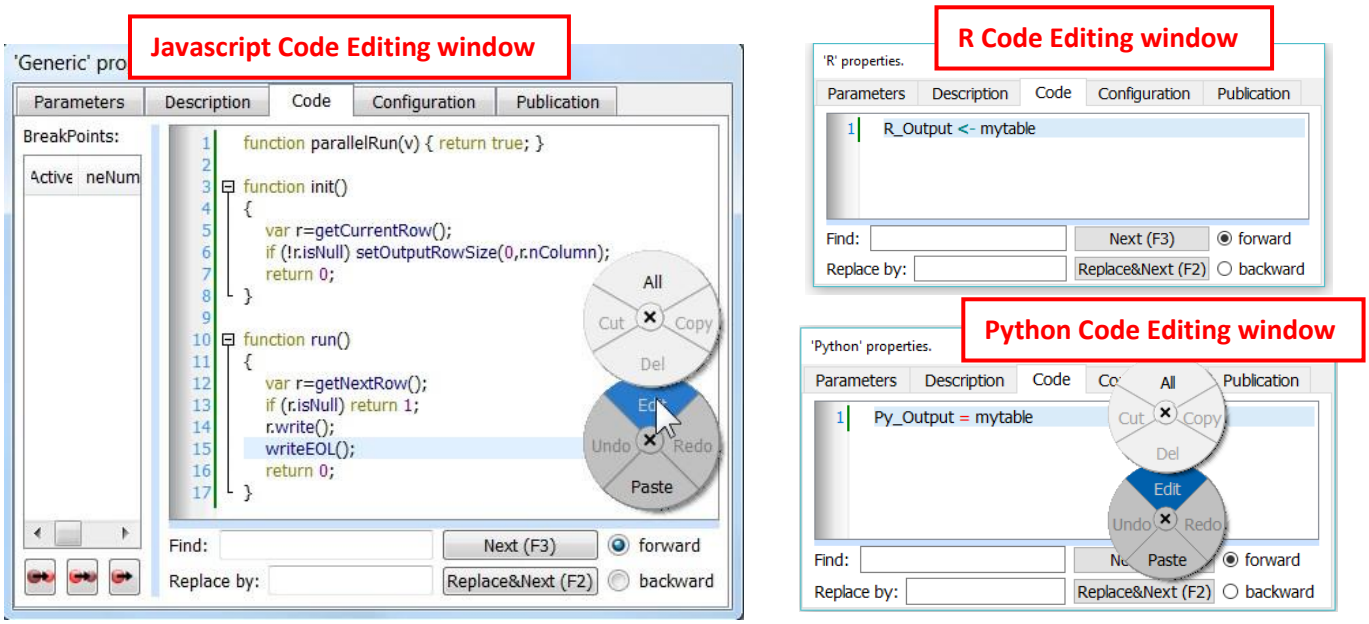


Let's drop into our Anatella Graph one of these "Templates". Double-click the Action (to edit its properties) and Switch to expert user mode (i.e. Click the button in the main toolbar of the application). You should now have something like:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's click on the "code" tab! You should now have:



You can directly edit and tweak the code of the action in the "code" tab and press F5 to execute it! To be comfortable when editing the code, you have access to:

1. A circular context menu (accessible with a right-mouse-click) that offers you the classical text-edition-functionalities: copy, cut, paste,…
2. **CTRL+Wheel:** Increase/decrease the font size.
3. The following standard shortcuts are available while editing your code:

Keypresses	Action
Backspace	Deletes the character to the left of the cursor.
Delete	Deletes the character to the right of the cursor.
Ctrl+C	Copy the selected text to the clipboard.
Ctrl+F	Finds a word into the code.
F3	Press F3 to repeat the last search.
Ctrl+H	Finds & Replaces a word into the code.
F2	Press F2 to repeat the last replacement.
Ctrl+Insert	Copy the selected text to the clipboard.
Ctrl+K	Deletes to the end of the line.
Ctrl+V	Pastes the clipboard text into text edit.
Shift+Insert	Pastes the clipboard text into text edit.
Ctrl+X	Deletes the selected text and copies it to the clipboard.
Shift+Delete	Deletes the selected text and copies it to the clipboard.
Ctrl+Z	Undoes the last operation.
Ctrl+Y	Redoes the last operation.
Left	Moves the cursor one character to the left.
Ctrl+Left	Moves the cursor one word to the left.
Right	Moves the cursor one character to the right.
Ctrl+Right	Moves the cursor one word to the right.
Up	Moves the cursor one line up.
Down	Moves the cursor one line down.
PageUp	Moves the cursor one page up.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

PageDown	Moves the cursor one page down.
Home	Moves the cursor to the beginning of the line.
Ctrl+Home	Moves the cursor to the beginning of the text.
End	Moves the cursor to the end of the line.
Ctrl+End	Moves the cursor to the end of the text.
Alt+Wheel	Scrolls the page horizontally (the Wheel is the mouse wheel).
Ctrl+Wheel	Increase/decrease the font size

9.2. About the integration of Javascript, R, Python inside Anatella

This section gives more details about how each scripting language (Javascript, R, Python) is integrated inside Anatella.

9.2.1. JavaScript Integration inside Anatella

This section explains how to write an Anatella Action using some Javascript code. We'll explain how the integration works by giving several examples.

9.2.1.1. Example 1

This is the simplest JavaScript that you can write:

```
function parallelRun(v) { return true; }

function init()
{
   var r=getCurrentRow();
   if (!r.isNull) setOutputRowSize(0,r.nColumn);
   return 0;
}

function run()
{
   var r=getNextRow();
   if (r.isNull) return 1;
   r.write();
   writeEOL();
   return 0;
}
```

This script simply "passes by" the rows from the input pin to the output pin without doing any treatment. This script is simply a "Basic Template" that you can "tweak" to create your own, complex scripts.

The script returned to the Anatella Engine must have at least two functions: "*init()*" and "*run()*", otherwise the JavaScript won't work. The "*parallelRun()*" function is facultative: if It's missing, Anatella assumes the following: "*function parallelRun(v) { return false; }*". See section 9.9.1. for more information about this subject.

The "init()" function is called only once, to allow you to initialize the variables used inside your script. The "*init()*" function must return 0 if the initialization went fine. If the initialization failed (for whatever reason), the "*init()*" function must return 1. Once the script is initialized properly, Anatella will execute the "*run*" function several times. Anatella will keep executing again and again the "*run*" function while the "*run*" function returns 0. If the "*run*" function returns 1, it means that the transformation script completed successfully. You can also stop the execution of your script with an error message, using the "*throw("message")*" function that throws an error to Anatella.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Basically, a JavaScript Action reads rows from its input pins (on the left of the "box") and writes rows to its output pins (on the right side of the "box"). The two global functions to "read rows" from the input pins are:

1. getCurrentRow(inputPinIndex)
2. getNextRow(inputPinIndex)

These two functions are returning "row" objects. You can thereafter write these "rows" to any output pin, using the method "*write(outputPinIndex)*" of the row object.

The syntax-highlight-system of Anatella allows you to directly check the correct syntax of your scripts, before even compiling them. A keyword that has the proper syntax will turn **dark blue**. If a keyword stays **black**, it's a syntax error.

Before writing anything on an output pin X (before any call to the "*write()*" method of the "row" object), you must define the size of the output pin X (i.e. the number of columns inside the table associated with the output pin X). To define the size of a specific pin, use the global function "*setOuputRowSize(outputPinIndex ,number_of_column)*".

Let's go back to the above example. We have, inside the "initialization" function:

```
1  function init()
2  {
3    var r=getCurrentRow();
4    if (!r.iNull) setOutputRowSize(0,r.nColumn);
5    return 0;
6  }
```

On line 3: We get the first row available on input pin 0. (if no parameters are given to the "*getCurrentRow()*" function, then it's is assumed that we want a row from pin 0). This first row is stored into an object named "r".

One line 4: If the table on input pin 0, does not contain any row, then the "*getCurrentRow()*" function returned a row of the "null"-type, to signal that no additional rows are available. Thus, we test if we get a valid row (i.e. a non-NULL row) and use the global function "*setOuputRowSize*" to signal to Anatella that the number of columns on the output pin 0 is "r.nColumn". "nColumn" is a property of the "row" object that give you the number of columns that the "row" contains.

To summarize, in the "initialization" function, we simple "say" that the size of the **output** pin 0 (i.e. the number of columns of the table associated to the output pin 0) is equal to the size of the **input** pin 0.

Let's now examine in more detail the "run" function:

```
8  function run()
9  {
10   var r=getNextRow();
11   if (r.isNull) return 1;
12   r.write();
13   writeEOL();
14   return 0;
15 }
```

On line 10: We get the next row available on input pin 0. (if no parameters are given to the "*getNextRow()*" function, then it's is assumed that we want a row from input pin 0). All the rows of an input table are accessible sequentially, one row-at-a-time, using the "*getNextRow()*" function. This new row is stored into an object named "r".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

On line 11: If the table on input pin 0, does not contain any more row (i.e. we have already read all the rows and we have now arrived at the "bottom" of the input table), then the "*getNextRow()*" function returned a row of the "null"-type, to signal that no additional rows are available. If no more rows on input pin 0 are available, than we return 1, to indicate to the Anatella Engine that this script completed successfully.

On line 12: We use the method "*write()*" of the object "r" to write our row "r" on the output pin 0 (If no parameters are given to the "*write()*" method, then it's is assumed that we want to write to the output pin 0).

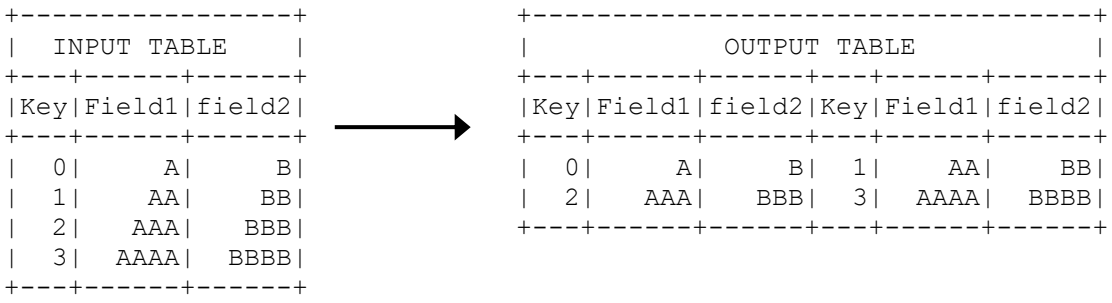
On line 13: We use the global function "*writeEOL(outputPinIndex)*" to signal to Anatella that the computation of the row for the output pin 0 is complete: Anatella can take the "completed" row on the output 0 and send it to the next Action. At each execution of the "*run()*" function, you can call maximum **one** time the function "*writeEOL()*" for each output pin. In other words, each execution of the "*run()*" function produces zero or one row on each output pin. The name of the "*writeEOL()*" function is the contraction of ""Write-End-Of-Line". In opposition, at each execution of the "*run()*" function, you can call as many times as you want the "*getNextRow()*" function (to **read** from the input pins as many rows as you want).

When Anatella executes the "*writeEOL(0)*" function, Anatella checks if the row on the output pin 0 has been properly initialized. For example, if you did setup an output pin of size 10 and you did only "write" to this pin 9 columns, then Anatella will stop the execution of the script with an error message. This "sanity check" allows easily detection of coding errors and it's also important to ensure consistency of the output table.

On line 14: We return 0 to indicate to Anatella that the transformation must continue because there are (most certainly) more rows to process.

9.2.1.2. Example 2

Using the knowledge acquired in the previous section, we can already design a new Anatella-script that performs the following transformation:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The Anatella-script of this new transformation is:

```
1  function init()
2  {
3    var r=getCurrentRow();
4    if (!r.iNull) setOutputRowSize(0,2*r.nColumn);
5    return 0;
6  }
7
8  function run()
9  {
10   var r1=getNextRow();
11   if (r1.isNull) return 1;
12   r1=rowDeepCopy(r1);
13
14   var r2=getNextRow();
15   if (r2.isNull) return 1;
16
17   r1.write();
18   r2.write();
19   writeEOL();
20   return 0;
21 }
```

The principle of the above script is simple:

1. On line 10 and 14, we read two consecutive rows of the input table (these rows are stored in 'r1' and 'r2')

2. On line 17 and 18, we write on the **same output row** on pin 0 the two rows ('r1' and 'r2') that we just read. These 2 input rows will thus both end-up on the same output row. On line 19, we write the "End-Of-Line" marker on the output pin 0, to mark the end of the output row.

We won't explain inside this section the meaning of the line 12: please refer to the later section 9.2.1.5. to have more information about this subject.

9.2.1.3. Example 3

The script of the previous section is not very nice because the output table contains several columns with the same name. Indeed the column names of the output table are "Key", "Field1", "Field2", "Key", "Field1", "Field2". We will thus improve our Anatella-script to obtain (please note the new prefixes on the column names: "T1." And "T2."):

```
+-----------------+
|  INPUT TABLE    |
+---+------+------+
|Key|Field1|field2|
+---+------+------+
|  0|     A|     B|
|  1|    AA|    BB|
|  2|   AAA|   BBB|
|  3|  AAAA|  BBBB|
+---+------+------+
```

```
+-------------------------------------------------------+
|                      OUTPUT TABLE                     |
+------+---------+---------+------+---------+---------+
|T1.Key|T1.Field1|T1.field2|T2.Key|T2.Field1|T2.field2|
+------+---------+---------+------+---------+---------+
|     0|        A|        B|     1|       AA|       BB|
|     2|      AAA|      BBB|     3|     AAAA|     BBBB|
+------+---------+---------+------+---------+---------+
```

The Anatella-script of this new transformation is (we have added into the code the lines marked with vertical red lines. These are the lines 1, 7, 11-19, 30-35):

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
1   var isFirstRow;
2
3   function init()
4   {
5     var r=getCurrentRow();
6     if (!r.iNull) setOutputRowSize(0,2*r.nColumn);
7     isFirstRow=true;
8     return 0;
9   }
10
11  function renameColumns(myRow,prefix)
12  {
13    for(i=0;i<myRow.nColumn;i++)
14    {
15      rowSetColumnName(myRow,
16                       i,
17                       prefix+rowGetColumnName(myRow,i));
18    }
19  }
20
21  function run()
22  {
23    var r1=getNextRow();
24    if (r1.isNull) return 1;
25    r1=rowDeepCopy(r1);
26
27    var r2=getNextRow();
28    if (r2.isNull) return 1;
29
30    if (isFirstRow)
31    {
32      isFirstRow=false;
33      renameColumns(r1,"T1.");
34      renameColumns(r2,"T2.");
35    }
36
37    r1.write();
38    r2.write();
39    writeEOL();
40    return 0;
41  }
```

The objective of the above code changes is to change the column names of the output table: Instead of the column names: “Key”, “Field1”, “Field2”, “Key”, “Field1”, “Field2”, we want to obtain the column names: “T1.Key”, “T1.Field1”, “T1.Field2”, “T2.Key”, “T2.Field1”, “T2.Field2”. The column names of a table are part of the “meta-data” of the table. Anatella has very strict rules concerning the “meta-data” of the output tables:

1. All the meta-data of an output pin P are defined when you are writing the first row on the pin P.
2. All changes to the meta-data of an output pin P after the first output row has been written to the pin P are ignored.

As you can see, these rules are making a strong distinction between the first output row of an Action and the subsequent ones: the first output row is the only row that is used to define the “meta-data” of the output table. Thus, when working with the meta-data of an output table, we must take extra-care when writing the first row on the output pin.

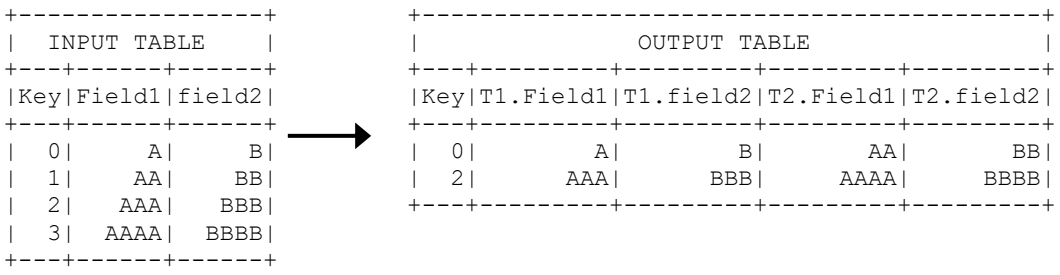
On line 1, we create a new boolean variable named ‘isFirstRow’. JavaScript has an implicit variable declaration: the first time we assign a value to a variable, the variable is created. The variables that are explicitly declared will always appear in the watch-window of the Anatella-debugger. Implicitly declared variable might not always appear in the watch-window of the Anatella-debugger (thus, if you want the debugger to work properly, you’d better always declare all your variables).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

On lines 30 to 35, we check (using the variable 'isFirstRow') if we are "at the first row" of the output table. If that's the case, we change the column names of the output table by calling the user-defined function "*renameColumns()*". The code of the user-defined function "*renameColumns()*" is on lines 11 to 19. It uses the two Anatella-specific-global-functions:
"*rowSetColumnName(row,columnIndex,string)*" and
"*rowGetColumnName(row,columnIndex)*".

9.2.1.4. Example 4

Let's now change one last time our Anatella-Script to obtain (Please note that we removed one of the two "Key" columns):



INPUT TABLE		
Key	Field1	field2
0	A	B
1	AA	BB
2	AAA	BBB
3	AAAA	BBBB

OUTPUT TABLE				
Key	T1.Field1	T1.field2	T2.Field1	T2.field2
0	A	B	AA	BB
2	AAA	BBB	AAAA	BBBB

The Anatella-script of this new transformation is (we have added into the code the lines marked with vertical red lines: These are the lines 7,34,37):

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
1   var isFirstRow;
2
3   function init()
4   {
5     var r=getCurrentRow();
6     if (!r.iNull) setOutputRowSize(0,2*r.nColumn);
7     dropColumn(0,r.nColumn);
8     isFirstRow=true;
9     return 0;
10  }
11
12  function renameColumns(myRow,prefix)
13  {
14    for(i=0;i<myRow.nColumn;i++)
15    {
16      rowSetColumnName(myRow,
17                       i,
18                       prefix+rowGetColumnName(myRow,i));
19    }
20  }
21
22  function run()
23  {
24    var r1=getNextRow();
25    if (r1.isNull) return 1;
26    r1=rowDeepCopy(r1);
27
28    var r2=getNextRow();
29    if (r2.isNull) return 1;
30
31    if (isFirstRow)
32    {
33      isFirstRow=false;
34      var s=rowGetColumnName(r1,0);
35      renameColumns(r1,"T1.");
36      renameColumns(r2,"T2.");
37      rowSetColumnName(r1,0,s);
38    }
39
40    r1.write();
41    r2.write();
42    writeEOL();
43    return 0;
44  }
```

The objectives of the code modifications are:

1. to avoid the duplication of the 'Key' column in the output table: see line 7.
2. to avoid renaming (to "T1.Key") the 'Key' column: see lines 34 and 37.

Inside this last script, we used the Anatella-specific-global-function "*dropColumn(outputPinIndex,columnIndex)*" that removes one column on one of the output pins. Anatella offers you several global functions that allow you to select exactly which column will be included in the output table. These global functions are: "*dropAllColumns(outputPinIndex)*", "*keepAllColumns(outputPinIndex)*", "*dropColumn(outputPinIndex,columnIndex)*", "*keepColumn(outputPinIndex,columnIndex)*".

You can use also the global function "*keepColumn(outputPinIndex,columnIndex)*" to re-order the columns of the output table. For example, the script:

```
dropAllColumns(0);
keepColumn(0,2);
keepColumn(0,1);
keepColumn(0,0);
```

… creates a 3-columns-output-table and the column-order is reversed (2,1,0 instead of 0,1,2) compared to the original order in which you have written the data.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The source codes of the default Anatella-Script Actions provided with the Anatella software are good examples and good sources of inspiration. I invite you to read them.

9.2.1.5. On the proper usage of the "rowDeepCopy()" function

The proper usage of this function requires a little bit more explanations. There are basically two "use cases" when you need to use the "*rowDeepCopy()*" function:

9.2.1.5.1. Use Case 1

Let's go back to our example of section 9.2.1.2:

```
1  function init()
2  {
3    var r=getCurrentRow();
4    if (!r.iNull) setOutputRowSize(0,2*r.nColumn);
5    return 0;
6  }
7
8  function run()
9  {
10   var r1=getNextRow();
11   if (r1.isNull) return 1;
12   r1=rowDeepCopy(r1);
13
14   var r2=getNextRow();
15   if (r2.isNull) return 1;
16
17   r1.write();
18   r2.write();
19   writeEOL();
20   return 0;
21 }
```

On line 12 of the above script, we used the global function "*rowDeepCopy()*".

Let's first examine what's happening if the line 12 is missing:

> Inside the above script, on line 10, we extract a new row from the table on input pin 0. This row is placed into the JavaScript "r1" variable. Inside Anatella, all row manipulations are highly optimized and what's really placed inside the "r1" variable are actually only references to a memory buffer that contains all the real row data. When you call, on line 14, the "*getNextRow()*" function, you will re-use the **same** memory buffer to store the content of the new row designed by "r2". For performance reasons, the variables "r1" and "r2" are sharing the same memory buffer to store their content. The variable "r1" now contains some references to a memory buffer that has been altered. You cannot use anymore the variable "r1". Any attempt to use variable "r1" will produce unforeseen results. The solution, of course, is in the usage of the "*rowDeepCopy()*" function.

The line 12 is present: Everything is ok:

> The instruction "`r1=rowDeepCopy(r1)`" will create a completely new fresh copy of the row "r1" into a newly allocated memory buffer. In software terms, Anatella performs a "*deep copy*" of the "r1" row object (and places the copy into "r1" again). The variable "r1" is now sharing its memory buffer with NO other variable. In this situation, the call to the "*getNextRow()*" function, on line 16 is totally harmless.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

To summarize: There exists one memory buffer per input pin. This memory buffer is shared by all the "Row objects" obtained using the "*getNextRow()*" or "*getCurrentRow()*" functions. When you call the "*getNextRow(pinP)*" function, all the "Row objects" obtained using previous calls to the "*getNextRow(pinP)*" or "*getCurrentRow(pinP)*" functions are invalidated: you cannot use them anymore. If you want to still be able to use an old Row object "R" after a call to the "*getNextRow(pinP)*" function, you must perform a deep copy of your Row object "R" (using the "*rowDeepCopy()*" function).

We just wrote in the previous paragraph "*There exists one memory buffer per input pin*". This is not a precise description of the real Anatella implementation but this description is accurate enough to be able to give the required explanation regarding the correct usage of the "*rowDeepCopy()*" function.

Actually, The Anatella C code has been optimized to avoid:
1. the creation of many un-necessary memory buffers.
2. large un-necessary memory copy.

We made an extensive usage of the C pointers arithmetics to achieve the highest performances.

Coding for speed.
You should try to avoid calls to the "*rowDeepCopy()*" function because these calls are CPU intensive: At each call, Anatella will duplicate all the row content and, for very long rows, this means that Anatella might be forced to copy several megabytes of RAM. This might slow down significantly your transformation script.

9.2.1.5.2. Use Case 2

As you already know, inside JavaScript, "Rows" are objects and objects are always access by reference. This means that the following script won't do what you expect it to do:

```
1  var r1=getNextRow();
2  var r2=r1;
3  r1.setColumn(0,"foo");
4  r2.setColumn(0,"bar");
5  r1.write();
6  r2.write();
7  writeEOL();
```

Since the "r1" and "r2" variables are manipulating the same Row Object, the instruction on line 3 will never have any effect at all because its effect is overridden by the instruction on line 4. To obtain a completely separated copy of a "Row object", use the "*rowDeepCopy(rowObject)*" global function. Here is the same example with the appropriate correction:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
1  var r1=getNextRow();
2  var r2=rowDeepCopy(r1);
3  r1.setColumn(0,"foo");
4  r2.setColumn(0,"bar");
5  r1.write();
6  r2.write();
7  writeEOL();
```

Here is an even better correction:

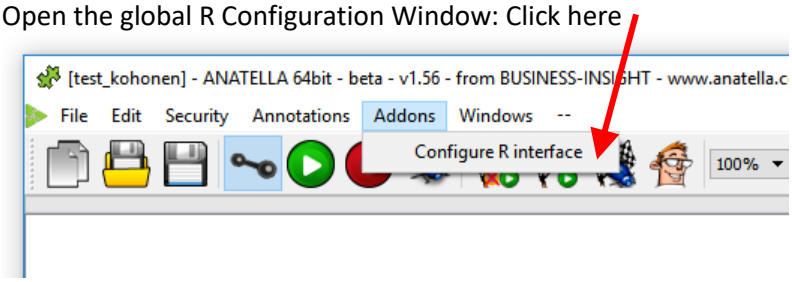
```
1  var r=getNextRow();
2  r.setColumn(0,"foo");
3  r.write();
4  r.setColumn(0,"bar");
5  r.write();
6  writeEOL();
```

9.2.2. R Integration inside Anatella

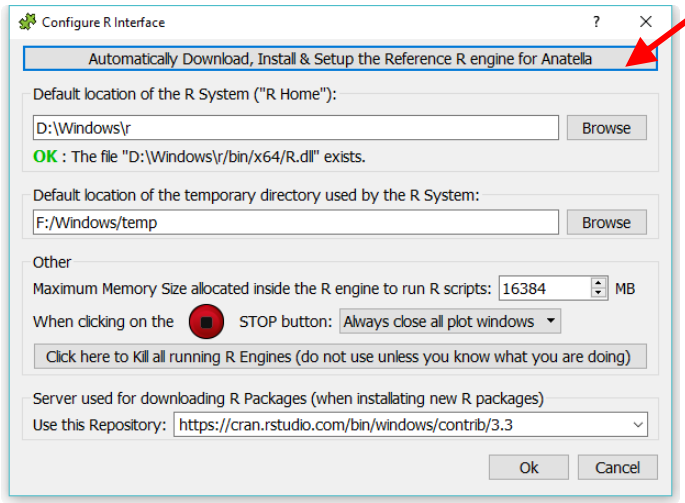
9.2.2.1. First Time Setup

Before using the R engine within Anatella, you must first configure Anatella so that it knows where to find the R engine: This is done in 4 steps:

1. Open the global R Configuration Window: Click here



2. Download & Install a R engine: The easiest way to do so is to click here (This R engine already contains all the optional libraries required to run all the standard R-based Actions in Anatella – You don't need any administrative right to Download & Install this R engine):

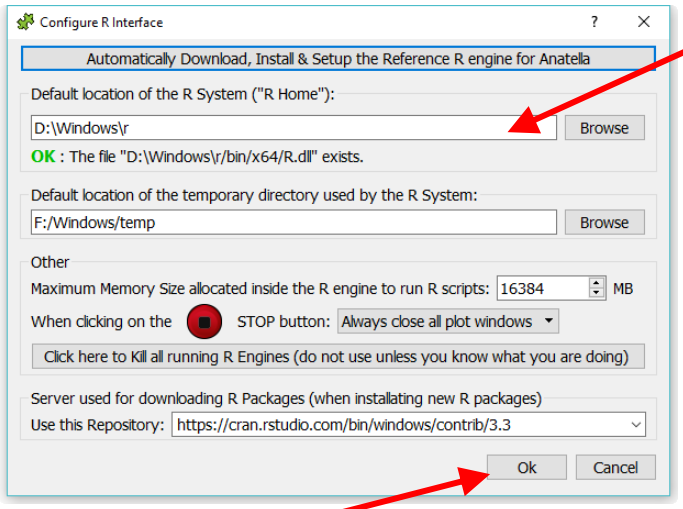


Alternatively, you can also download the latest R engine from here (Administrative rights are required – Anatella will download & install some more optional R libraries when they are first used):

https://cran.r-project.org/bin/windows/base/

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. Select the Location of the R engine (i.e. the directory where you just installed R):

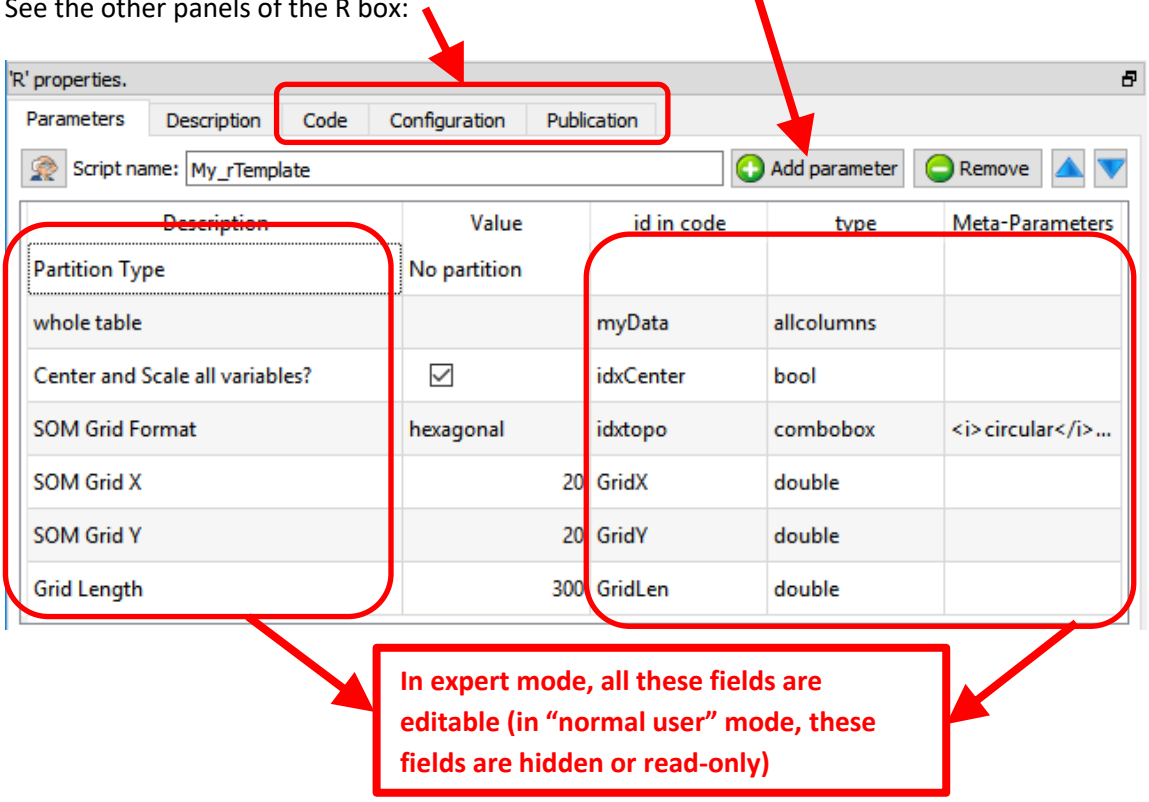


4. Click the OK button:

9.2.2.2. Coding in R inside Anatella

After clicking the icon, you switched to "Expert" mode. While in "Expert" mode, you can:

- Add new parameters (and edit the parameters) of the Action:
- See the other panels of the R box:

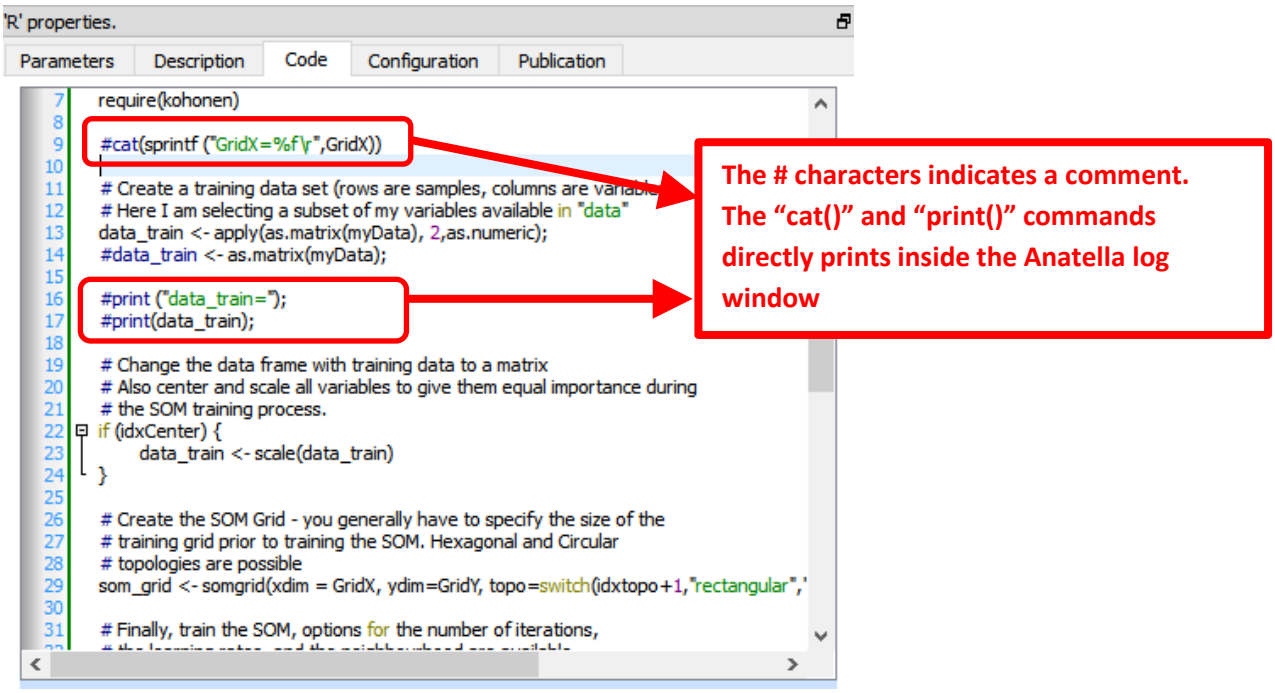


In the above example, we can see that Anatella will initialize the R engine with 6 variables (these 6 variables are named "myData", "idxCenter", "idxtop", "GridX", "GridY, "GridLen") before running the

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

R code. All the parameters that are tables (coming for the input pins) are injected inside the R engine as “data frames”.

In particular, the “Code” panel is interesting: it contains the R code: Here is an example of R code:

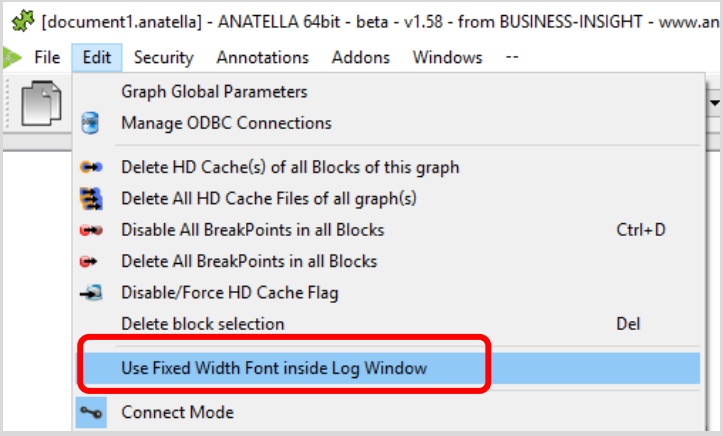


You can use the "print()" or the “cat()” command to display some values inside the Anatella Log window (this is useful for debugging your code).

When the R engine prints some results inside the Anatella Log Window, it assumes that the font used to display the results is of Constant-Width (e.g. so that, we you print an array, the different columns from your array are correctly aligned on each row).

By default, Anatella uses a Variable-Width font inside the Log Window (and thus the array’s are not displayed very nicely).

You can change the font used inside the Log Window here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

(You can also use "CTRL+Wheel" to zoom in/out the text inside the Log Window)

The # characters at the start of a line indicates a comment. Inside R, there are no easy way to comment several lines of codes (i.e. there are no /*…*/ as in other languages such as Javascript, C, C#, Java, etc.).

Here is a special extension that is only available inside Anatella: Use the string "##astop" (without the double-quotes at the beginning and the end) at the very start of a line to prevent Anatella to execute any R code located beyond the "##astop" flag. This is handy when developing new R Actions.

Here are some "good practice" rules to follow when creating a new R code:

1. It might be easier to use an interactive tool such as "R-Studio" to develop the first version of your R code (i.e. during the first "iterations" of code development). Once your R code is working more-or-less properly, you can fine-tune its integration inside an Anatella box using the Anatella GUI. Once the integration inside Anatella is complete, you'll have a block of R code (i.e. an Anatella box) that you can re-use easily everywhere with just a simple drag&drop! (…and without even looking at the R code anymore!)

When developing a new code in R, it happens very often that the R engine computes for a very long time (for example, because you didn't define properly a parameter) and the whole data-transformation is "freezing" abnormally for very long period of time. In such situation, don't hesitate to click the button inside the toolbar to abort all the computations prematurely. The Anatella GUI remains stable even if you cancel all the time the R engine. This allows to make many iterations, to quickly arrive to a working code.

2. If you need a specific data-type to run your R computations, ensure that you convert to this specific data-type before doing any computation. For example, don't assume that you'll always receive a matrix full of numbers in input: More precisely: Always force the conversion to the "number type", if you specifically need "numbers" to do your computations. To convert a data frame (received as input) that is named "myDataFrame" into an Array of numbers (and get rid of the strings!) that is named "myArray":

```
myArray <- apply (as.matrix (myDataFrame), 2, as.numeric);
```

3. If you need some specific packages in order to run your R code, add a few lines of code (at the top of your R script) that installs the required packages if they are not there yet. For example, this install the "kohonen" package if it's not there yet:

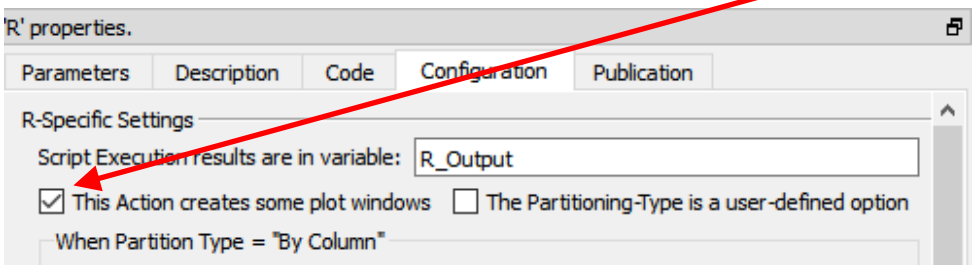
```
if("kohonen" %in% rownames(installed.packages()) == FALSE)
{
    print("installing cohonen package for first time use");
    install.packages("kohonen",repos=RemoteRepository)
}
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

(don't forget to use the option "`repos=RemoteRepository`" inside the above "`install.packages`" command)

When creating a R box that displays some plot window:

1. in the "Configuration" panel, check the option "*This Action creates some plot windows*" (otherwise the "plots windows" are destroyed as soon as the box stops running):



2. Use "x11();" to open new "plot windows" (otherwise all plots ends up inside the same plot-window and you only see the last plot because it has "overwritten" all the previous ones)

3. **Optional:** To avoid consuming much RAM memory for nothing (while R is just busy showing your plots), add at the end of your R code a few lines to destroy all large matrices stored in RAM. For example:

```
#free up memory:
myDataFrame=0; # replace the large data-frame named "myDataFrame" with
               # a single number (0) to reclaim RAM
gc();          # run garbage collector to force R to release RAM
```

To pass back some table-results as output of the R Action, use the "R_Output" variable. The data-type of the variable used as output is very precise: it must be a data frame (and not an array). To convert your variables to data frames, use the following command:

```
myDataFrame <- data.frame( MyVariable, stringsAsFactors=FALSE)
```

(don't forget the option "`stringsAsFactors=FALSE`", otherwise R does strange things!!).

Here are some example of usage of the output variable named "R_Output":

1. To pass on output pin 0 the data frame named "myDataFrame", simply write:

```
R_Output <- myDataFrame
```

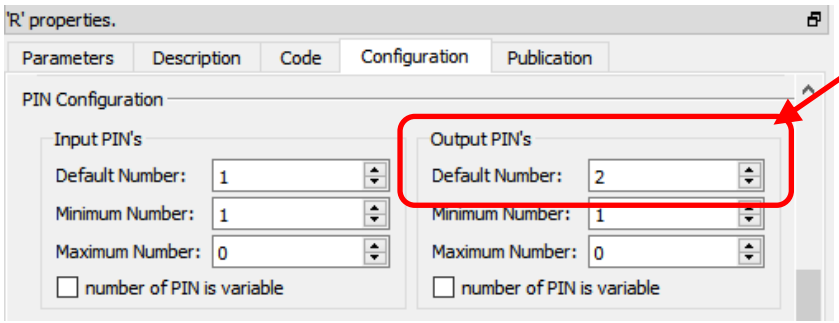
   Please ensure that the type of the variable named "myDataFrame" is indeed a "data frame" (and not an "Array"), otherwise it won't work.

2. To pass on output pin 0 the data frame named "myDataFrame1" and to pass on output pin 1 the data frame named "myDataFrame2", simply write:

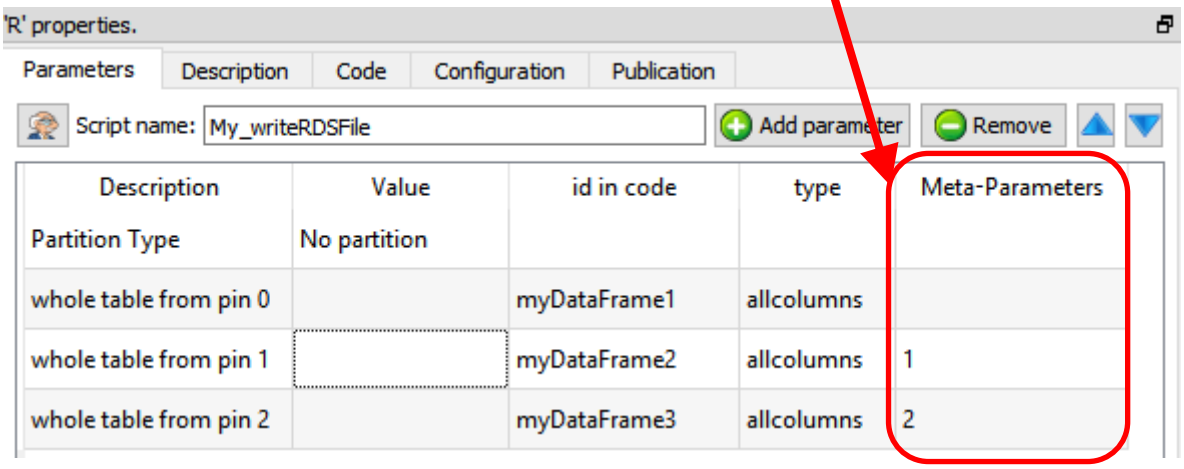
```
R_Output <- list(myDataFrame1, myDataFrame2)
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

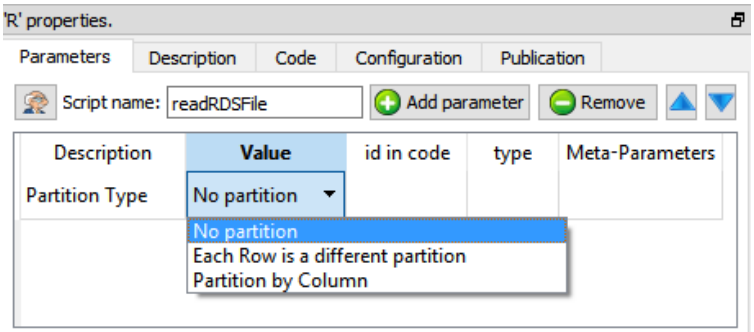
For the above code to work, you must also setup Anatella to have 2 output pins on your box:



To get inside the R environment the input tables available on the second, third, fourth,... pins, use the column "Meta–Parameters" inside the "Parameters" panel: For example, this define 3 data frames (named "myDataFrame1", "myDataFrame2" and "myDataFrame3") that contains the tables on the input pins 0,1 & 2:



The R engine is quite limited in terms of the size of the data it can analyze because all the data must fit into the (small) RAM memory of the computer. To alleviate this limitation of R, you can ask to Anatella to partition your data. There are currently 3 different partitioning options inside Anatella:



How does it work? The table on the first input pin (i.e. on pin 0) is "splitted" in many different "little" tables (the tables on the other pins – pin 1, pin 2, pin 3, etc. – are always injected completely inside the R engine without any "splitting"). The R engine can process easily each of this "little" table because they only consume a small quantity of RAM memory. After the split, Anatella calls the R engine "in-a-loop", several times: At each iteration, the R engine process one different "little" table (and it might also produce some output).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

There are three "Partition Types":

- **No partition** (the default option): self-explaining.
- **Each Partition has the same number of rows (with the exception of the last partition)**: self-explaining.
- **Partition by Column**: When using this option you must select a "Partitioning Column". For example, if you select as "Partitioning Column" the column "Age", then each partition will containing all the people (i.e. all the rows) with the same "Age".

  The concept of "Partition" is used many times inside Anatella: e.g. See the sections 5.5.3. (Partitioned Sort), 5.12.4 (Time Travel), 5.7.9. (Quantile), 5.5.9. (Flatten), 5.3.2.5. (Multithread Run), 5.12.2. (Smoothen Rows) of the "AnatellaQuickGuide.pdf" where the same partitioning concept is also used.

More precisely, when using partitioning, Anatella does the following:

1. Split the table on pin 0 into many different "little" tables (one table for each different partition).
2. Inject into R the variables that contains that data from all the rows of the tables available on pin 1, pin 2, pin 3, etc.
3. Inject into R the variable named "partitionType" (whose value is 0, 1 or 2, depending on which "Partition Type" you are using).
4. Inject into R the variable named "iterationCount" with the value 0.
5. Inject into R the variable named "finished" with the value "false".
6. Run the loop (i.e. process each partition):
   - Inject into R one of the "little" tables (that are coming from the "big" table available on the first input pin – input pin 0).
   - Run your R code inside the R engine.
   - Get back some output results to forward onto the output pin.
   - Increase by one the value of the R variable "iterationCount".
   - Execute the next iteration of the loop (i.e. re-start step 6) until there are no more "little" tables to process.
7. If the Anatella option "*Execute one "Final" iteration with a NILL input when executing with a partition*" is checked (i.e. it's TRUE):
   - Set the variable named "finished" to the value "true".
   - Reset the variables that contains the "little" tables to NILL
   - Run the R engine one last time.
   - Get back some output results to forward onto the output pin.

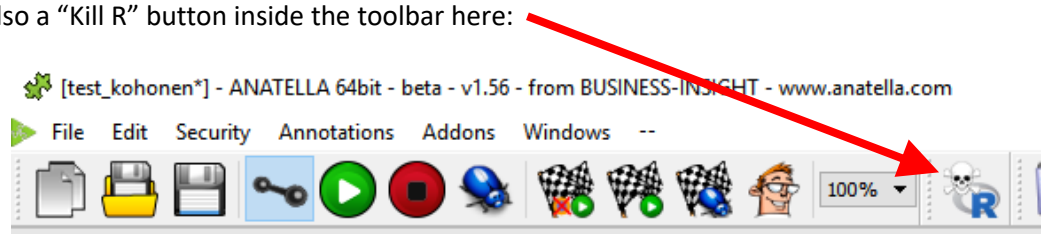
One example where partitioning "makes sense" is when you want to use a predictive model to score a new dataset. In opposition to "learning datasets", "scoring datasets" can be really big (so big that they don't fit into RAM). When you use a partition, **you can compute the predictions for any (scoring) dataset, whatever the size of the dataset.** Nice! ☺

Another example where partitioning "makes sense" is the following: Let's assume that you are working for a large retailer (such as Carrefour, Wallmart, etc) and you need to manage the stocks of all your different products (the products are also named "SKU"=Stock Keeping Unit) at all your different Point-Of-Sales (POS). One important part of stock management involves predicting what will be the demand of each (SKU;POS) pair in the forecoming weeks. If you predict a high demand for a specific (SKU;POS) pair, then you'd better have

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

a significant stock for this same (SKU;POS) pair (unless you want to lose sales because of "out-of-stock" conditions). A typical large retailer has around 20,000 SKU's at each of their POS. Let's assume that we have 200 POS. We thus have 20,000x200=4,000,000 (SKU;POS) pairs. This means that we'll have to compute 4,000,000 predictions (one for each (SKU;POS) pair). This also means that we'll typically have a matrix with 4,000,000 rows: Each row contains information about the past demand for a (SKU;POS) pair. Using the values available on the current row (about past demands), we'll typically use some "time series" algorithm to predict the future demand (for the forecoming weeks). Actually, to compute the prediction for a specific (SKU;POS) pair, **you only need one row of the table**. In other words, all computation can be done row-by-row. This means that we can use the "Partition Type" named "*Each Row is a different partition*". Because of the Anatella-Partitioning-Algorithm, we can handle any number of SKU or POS without being limited by the R engine when it comes to handle large matrices.

There is also a "Kill R" button inside the toolbar here:



This button kills all the R engines currently running. This means that, when you click this button:

- All (R-based) plot-windows are closed
- All R computations are stopped (i.e. don't click this button when your graph is running!).

TIP: Use the button inside the toolbar to close in one click all the plot-windows.

Once you are satisfied with your R-based Action, you can "publish it" so that it always becomes available inside the "common" re-usable Actions: See section 9.7. for more details on thi subject.

9.2.3. Python Integration inside Anatella

9.2.3.1. First Time Setup

Before using the Python engine within Anatella, you must first:

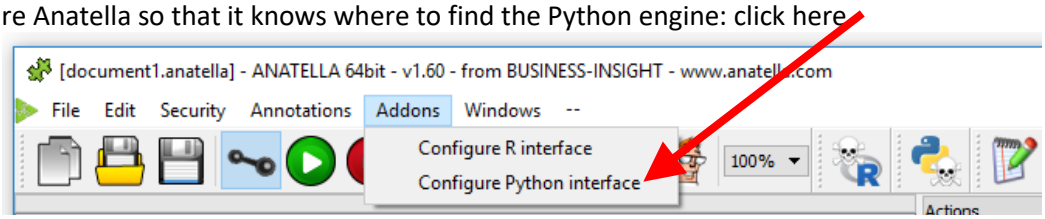
1. Install & download the Python engine named "Anaconda".
   This Python engine is available here:
   https://www.anaconda.com/distribution/

   Currently, Anatella can execute Python v3.x (and not Python 2.x: i.e. Anatella links to "Python3x.dll" only and not to "Python2x.dll").

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

For your convenience, you can also download your Anaconda Python engine (as a simple ZIP file to unzip – no administrative privileges required) from here:
http://download.timi.eu/Python/

2. Configure Anatella so that it knows where to find the Python engine: click here

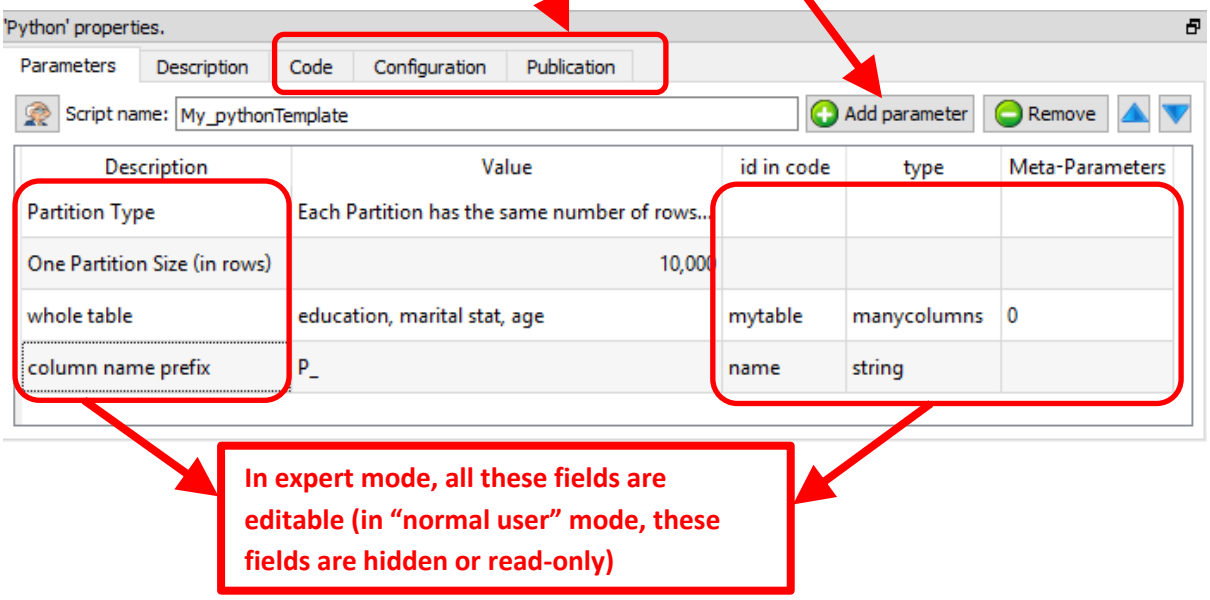


Anatella can only use the **Anaconda** Python Engine.
Anatella cannot use any v2.x Python Engine (anyway the Python 2.x has gone out of support from the whole community).

9.2.3.2. Coding in Python inside Anatella

After clicking the icon, you switched to "Expert" mode. While in "Expert" mode, you can:

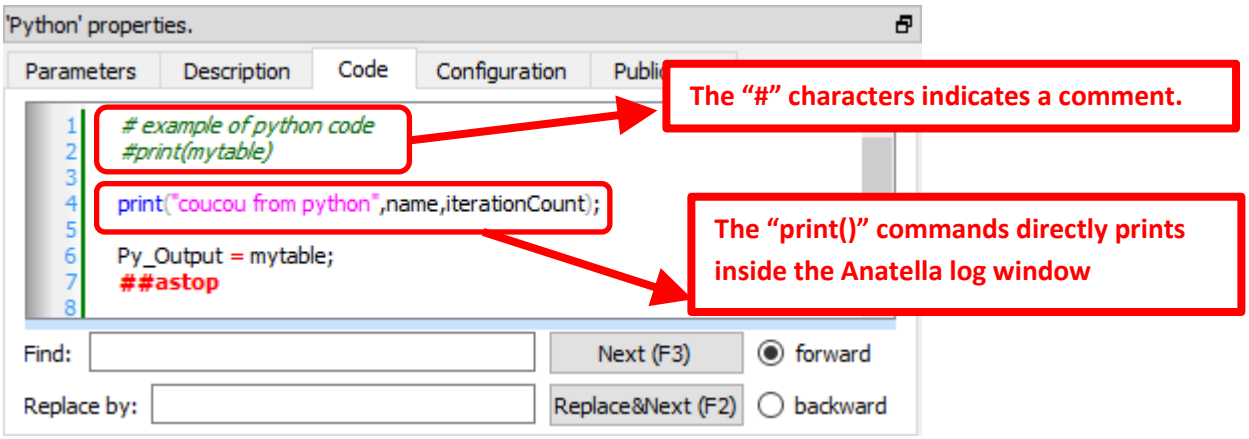
- Add new parameters (and edit the parameters) of the Action:
- See the other panels of the Python Actions:



In the above example, we can see that Anatella will initialize the Python engine with 2 variables (these 2 variables are named "myTable" and "name") before running the Python code. All the parameters that are tables (coming for the input pins) are injected inside the Python engine as "Panda data frames".

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In particular, the “Code” panel is interesting: it contains the Python code: Here is an example of Python code:

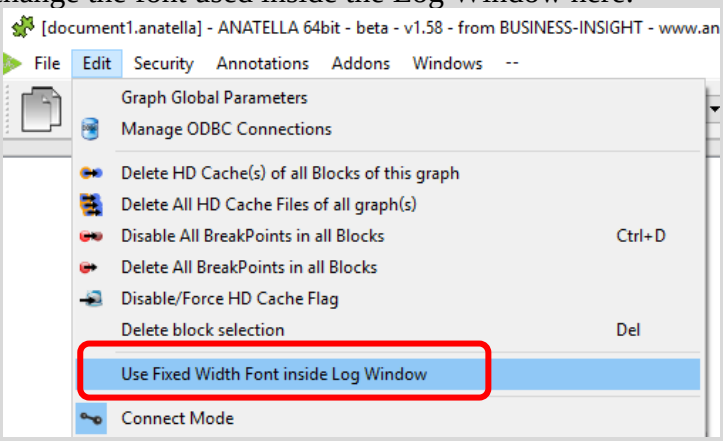


You can use the "print()"command to display some values inside the Anatella Log window (this is useful to debug your code).

When the Python engine prints some results inside the Anatella Log Window, it assumes that the font used to display the results is of **Constant-Width** (e.g. so that, when you print an array, the different columns from your array are correctly aligned on each row).

By default, Anatella uses a Variable-Width font inside the Log Window (and thus the array’s are not displayed very nicely).

You can change the font used inside the Log Window here:



(You can also use “CTRL+Wheel” to zoom in/out the text inside the Log Window)

The # characters at the start of a line indicates a comment that spans over the current line only. Inside python, if you want to easily comment several lines of codes, just write ''' (three consecutive single-quote characters) at the start and at the end of the block to comment (This is equivalent to /* … */ in other languages such as Javascript, C, C#, Java, etc).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is a special extension that is only available inside Anatella: Use the string "##astop" (without the quotes at the beginning and the end) at the very start of a line to prevent Anatella to execute any Python code located beyond the "##astop" flag. This is handy when developing new Python Actions.

Here are some "good practice" rules to follow when creating a new Python code:

1. It might be easier to use an interactive tool such as a "Jupyter Console" or "Spyder" to develop the first version of your Python code (i.e. during the first "iterations" of code development). Once your Python code is working more-or-less properly, you can fine-tune its integration inside an Anatella box using the Anatella GUI. Once the integration inside Anatella is complete, you'll have a block of Python code (i.e. an Anatella box) that you can re-use easily everywhere with just a simple drag&drop! (…and without even looking at the Python code anymore!)

When developing a new code in Python, it happens very often that the Python engine computes for a very long time (for example, because you didn't define properly a parameter) and the whole data-transformation is "freezing" abnormally for very long period of time. In such situation, don't hesitate to click the button inside the toolbar to abort all the computations prematurely. The Anatella GUI remains stable even if you cancel all the time the Python engine. This allows to make many iterations, to quickly arrive to a working code.

2. If you need a specific data-type to run your Python computations, ensure that you convert to this specific data-type before doing any computation. For example, don't assume that you'll always receive a matrix full of numbers in input: More precisely: Always force the conversion to the "number type", if you specifically need "numbers" to do your computations. To convert a Pandas data frame (received as input) that is named "myDataFrame" into an NumPy Array of numbers that is named "myArray" (…and get rid of all the strings, …and keep all the column names):

```
myArray = df_to_array(myDataFrame);
```

…where we defined the function "df_to_array()" in the following way:

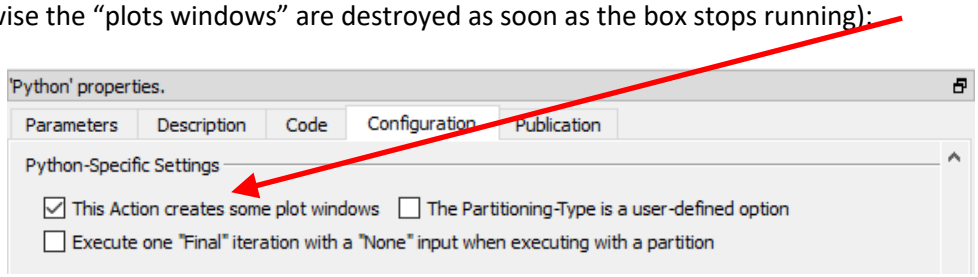
```
def df_to_array(df):
  mycols = df.columns;
  mytypes=[(mycols[i], np.float64) for (i,k) in enumerate(mycols)];
  return df.astype('float64').values.ravel().view(dtype=mytypes);
```

The function "df_to_array()" is always declared for you inside the Anatella Python engine (i.e. you don't need to declare it yourself).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

When creating a Python box that displays some plot window:

1. in the “Configuration” panel, check the option "*This Action creates some plot windows*" (otherwise the “plots windows” are destroyed as soon as the box stops running):



2. Create the required plots, as usual, using normal python commands but **DO NOT call** the function “matplotlib.pyplot.show()” at the end (i.e. Anatella will do that for you later) because this function will block all code execution (i.e. the dataflow inside Anatella will be blocked as long as the plots are visible).

   At the end of the execution of your Python code, the Anatella Python Engine will, *in order*:

   a. Signal to Anatella that the execution of the python script is complete: This means that the dataflow inside Anatella may now proceed further.

   b. Display all your plots by executing:

```
import matplotlib.pyplot as plt;
plt.show();
```

3. **Optional:** To avoid consuming much RAM memory for nothing (while Python is just busy showing your plots), add at the end of your Python code a few lines to destroy all large matrices stored in RAM. For example:

```
#free up memory:
myDataFrame=0; # replace the large data-frame named “myDataFrame” with
               # a single number (0) to reclaim RAM
gc.collect();  # run garbage collector to force Python to release RAM
```

To pass back some table-results as output of the Python box, use the “Py_Output” variable. The data-type of the variable used as output is very precise: it must be a Panda data frame (and not a Numpy Array). To convert your variables to data frames, use the following command:

```
myDataFrame = pd.DataFrame( MyNumpyArray )
```

(The above command assumes that the Numpy Array already contains the correct column names: This will be the case if you used the function “df_to_array()” to create the Numpy Array).

Here are some examples of usage of the output variable named “Py_Output”:

1. To pass on output pin 0 the data frame named “myDataFrame”, simply write:

```
Py_Output = myDataFrame
```

   Please ensure that the type of the variable named “myDataFrame” is indeed a “Panda data frame” (and not an “Array”), otherwise it won’t work.

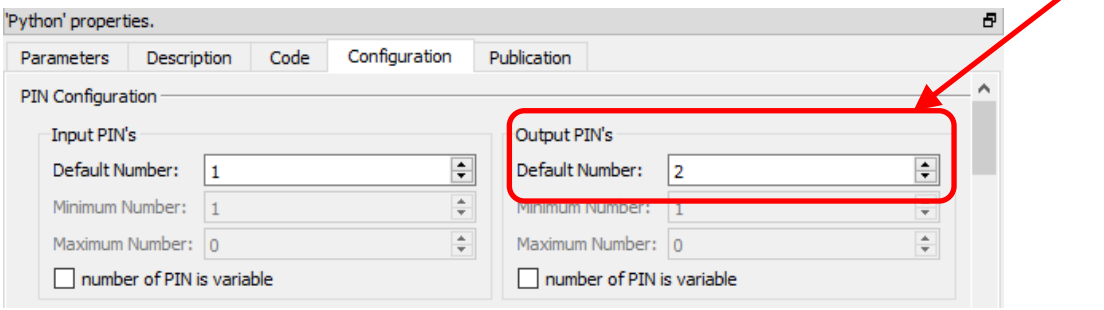
2. To pass on output pin 0 the data frame named “myDataFrame1” and to pass on output pin 1 the data frame named “myDataFrame2”, simply write:

```
R_Output = [ myDataFrame1, myDataFrame2 ]
```

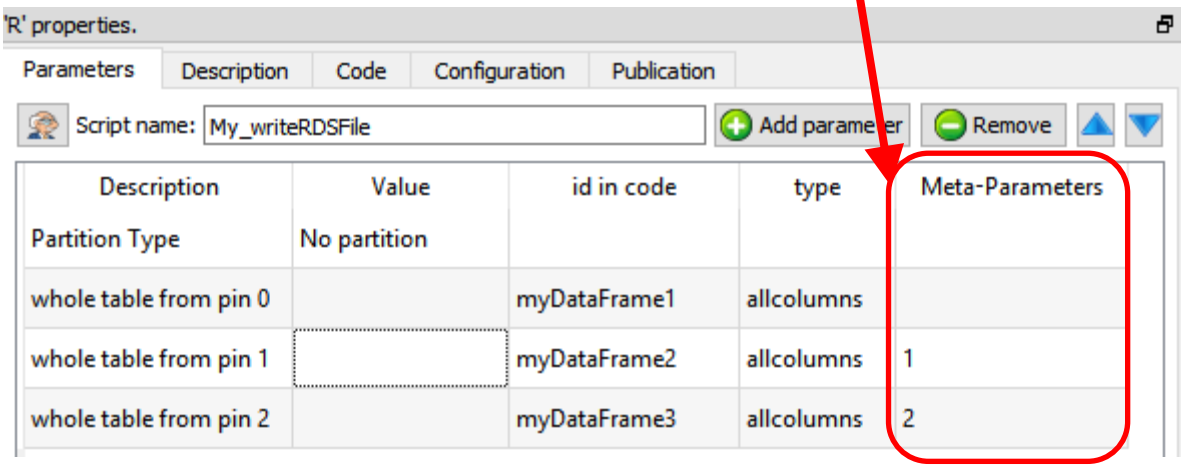

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Please be sure to use the square brackets ("[" and "]") and not the parenthesis.

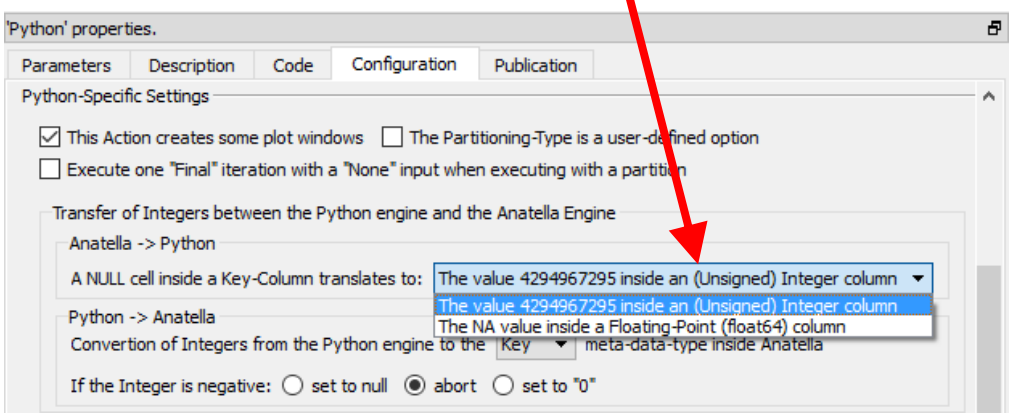
For the above code to work, you must also setup Anatella to have 2 output pins on your box:



To get inside the Python environment the input tables available on the second, third, fourth,... pins, use the column "Meta–Parameters" inside the "Parameters" panel: For example, this define 3 data frames (named "myDataFrame1", "myDataFrame2" and "myDataFrame3") that contains the tables on the input pins 0,1 & 2:



The "Pandas" library (that manages Data Frames in Python) has a limitation related to the way it stores the "null" value (i.e. the "null" from the database world). More precisely, The "Pandas" library cannot store a "null" value inside the columns with the data type "int32", "uint32", "int64" or "uint64" (if you try to set a "null" in such a column, Pandas automatically converts the column to a 'float64" column and thereafter set the null value). This is somewhat annoying because the columns with the "Key" meta-data-type inside Anatella are injected into python as columns with the "uint32" type. …and this Python-type does not support the "null" value (in opposition to Anatella that handles correctly the null values inside the column of the "Key" meta-data-type). To get around this limitation of python, we can choose between two options:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

These 2 options are:

1. **Convert Null value to the value 4294967295 (2^32-1):** This is the best & easiest option. This is perfect when the "key" columns contains "foreign keys" that are used in different joins between different tables (because the 4294967295 value will prevent the "join" to succeed, as would have done the real "null" value).
2. **Use the NA value inside a column with the floating-point ("float64") data-type:** Anatella converts the column from the "Key" meta-data-type to the "Float" meta-data-type only if required (i.e. only if the column actually contained a null). This means that, when such a column "goes out" from the Python Engine back inside the Anatella Engine, its meta-data-type is undefined: it can either be "Key" or "Float" (depending if a null value was encountered). To be sure of the meta-data-type, please add a ChangeDataType Action just after the Python Action (to transform back the "Float" columns into "Key" columns).

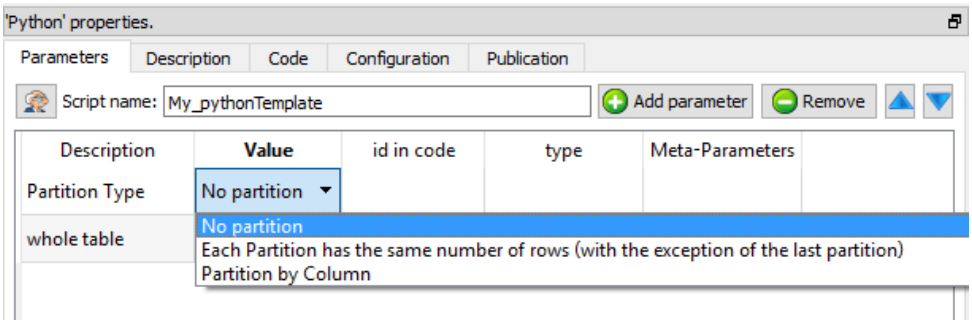
When the Anatella Python Engine starts:

- It automatically creates some Pandas Data Frames containing data originating from the current Anatella Graph.
- It automatically executes:

```
import numpy as np;
import pandas as pd;
import gc;
import sys;
```

  (This is required for the Pandas Data Frame to work).
  This means that you don't need to write the above "import" statements yourself.
- It defines the function "df_to_array()".
- It redirects stdout&stderr to the Anatella console.

The Python engine is quite limited in terms of the size of the data it can analyze because all the data must fit into the relatively (small) RAM memory of the computer. To alleviate this limitation of Python, you can ask to Anatella to partition your data. There are currently 3 different partitioning options inside Anatella:



How does it work? The table on the first input pin (i.e. on pin 0) is "splitted" in many different "little" tables (the tables on the other pins – pin 1, pin 2, pin 3, etc. – are always injected completely inside the Python engine without any "splitting"). The Python engine can process easily each of this "little" table because they only consume a small quantity of RAM memory. After the split, Anatella calls the Python engine "in-a-loop", several times: At each iteration, the Python engine process one different "little" table (and it might also produce some output).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

There are three "Partition Types":

- **No partition** (the default option): self-explaining.
- **Each Partition has the same number of rows (with the exception of the last partition)**: self-explaining.
- **Partition by Column**: When using this option, you must select a "Partitioning Column". For example, if you select as "Partitioning Column" the column "Age", then each partition will contain all the people (i.e. all the rows) with the same "Age".

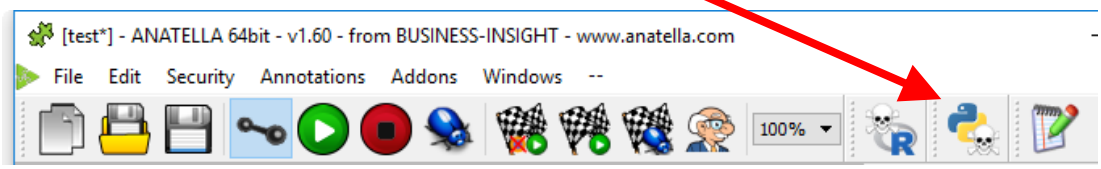
The concept of "Partition" is used many times inside Anatella: e.g. See the sections 5.5.3. (Partitioned Sort), 5.12.4 (Time Travel), 5.7.9. (Quantile), 5.5.9. (Flatten), 5.3.2.5. (Multithread Run), 5.12.2. (Smoothen Rows) where the same partitioning concept is also used.

More precisely, when using partitioning, Anatella does the following:

1. Split the table on pin 0 into many different "little" tables (one table for each different partition).
2. Inject into Python the variables that contains that data from all the rows of the tables available on pin 1, pin 2, pin 3, etc.
3. Inject into Python the variable named "partitionType" (whose value is 0, 1 or 2, depending on which "Partition Type" you are using).
4. Inject into Python the variable named "iterationCount" with the value 0.
5. Inject into Python the variable named "finished" with the value "false".
6. Run the loop (i.e. process each partition):
   - Inject into Python one of the "little" tables (that are coming from the "big" table available on the first input pin – input pin 0).
   - Run your Python code inside the Python engine.
   - Get back some output results to forward onto the output pin.
   - Increase by one the value of the Python variable "iterationCount".
   - Execute the next iteration of the loop (i.e. re-start the step 6) until there are no more "little" tables to process.
7. If the Anatella option "*Execute one "Final" iteration with a NILL input when executing with a partition*" is checked (i.e. it's TRUE):
   - Set the variable named "finished" to the value "true".
   - Reset the variables that contains the "little" tables to NILL.
   - Run the Python engine one last time.
   - Get back some output results to forward onto the output pin.

One example where partitioning "makes sense" is when you want to use a predictive model to score a new dataset. In opposition to "learning datasets", "scoring datasets" can be really big (so big that they don't fit into RAM). When you use a partition, **you can compute the predictions for any (scoring) dataset, whatever the size of the dataset.** Nice! ☺

There is also a "Kill Python" button inside the toolbar here:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This button kills all the Python engines currently running. This means that, when you click this button:

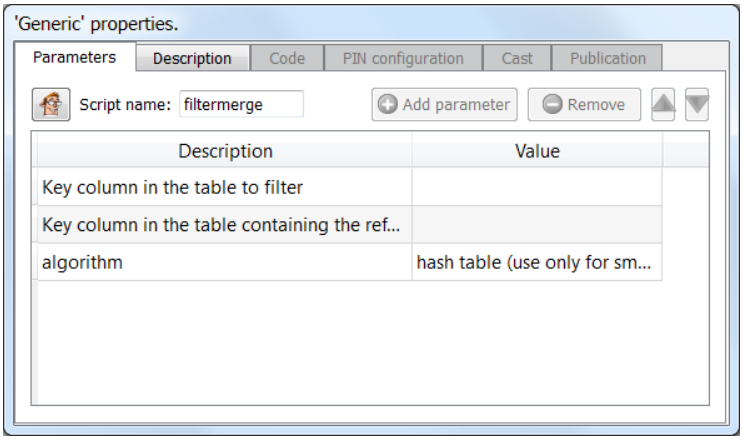
- All (Python-based) plot-windows are closed (this is very handy!)
- All Python computations are stopped (i.e. don't click this button when your graph is running!).

Use the button inside the toolbar to close in one click all the Python-based-plot-windows.

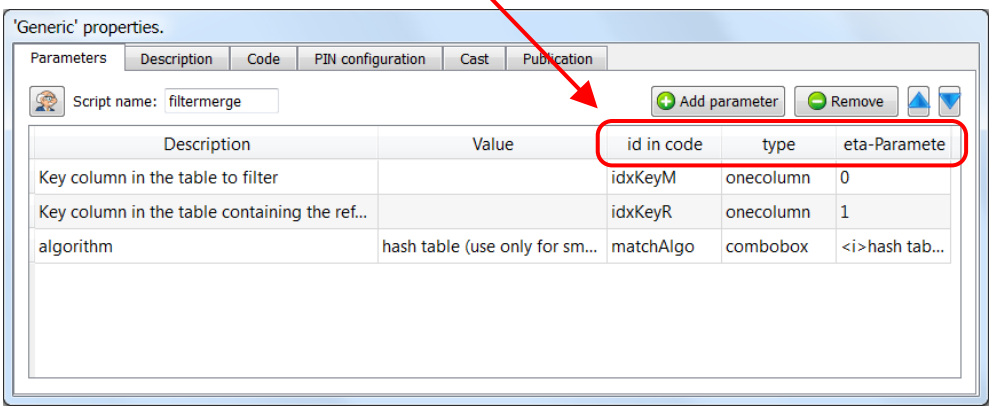
Once you are satisfied with your Python-based Action, you can "publish it" so that it always becomes available inside the "common" re-usable Actions: See section 9.7. for more details.

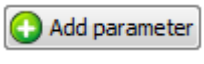
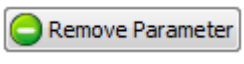
9.3. Defining script parameters

The script parameters are defined in the first tab (named "parameters"). For example, here are the script parameters for the "Filtermerge Action":



When you switch to expert user mode: (Click the button in the main toolbar of the application), three additional columns appear:



You can click on the  button or the  button to add or remove script parameters.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The columns that just appeared are:

1. "id in code": This column defines the name of the JavaScript/R/Python variable that will be created (and initialized properly) automatically by Anatella before running your script. The initialization technique of the variable depends on the parameter type (that is defined in the next columns).

2. "type": the parameter type. This type can be:
   - String
   - Double
   - Boolean
   - ComboBox
   - Onecolumn
   - Manycolumns
   - Onefilesave
   - Onefileopen
   - ManyFileOpen
   - OneDir

   Anatella will adapt the interface to the parameter type: For example: if the parameter type is "Boolean", then you will see a checkbox. See the next sections for more information about each type.

3. "Meta-Parameters": Some additional parameters that controls how the parameter must be handled by Anatella. See the next sections for more information.

9.3.1. The script parameter type "String"

Value Editor: When the user wants to edit a variable of this type, a simple in-line text box appears. The user can enter any string.

Meta-parameter: none

9.3.2. The script parameter type "Double"

Value Editor: When the user wants to edit a variable of this type, a simple in-line text box appears. The user can enter any number with a precision of 2 decimal places. If you need more precision, you can:

- use the standard "String" type and convert the string value to a number in the JavaScript code using the "*Number()*" function.
- Write into the meta-parameter column a number that is the number of decimal places.

Meta-parameter: number of decimal places (default is 2).

9.3.3. The script parameter type "Bool"

Value Editor: When the user wants to edit a variable of this type, a simple checkbox appears.

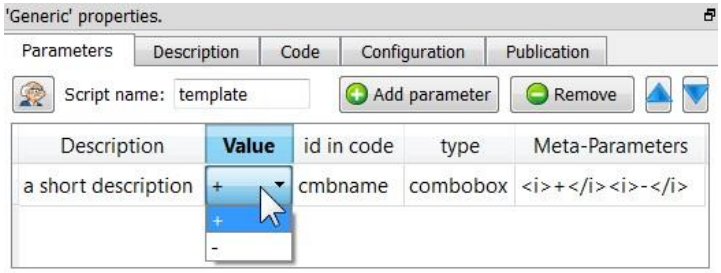
Meta-parameter: none

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.3.4. The script parameter type "Combobox"

Value Editor: When the user wants to edit a variable of this type, a simple combobox appears. The different elements inside the combobox are defined inside the Meta-Parameters.

Meta-parameter: The Meta-Parameters contain the list of elements that must be included inside the combobox. Each element is inside a XML-Tag named "i". For example:



The JavaScript variable associated with the Combobox is a number. This number is the index of the selected option inside the combobox. In the above example:

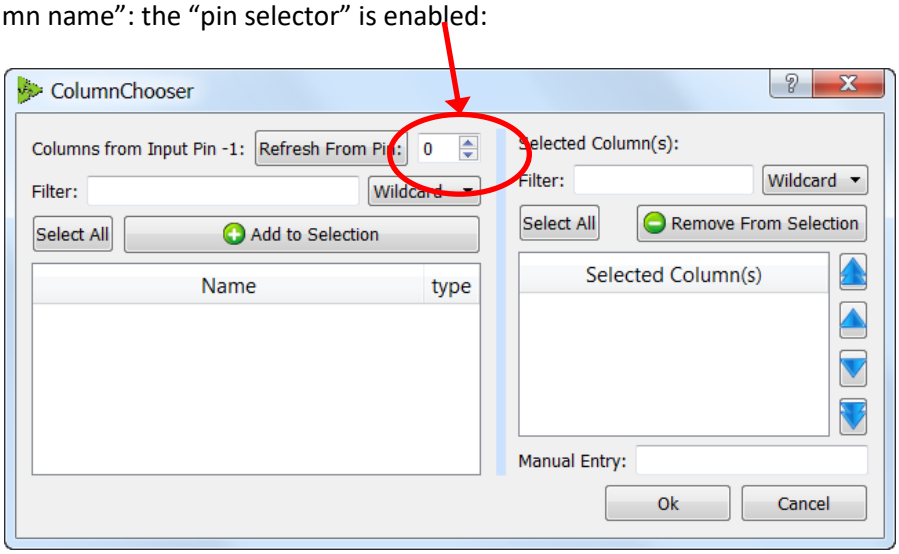
- If the user selected the option "+", the JavaScript variable "cmbname" is equal to 0.
- If the user selected the option "-", the JavaScript variable "cmbname" is equal to 1.

9.3.5. The script parameter type "OneColumn"

Value Editor: When the user wants to edit a variable of this type, the standard Anatella "column chooser" window appears (see section 5.1.4. to know how to operate such window).

Meta-parameter: The meta-parameter is a number. There are two completely different operating modes for this meta-parameter:

- **Operating mode 1: The meta-parameter is "–1" (only available for Javascript-based Actions).** In this mode, the user can freely choose from which input pin he will "collect" the required "column name": the "pin selector" is enabled:

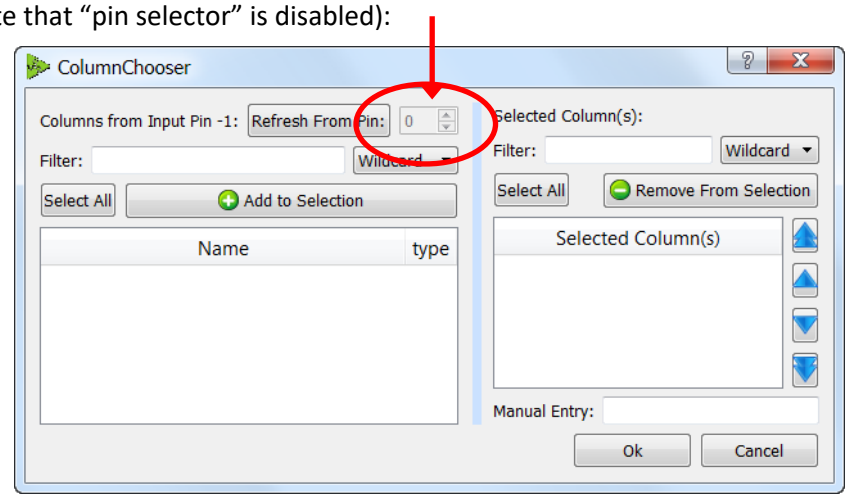


The Anatella-script-variable will be initialized with a **string** that is the selected column name. You must still use into your Anatella-script the function "*rowGetIndexOfColumn(RowObject,String)*" to obtain the corresponding column index.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- **Operating mode 2: The meta-parameter is a positive number 'p'**

  In this mode, the user is forced to choose one "column name" that is coming from the input table on pin 'p'. On this example, the user must select a column name from the input pin 'p=0' (note that "pin selector" is disabled):



  For Javascript-based Actions: The script-variable will be initialized with a **number** that is the index of the selected column inside the table available on input pin 'p'.
  For R-based or Python-based Actions: The script-variable will be initialized with the content of the selected column inside the table available on input pin 'p'.

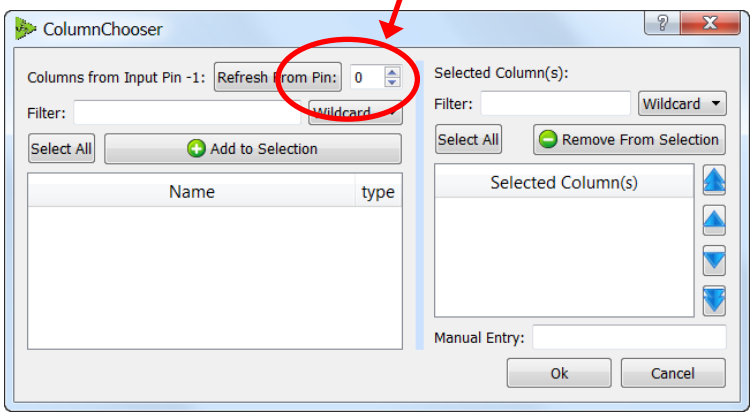
To summarize, depending on the value of the Meta-parameter, the value of the Script variable linked to this parameter will be computed in a totally different way.

9.3.6. The script parameter type "ManyColumn"

Value Editor: When the user wants to edit a variable of this type, the standard Anatella "column chooser" appears (see section 5.1.4. to know how to operate such window).

Meta-parameter: The meta-parameter is a number. There are two completely different operating modes for this type of variable:

- **Operating mode 1: The meta-parameter is "–1" (only available for Javascript-based Actions).**
  In this mode, the user can freely choose from which input pin he will "collect" the required "column names": the "pin selector" is enabled:

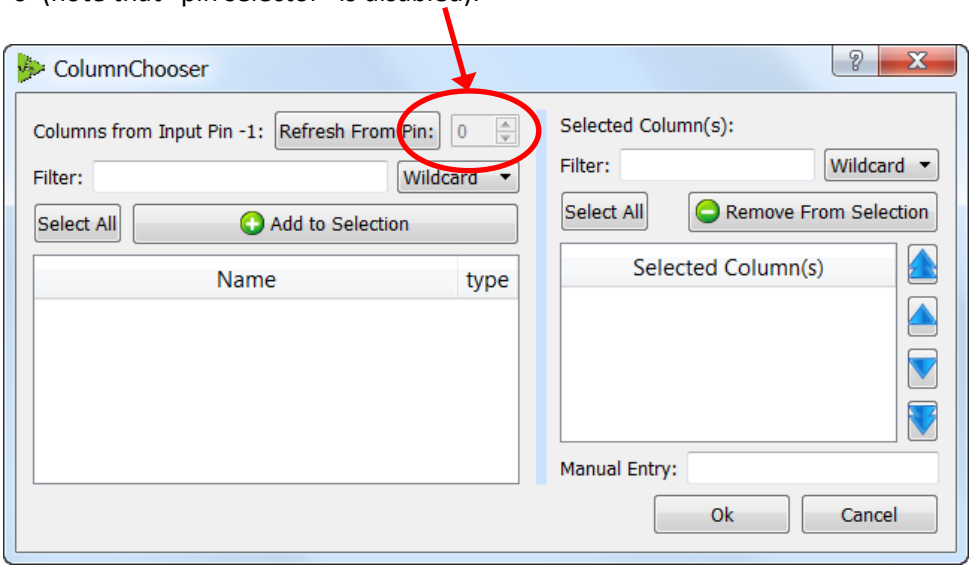


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The Anatella-script-variable will be initialized with an **array** of **strings** that are the selected column names. You must still use into your Anatella-script the function "*rowGetIndexOfColumn(RowObject,String)*" to obtain the corresponding column index.

1. **Operating mode 2: The meta-parameter is a positive number 'p'**

In this mode, the user is forced to choose one "column name" that is coming from the input table on pin 'p'. On this example, the user must select a column name from the input pin 'p=0' (note that "pin selector" is disabled):



For Javascript-based Actions: The script-variable will be initialized with an **array** of **numbers** that are the indexes of the selected columns inside the table available on input pin 'p'.
For R-based or Python-based Actions: The script-variable will be initialized as a dataframe that contains the selected columns inside the table available on input pin 'p'.

To summarize, depending on the value of the Meta-parameter, the value of the Script variable linked to this parameter will be computed in a totally different way.

9.3.7. The script parameter type "OneFileSave"

Value Editor: When the user wants to edit a variable of this type, a "file chooser" dialog appears. The user can enter any filename.

Meta-parameter: none

9.3.8. The script parameter type "OneFileOpen"

Value Editor: When the user wants to edit a variable of this type, a "file chooser" dialog appears. The user can select **one** filename from the list of available files.

Meta-parameter: none

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.3.9. The script parameter type "ManyFileOpen"

Value Editor: When the user wants to edit a variable of this type, a "file chooser" dialog appears. The user can select **many** filenames from the list of available files.

Meta-parameter: none

9.3.10. The script parameter type "OneDir"

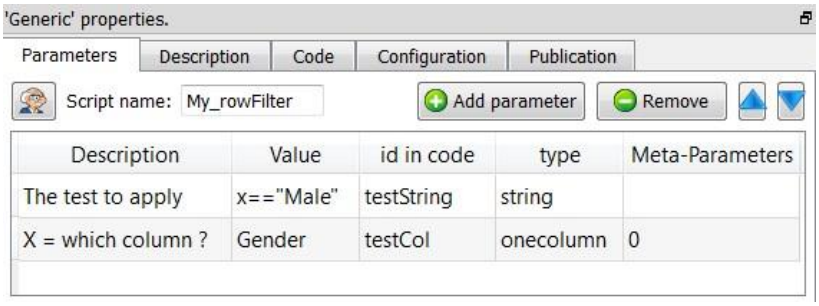
Value Editor: When the user wants to edit a variable of this type, a "directory chooser" dialog appears. The user can select any directory (and also create a new one if required).

Meta-parameter: none

9.4. Using script parameters

9.4.1. Script Parameters From the "Script-Parameter-tab" of the Action-Property-Window

Using script parameters is really straight forward and easy. For example, here is the "Script Parameters tab" for a simplified version of the "Row Filter Action":



As you can see here above, we asked Anatella to define two new parameters:

1. A first parameter named "testString". This parameter can contain any string but the user is required to enter here a valid expression that will be used to "Filter the rows": if the expression is true, then the row is allowed to "go" to the output pin 0, otherwise, the row is dropped.
2. A second parameter named "testCol" : Since the meta-parameter value is zero, we can say that this variable will contain the index of a user-selected column inside the table on input pin 0. (This will be the index of the column to test).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can directly use these parameters in your “row filter” JavaScript-based Action:

```
1   var isOK;
2
3   function parallelRun(v) { return true; }
4
5   function init()
6   {
7     var r=getCurrentRow();
8     setOutputRowSize(0,r.nColumn);
9     isOK=new Function("x", "return "+ testString);
10    return 0;
11  }
12
13  function run()
14  {
15    var r=getNextRow();
16    if (r.isNull) return 1;
17
18    var x=r.col( testCol );
19    if (isOK(x))
20    {
21        r.write(); writeEOL();
22    }
23    return 0;
24  }
```

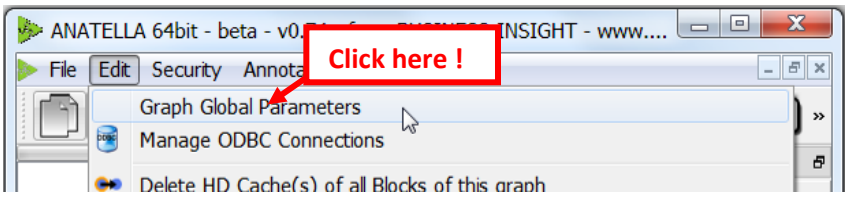
On line 9, we define a new function, that is named “isOk()”. This function will be used to test if a row is “allowed to pass”. The code of the new function “isOk()” is based on the user-defined-script-parameter “testString”.

On line 18, we extract the content of the user-selected-column on the current row and put it into the “x” variable. To extract the content of a specific column, we used the method “*col(columnIndex)*” of the Row object “r”.

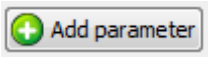
On line 19, we test the “x” variable using the “isOk()” function to check if the row is “allowed to pass”.

9.4.2. Script Parameters From the “Graph Global Parameter” window (Javascript Actions only)

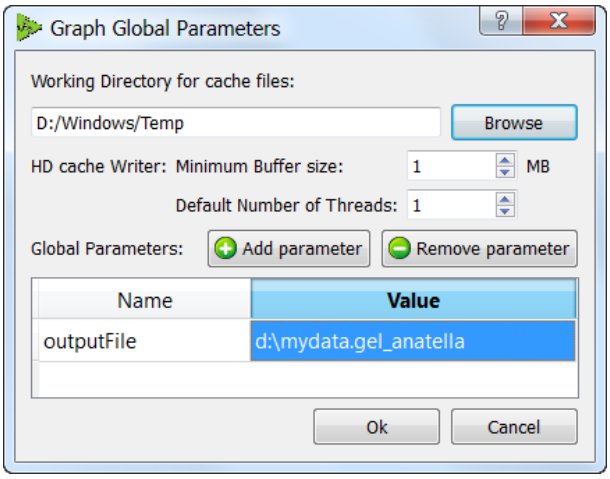
You can also define “Global parameters” that will be shared amongst all the Javascript Actions inside your Anatella-Graph. To do that, click on the “*Graph Global Parameters*” option in the “*Edit*” drop-down-menu:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The “Graph Global Parameter” window appears. Click on the 

button to add a new global parameter. For example, let’s define a new global parameter named “outputFile” that is initialized with the “d:\mydata.gel_anatella” value:



From now on, the JavaScript variable “outputFile” will be available in all your Javascript-based Actions (and it will have the value “d:\mydata.gel_anatella”).

Many Non-Javascript Actions (i.e. Many Actions developed in C/C++) are also making use of the “Graph Global Parameter”: For example, the readCSV Action, the

ReadSAS Action, the ReadGel Action, the writeCSV Action, the WriteGel Action are all able to define their “file name” using a javascript code that can include some references to “Graph Global Parameters”.
When executing an Anatella-Graph in batch-mode, you can define, on the command-line any number of “Graph Global Parameters”: see section 4.7.1.

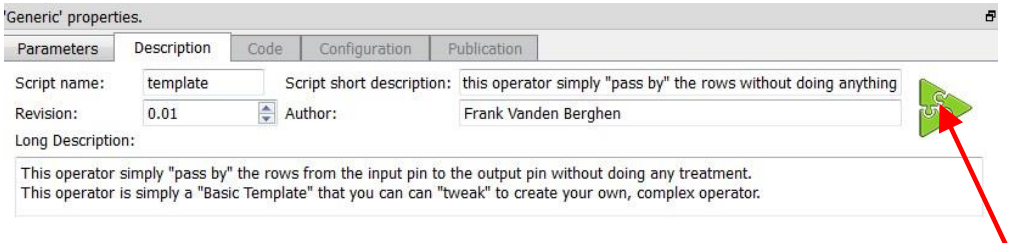
“Graph Global Parameters” are also essentials when using the “Parallel Run ” Action: see section 5.3.3.

From inside a Javascript Action, you can change the value of a “*Graph Global Parameter*”. This change will be propagated to all the places where the “*Graph Global Parameters*” is used. This allows a primitive form of “global” communication between all the Actions (e.g. Two actions that are not even connected together can communicate together through a “*Graph Global Parameters*”). For example, this mechanism is interesting when you want to initialize a “*Graph Global Parameters*” to a specific value that is computed using a Javascript code.

9.5. Editing the Action descriptions

The “Description Tab” allows you to edit the different descriptions of your script-based Action.
Here is a screenshot:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

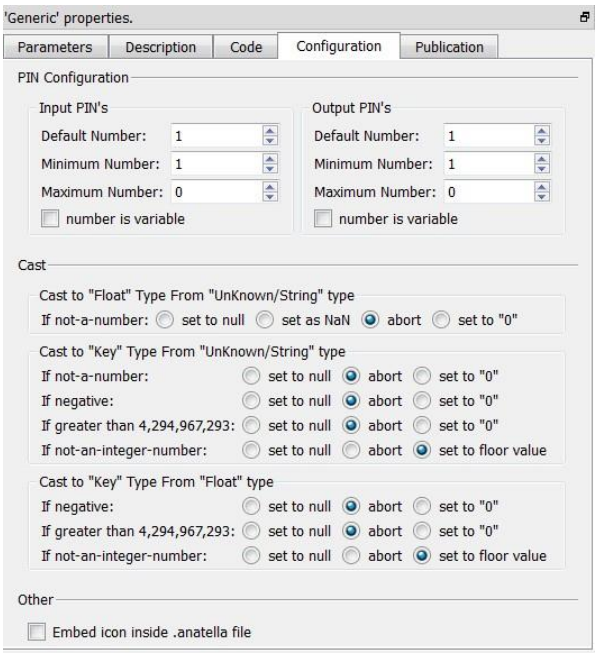


When in "standard user mode", this window is 100% read-only.

When in "expert user mode", you can change all the texts AND you can click on the icon here to change the Action icon! The icon can be any PNG file that is inside the "plugin" directory of the Anatella application. The recommended size for the icon is 48x48.

If you publish your Action as a "*re-usable script*", the "*operator short description*" will appear as a "tooltip" text above the icon of your Action in the "*Transformation Actions*" list-window.

9.6. The Configuration Tab



9.6.1. Changing the pin configuration of a script-based Action

The "pin configuration" section allows you to configure the input and output pins of your script-based Action. The settings on this section are self-explanatory.

The "*minimum*" and "*maximum*" settings are only used when the option "*number is variable*" is checked. The value '0' for the "*maximum*" setting has a special meaning: it means that no maximum is defined and the user can use as many pins as he wants.

9.6.2. Cast

A very common operation when programming in any language is to convert a value from one data-type to another (e.g. we want to convert a string to a floating-point number). In technical terms,

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

changing the data-type of a value is named "*casting*". For example: This will force Anatella to cast the String "42" to a number (because the first column of the Row Object "r" can only contain floating-point numbers):

```
var r=new Row(1);
rowSetMetaType(r,0,'F');
r.setColumn(0, "42");
```

The above example works nicely, but what happens if we write `r.setColumn(0, "foobar");` ?
This cast will fail (because it's not possible to cast "foobar" to a number).

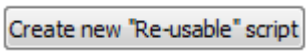
The "Cast" box defines how Anatella must react to "erroneous" casting. By default, for the above example, Anatella will abort the execution of the data-transformation-graph.

9.6.3. Embed icon inside an .anatella file

When you are sending an .anatella graph containing a brand-new Javascript/R/Python Action to a colleague, you are actually sending (by default), only the "code" of the Action (so that your colleague can execute the Action) but not the icon of the Action (it is assumed that the associated .png file, defining the icon of the Action, is already available inside the "plugin" directory of your colleague: See section 9.8. for more information of the "plugin" directory). In such situation, the display of the Action will revert to the "default" icon (because Anatella can't find the .png file for the icon):

To have a "nice" display, you can:

1. Enable the option "*Embed icon inside .anatella file*". When this option is enabled, Anatella store the whole .png file containing the icon of the Action inside the .anatella file: The .anatella file now contains both the code and the icon of the Action (so that your colleagues can see how good your artistic skills are! 😉). This is the easiest solution.
2. Send the associated .png file (located inside your "plugin" directory) alongside with your .anatella file.

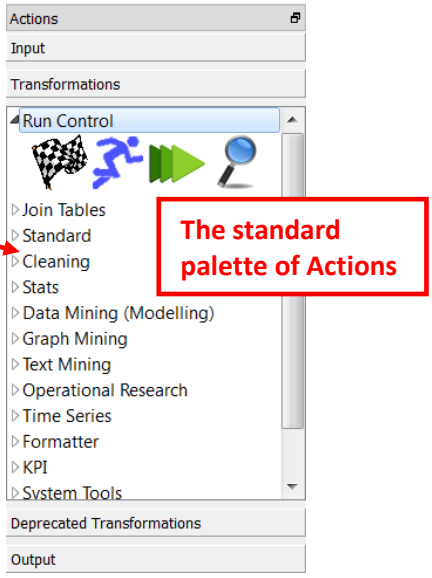
When your click the 

button in the "publication" tab of the Javascript Action (see section 9.7. about this button), Anatella extracts the icon embedded inside the .anatella file and place it inside your "plugin" directory. For now on, since you now have the (new) .png file, the display of this Action will always be ok.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.7. Publishing, Maintaining and Sharing Script-Based Actions

9.7.1. Creating New Script-Based Actions



Once you arrived to a working JavaScript/R/Python Action, you might be interested in adding this script to the standard palette of Actions available inside Anatella:

Adding a script in the standard palette of Actions is also named "publishing a script".

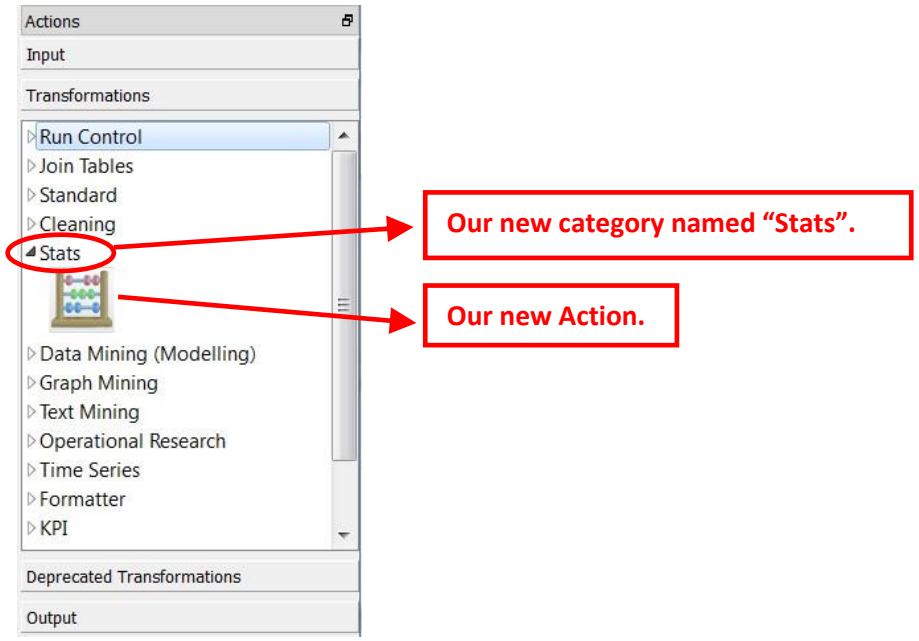
Before any publication, you must give to the script a proper "*script name*" (i.e. one that doesn't start with "My_"). The "*script name*" cannot contain any special characters (since the "*script name*" is also used as the filename of the .xml file that contains your script).

Beside the "*script name*", you should also select some "categories" in which the new Action will appear. The default categories are:

Category Name
050_Run Control
100_Join Tables
150_Standard
200_Cleaning
250_Data Mining (Modeling)
300_Graph Mining
350_Text Mining
365_R Visualization
370_R Predictive Analytics
375_R Discovery Analytics
376_R Scoring
380_Python
400_Operational Research
450_Time Series
470_Stats
500_Formatter
550_KPI
600_System Tools
610_Distributed Computations
700_Cloud Services

You can also define your own categories. For example, if we decide to create a new Action inside the new category "210_Stats", Anatella will automatically insert the category "210_Stats" in between the categories "200_Cleaning" and "250_Data Mining":

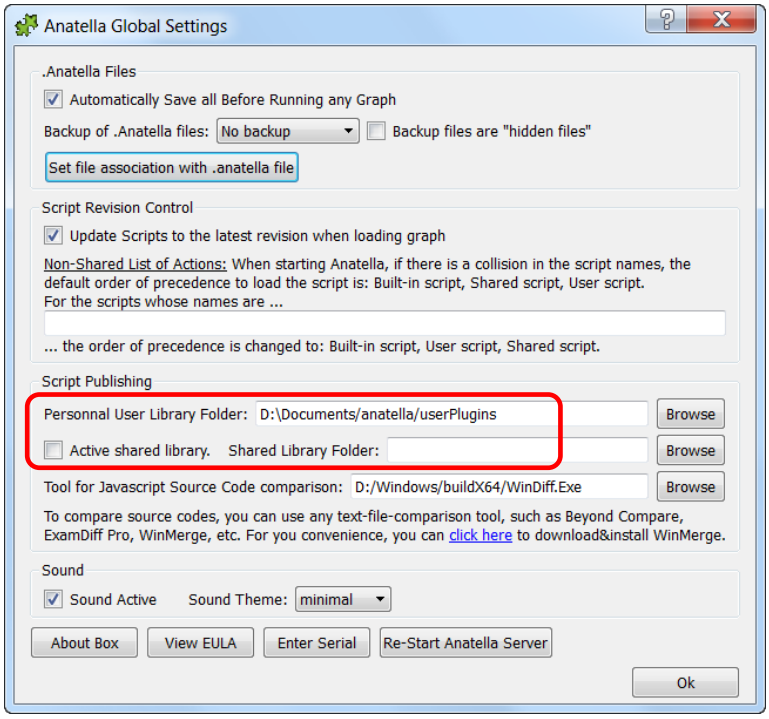
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



When you publish a script, Anatella creates a new XML file that contains the published Action inside one of these 3 folders:

1. **The Built-In folder:** Usually "c:\soft\TIMi\plugin" or "c:\Program Files\TIMi\plugin"
   The Actions (i.e. the .xml files) that are saved inside this directory belongs to the standard, official distribution of Anatella. When you un-install (or re-install) Anatella, all your user-defined Actions inside this folder will be deleted.

2. **The User folder:** Usually: "c:\users\<my name>\documents\anatella\userPlugin".
   The Actions (i.e. the .xml files) that are saved inside this directory belongs to current user. The Actions inside this folder will never be deleted, even when un-installing (or re-installing) Anatella. You can choose the location of your "user folder" in the "Global Settings" window: See illustration below.



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

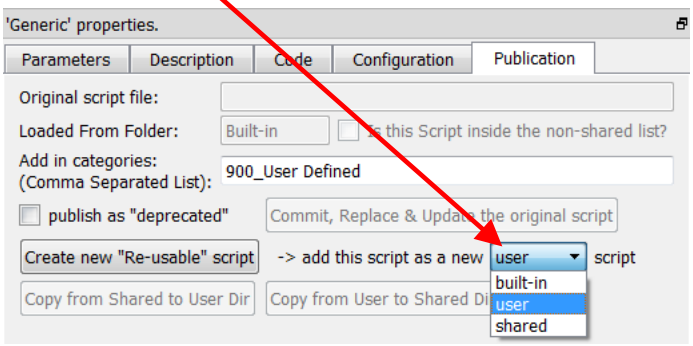
3. **The Shared folder:** This folder is "synchronized" between all the Analysts that are working collaboratively together. The "synchronization" of the files inside the Shared Folder can be achieved using:
    - a versioning system (Git, SVN, mercurial, etc.)
    - a network share
    - a "DropBox" or a "Google Drive" synchronized folder.

    The Actions (i.e. the .xml files) that are saved inside this directory are shared between all the Anatella users that are working together.
    For example:
    - You publish a new script to the "Shared Folder" on your machine.
    - The files inside your "Shared Folder" are replicated (This is automatic if you use an automatic synchronization tool such as DropBox), to the "Shared Folder" of all your collegues.
    - At the next start of Anatella, your collegues will see your new Action inside their own standard palette of Actions on their own computer.

    The "Shared Folder" allows easy collaboration between several analysts working on the same problem. You can enable/disable and choose the location of your local "Shared folder" in the "*Global Settings*" window: See illustration above.

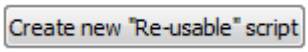
Each XML file in these folders is a different self-contained "re-usable" script.

You can select the publication folder ("built-in", "user" or "shared") in which your new "re-usable" Action will be saved (as an .xml file) using this combo-box:



Once you have:

- decided of a proper "*script name*" (remove the "My_"),
- selected the required categories to which your new Action belongs to,
- selected the publication folder ("built-in", "user" or "shared"),

…you can press the 

button in the "publication" tab and re-start Anatella.

Your new script should now appear inside the "Transformation" list in the required categories (or inside the "Deprecated Transformations" list if you checked the "*Publish as 'deprecated'*" option), alongside with all the other re-usable scripts.

Now that, once your script is published, you still have to "maintain" it. Maintenance includes bug fixes and improvements. Anatella includes a "versioning system" that helps you "maintain" your scripts: See the sections 9.7.2. and 9.8. about this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.7.2. A Typical Workflow

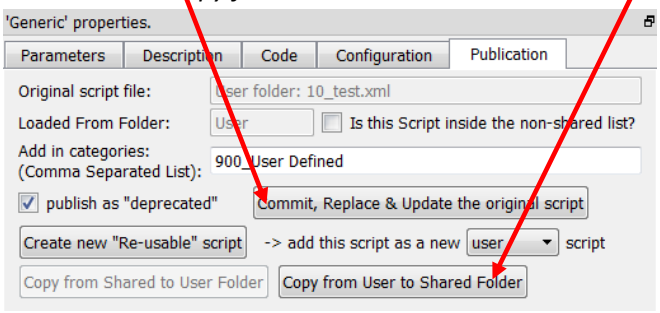
Typically, an analyst working on a Javascript Action will follow the following steps:

- The analyst selects an already existing Script-based Action and edit it to reach its objective. When an Action is being edited, you'll notice a small "E" letter inside the Icon Action:

- The analyst publishes this Action as a new Re-Usable Script inside the "User" folder. All Actions inside the "User" folder are marked with a small "U" letter:

- Let's now assume that the Analyst wants to make some improvements to its script: it will change its script and publish the new modified version using the "*Commit, Replace & update the original script*" button.

- Following some request from its collegue, the analyst decide to publish its Action inside the "Shared" folder: It clicks the "*Copy from User to Shared Dir*" button:



If the "*Copy from User to Shared Dir*" button is disabled, this means that your "shared folder" is disabled. To enable the "*Copy from User to Shared Dir*" button, Click the "*Active Shared library Folder*" inside the "Global Settings" window: Refer to the section 7. for more information about the "Global Settings" window.

All the Actions inside the "Shared" folder are marked with a small "S" letter:

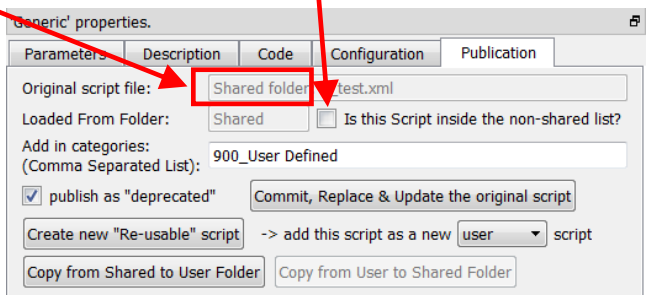
- There are now 2 .xml files that represents the same "Re-Usable" Script-Based Action:
  - One .xml file is inside the "User" Folder.
  - One .xml file is inside the "Shared" Folder.

  When Anatella starts, if the same script is present inside both the "User" folder and the "Shared" Folder (i.e. there is a name collision), Anatella only uses, by default, the version that is inside the "Shared" Folder (and completely discard, by default, the .xml file inside the "User" Folder).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You can change the default behavior by adding the script inside the non-shared list: click here:

- Let's now assume that the Analyst wants to modify its script again (to improve it or to fix it). There are two different options here:
  1. The analyst directly modifies the script located inside the "Shared Folder":
     The procedure to do that is the following:
     a. verify that the "*Original Script File*" parameter is located inside the "Shared Folder":



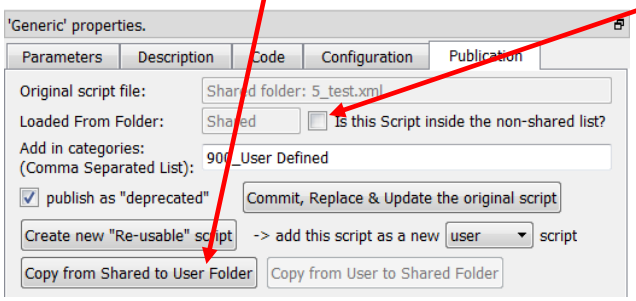
If the "*Original Script File*" parameter is erroneously set to "*User Folder*", you can either:
1. remove the script with the same name that is inside the "*User Folder*".
2. drag&drop a new instance of the Action from the palette and make the required modifications directly, without saving the graph.

     b. Make the required modifications and click the "*Commit, Replace & update the original script*" button.

  2. The Analyst can make a copy of the script located inside the "Shared" folder to its own "User" folder (i.e. do a "checkout"), work on the copy located inside the "User" folder and, once it's ready, copy it back into the "Shared" folder (i.e. do a "commit").

     This way of working offers the best flexibility and a higher security: Since the analyst is working on its own local copy, there is no danger of "breaking" the script located inside the "Shared" Folder that is used by everybody.

     The procedure to do that is the following:
     a. Click the "*Copy from Shared to User Folder*" button to get a copy of the Action inside your "User" Folder.
     b. By default, the copy located inside the "*User*" Folder (that we just created) is totally ignored by Anatella (the copy located inside the "Shared" folder gets the preference). To be able to use the copy located inside the "*User*" Folder, you must add the script inside the "*non-shared list of Actions*": Click here:



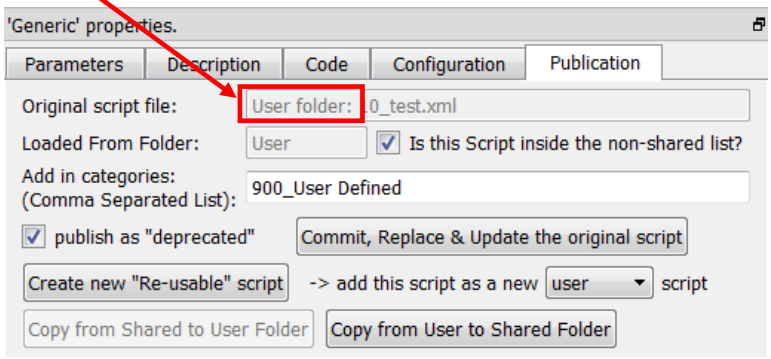
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The Actions that are inside the "non-shared list" are never read or written to the "Shared" folder (as long as they also exists inside the "User Folder").

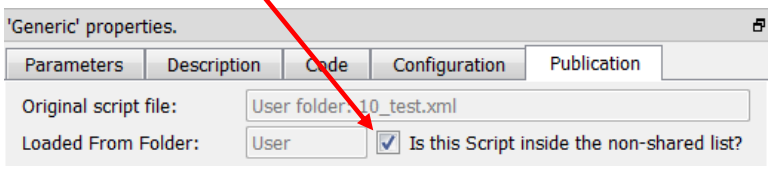
Re-start Anatella. You should now notice that your Action is now marked with the "U" letter again (and not the "S" letter anymore):

This means that you are now correctly working on the "User" folder again.

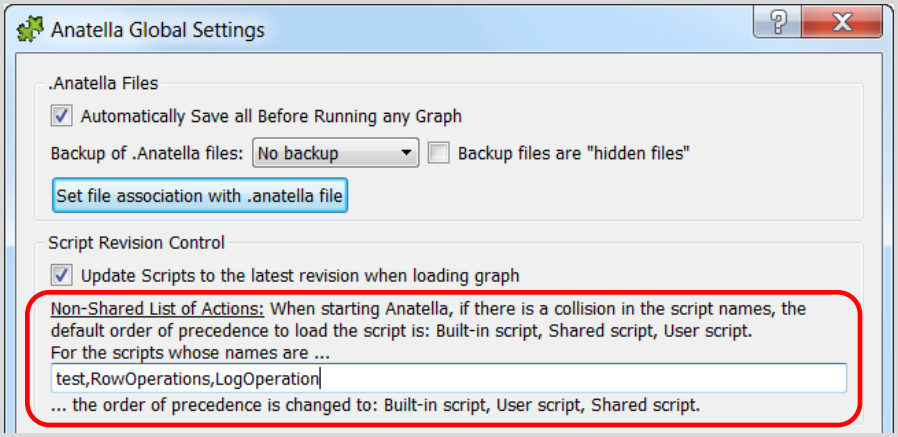
c. Verify that the "*Original Script File*" parameter is located inside the "*User Folder*", make the required modifications, click the "*Commit, Replace & update the original script*" button as many times as required.



d. Let's now assume that the modifications are now complete and tested: We are ready to "commit" back our script to the "Shared folder": Click the "*Copy from User to Shared Folder*" button" and remove the script from the "non-shared list" (i.e. click here)



You can see & edit the complete list of "non-shared action" inside the "Global Setting" window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.8. Anatella-Scripts automatic versioning

The automatic versioning system of Anatella transparently updates your Javascript/R/Python Actions to the latest revision to ensure you that you are always using the latest and best version of all your customized & precious Anatella-scripts-Actions.

You can always disable the automatic update system by un-checking the "*Update scripts to the latest version when loading graph*" option inside the "*Global Settings*" window.

The update mechanism works this way:

a) When Anatella starts, it loads in memory all the "re-usable" Script-Based actions from the hard-drive. These actions will be used as "references" later on (when loading Anatella graphs). Each different "re-usable" script-based Action is actually loaded from a different .xml file. These .xml files are located in three different folders:
   1. **The Built-In folder:** Usually: "c:\program file\TIMi\plugin".
   2. **The User folder:** Usually: "c:\users\<my name>\documents\anatella\userPlugin".
   3. **The Shared folder:** Usually: a network share or a drop-box synchronized folder.

   (See section 9.7. for more information about these different folders).

   It can happen that one specific Script-Based actions is defined by an .xml file located inside the "User Folder" and also by an .xml file located inside the "Shared Folder" (these two .xml file share the same name/ID). In such situation, by default, Anatella uses the .xml file located inside the "Shared Folder" (unless the Action is listed inside the "*non-shared list of action*". In which case, Anatella uses the .xml file located inside the "User Folder").

b) When Anatella opens a data-transformation-graph for editing, it checks all the Script-Based actions contained in the .anatella graph file against the reference "Re-Usable" Actions currently in memory: More precisely, the check succeeds if:
   - The "*script name*" of the loaded Action matches the "*script name*" of a reference Action XX currently in memory.
   - The "Auhor" of the loaded Action matches the "Auhor" of the same reference Action XX currently in memory.
   - The "Revision Number" of the loaded Action is inferior to the "Revision Number" of the same reference Action XX currently in memory.

   When the check is successful, the JavaScript/R/Python code of the loaded Script-Based-Action is replaced (i.e. "upgraded") by the code of the corresponding "reference" Action XX currently in memory. This allows to automatically "upgrade" old Javascript/R/Python code to new code.

As you can see, the versioning system relies heavily on the "*script name*" of the Actions.

Each time that you customize the source code of an Action, Anatella automatically adds the "My_" prefix to the "*script name*" of the customized Action. This automatic change is required to prevent Anatella to make any "abusive update" of your customized Action (an "abusive update" is an update that replaces & destroys your nice&new customizations with the standard JavaScript/R/Python source code of one "reference Script" currently in memory).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

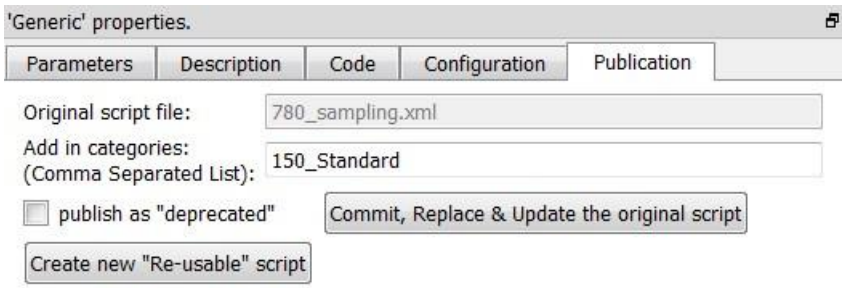
There are basically two reasons why you might want to change the source code of an Action:

1. **You are developing a completely new transformation.**

   Click the Create new "Re-usable" script button in the "publication" tab and re-start Anatella: Your new script should now appear inside the "Transformation" list in the required categories (or inside the "Deprecated Transformations" list if you checked the "*Publish as deprecated*"" option), alongside with all the other re-usable scripts.

2. **You are correcting/fixing a bug that you found inside the JavaScript/R/Python source code.**

   Once you have finished working on the source code, simply click on the Commit, Replace & Update the original script button inside the "publication" Tab:



   …and restart Anatella. From now on, each time that you open an Anatella-Graph, Anatella will automatically uses the latest modifications that you just committed (unless you de-activated the "*Update scripts to the latest version when loading graph*" option in the "*Global Options*" window).

9.9. Executing Scripts inside a N-Way Multithread section

For more information about N-Way multithreaded sections, see section 5.3.2.4.

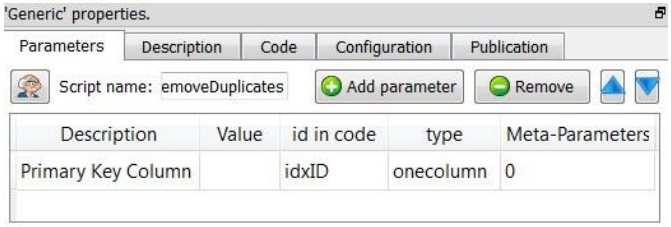
Not all Actions can be included inside a N-Way section.

9.9.1. For JavaScript-based Actions

Some Action can be included inside a N-Way section only if the partitioning variable of the N-Way section is a specific variable. For example: the "Flatten" Action can be included inside a N-Way section only of the partitioning variable of the N-Way section is equal to the "Key Column" parameter of the "Flatten" Action.

Let's take an example: Let's write in JavaScript a "simple Row De-Duplicate" (more or less equivalent to the NaïveDeDuplicate Action). There is only one parameter to the SimplifiedRemoveDuplicate Action: Which column is the primary key?

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



The JavaScript code is:

```
1  var prevID;
2
3  function parallelRun(v) { return (v==  idxID  ); }
4
5  function init()
6  {
7     var r=getCurrentRow();
8     if (r.isNull) return 0;
9
10    setOutputRowSize(0,r.nColumn);
11
12    var st=rowGetSortType(r,0);
13    if (((st!='0')&&(st!='9')&&(st!='A')&&(st!='Z'))||(rowGetSortColumnIdx(r,0)!=idxID))
14    {
15       throw ("You must sort the input table on the primary key column"); return 1;
16    }
17
18    prevID="__"+r.col(idxID);
19    return 0;
20 }
21
22 function run()
23 {
24    var r=getNextRow();
25    if (r.isNull) return 1;
26
27    var curID=r.col(idxID);
28    if (prevID!=curID)
29    {
30       prevID=curID;
31       r.write();
32       writeEOL();
33    }
34    return 0;
35 }
```

The idxID variable is a string that is the name of the primary key column inside the input table.

The idxID variable is a number that is the index of the primary key column inside the input table.

We check that the input table is sorted on the primary key.

When the key column "just changed", we output the row.

The algorithm of the "SimplifiedRemoveDuplicate Action" is very simple: We check that the input table is sorted on the key column. Because of the sort, all the rows with the same key-column are "grouped together" in contiguous rows. We read the (sorted) input table row-by-row. When we receive a row with a different key-column as the previous row (i.e. when the key column "just changed"), we output the row.

The above algorithm works inside a N-Way Multithreaded Section only if the partitioning variable of the N-Way section is equal to the "Primary Key" parameter of the "SimplifiedRemoveDuplicate Action" (i.e. is equal to the idxID variable). This check is performed inside the "*parallelRun(v)*" function. The content of the argument "*v*" to the "*parallelRun(v)*" function is the name of the partitioning variable of the N-Way section. The content of the variable "idxID" is slightly different inside the the "*parallelRun(v)*" function than inside the other functions (i.e. inside "*init()*" and "*run()*"): it contains the **name** of the selected column (instead of the **index** of the selected column).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

9.9.2. For R/Python based Actions

A given R or Python Action can work inside a N-Way Multithreaded Section only if the "Partition Type" is:

- **Each Partition has the same number of rows (with the exception of the last partition)**
- **Partition by Column**: When using this option, you must select a "Partitioning Column" that is equal to the partitioning variable of the N-Way section.

When the "Partition Type" is "**No partition**" (this is the default option), then the R/Python Action won't work inside a N-Way Multithreaded Section.

9.10. Debugging JavaScripts-based Actions

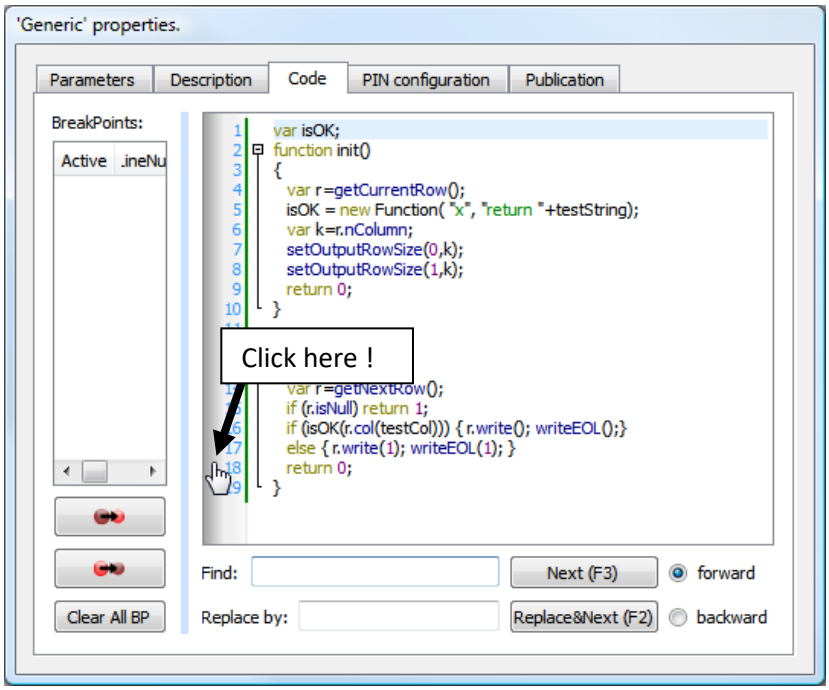
9.10.1. Starting the debugger

To start the Anatella-JavaScript debugger, you must do two things:

1. Define a break point into the javascript-source-code that you intend to debug (or you can also simply re-enable an old break point that has been defined previously).
2. Run your Anatella-Graph in "debug" mode.

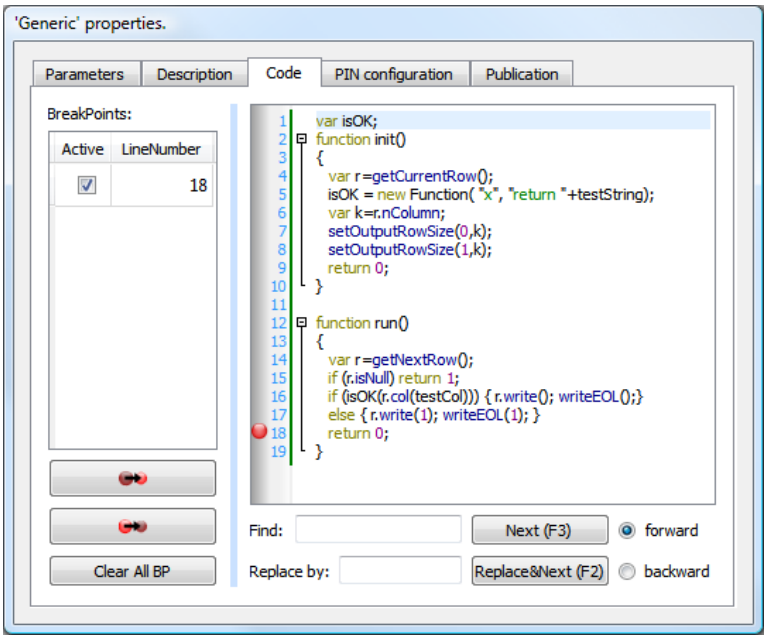
9.10.1.1. Managing break points

For example, if i want to put a breakpoint at the line 18 of my JavaScript, I simply click just next to the "18" number inside the code window:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

… and we obtain:



You can re-click on the same spot to remove the breakpoint.

Inside the breakpoint-list table (in the left panel), you can:

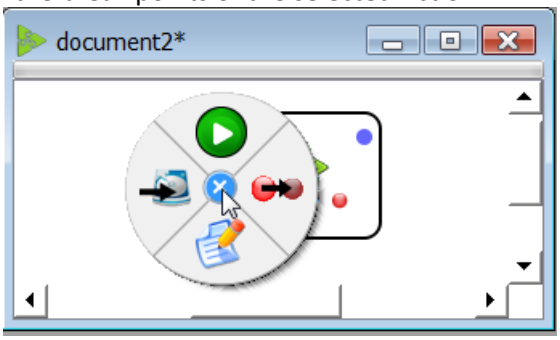
1. Click on the check box to disable/enable a breakpoint. If all the breakpoints are disabled, the debugger won't be used.
2. Click on the second column (containing the "Line number" of the breakpoint) to directly "move to" this line into the code window.

Using the button, you can enable all breakpoints.

Using the button, you can disable all breakpoints.

Using the button, you can clear (delete) all breakpoints.

To Enable/disable breakpoints, you can also use the circular-context-menu of your Action: Right-mouse-click on a script-based Action inside your Anatella-Graph and select the option: For example: this will disable all the break points of the selected Action:



The Anatella-debugger has one limitation: You cannot debug several different Actions at the same time. Only one Action "at-a-time" can have some **active** breakpoints. Only the action with the active breakpoints will appear into the debugger. You can easily select which Action will be debugged using

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

the above circular menu and the / option (to quickly "enable" the breakpoints of the Action that you want to debug and disable all the other ones).

9.10.1.2. Running an Anatella-Graph in "debug" mode

There are two "normal" ways of executing an Anatella-Graph (see section 4.4):

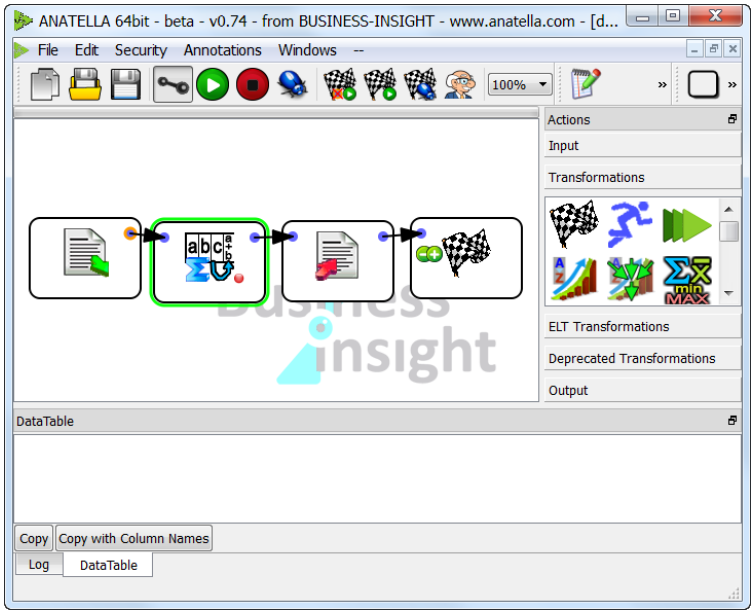
1. You can click on the button inside the main toolbar of Anatella: see section 4.4.1
2. When you are in "Run Mode" (i.e. when the button in the main toolbar is "checked"), each time you click on an output pin, Anatella runs the graph up to this point and creates the corresponding "HD Cache file": see sections 4.2.6, 4.4.2 and 4.5.2

There are two corresponding ways to run an Anatella-Graph in "debug" mode (to be able to use the Anatella-Script-debugger):

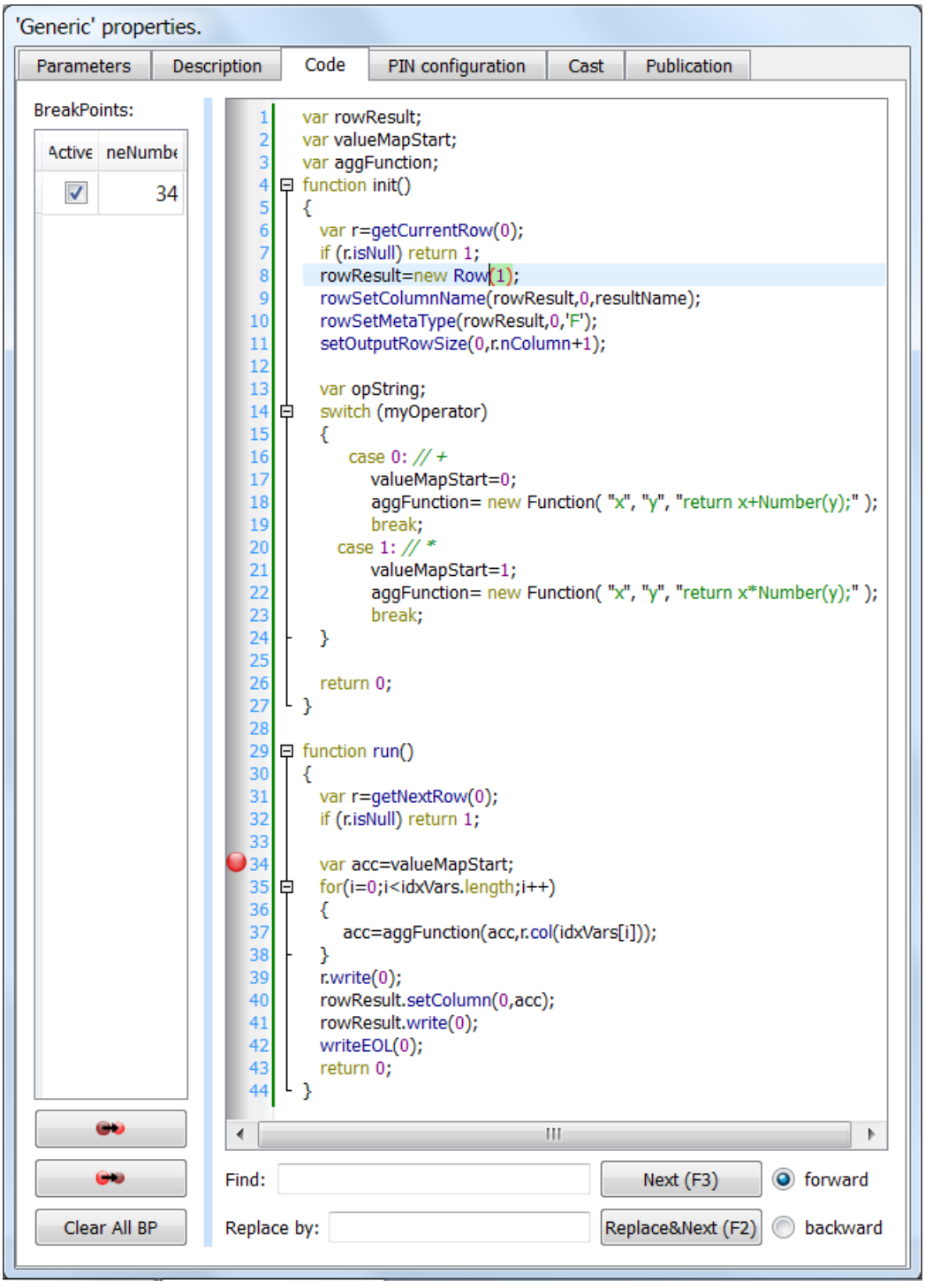
1. You can click on the button inside the main toolbar of Anatella.
2. Switch to "Run Mode" (i.e. check the button) and enable debug: check the button inside the main toolbar of Anatella. Now, each time you click on an output pin, Anatella runs the graph (with the debugger) up to this point (and Anatella also creates the corresponding "HD Cache file").

9.10.2. Using the debugger

Let's debug the following script:

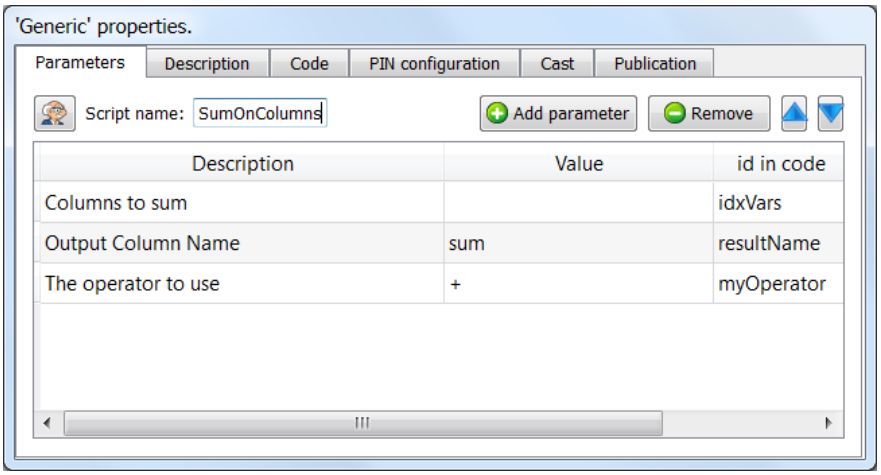


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The script parameters are:



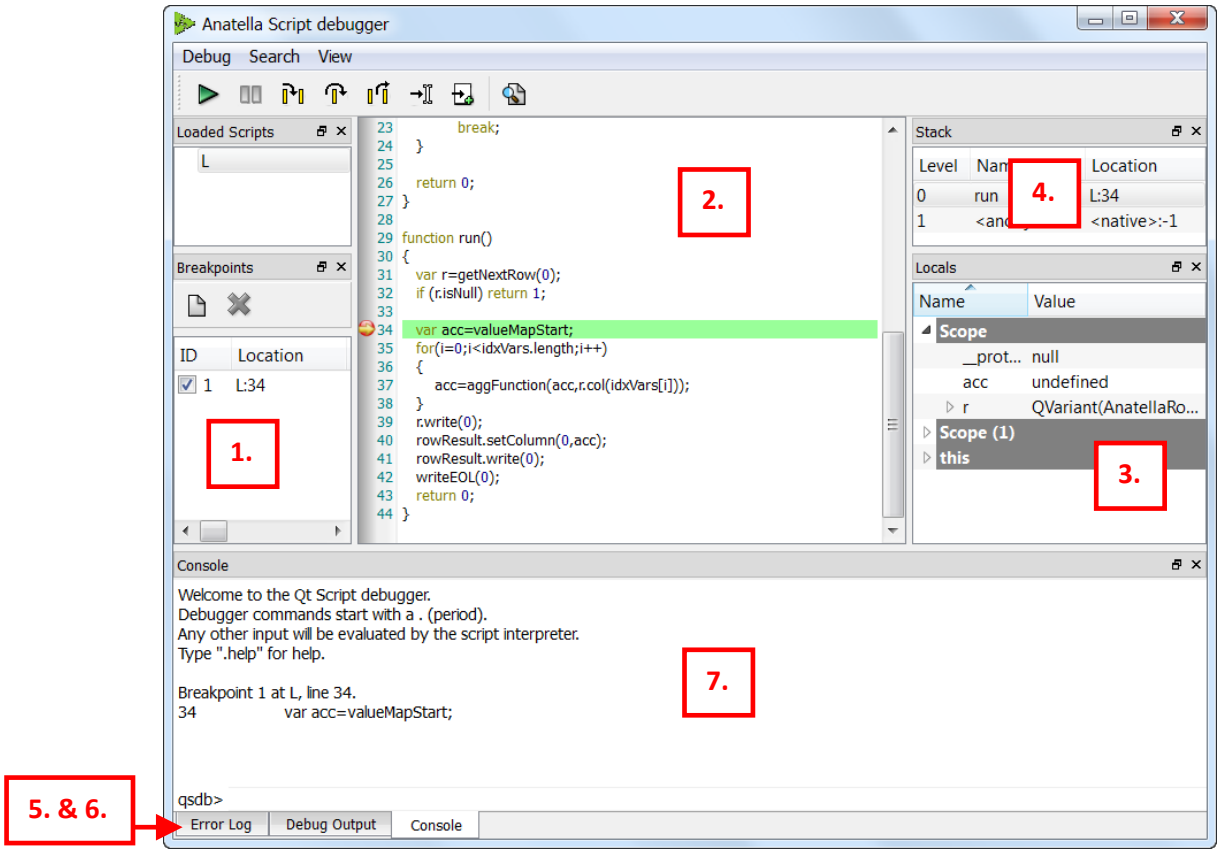
This script adds a new column into the output table. This new column is the sum of some user-selected-columns. To create new columns into an output table, we will simply write “row objects” (containing the additional columns) to the corresponding output pin. Inside this script we only add one column to the output table, so we only need to create one “row object” of size 1: see line 8. On line 11, we define the column name of the new column. On lines 39, 41 and 42, we are writing on the output pin 0: we write the “original” row and the “new” row (of size 1) (containing only our summation). The computation of the summation takes place on lines 34 to 38. The result of the summation is stored into a temporary JavaScript-variable named “acc” (short for “accumulator”). On line 40, this temporary result (stored in variable “acc”) is copied into the row object “rowResult”, so that it can be written on the output pin.

Note that we used on lines 18 and 22 the “*Number()*” function to obtain a numerical value from the string that is returned by the “*col()*” method of the row Object “r”. The same script without the call to the “*Number()*” function would have worked also but it would have delivered the concatenation of the strings inside the user-selected-columns (instead of the sum).

Click on the button (or Press F6)!

The Anatella-debugger starts:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



As requested, the execution stopped on row 34 of your script.
There are 7 different dock-windows available inside the main debugger window:

1. **Breakpoint list:**

   The breakpoints list shows all the breakpoints that are set. A breakpoint can be disabled or enabled by clicking the checkbox next to the breakpoint's ID (the ID is provided so that the breakpoint can be manipulated through the console widget as well).

   A condition can be associated with the breakpoint; the condition can be an arbitrary expression that should evaluate to true or false. The breakpoint will only be triggered when its location is reached **and** the condition evaluates to true.

   Similarly, if the breakpoint's ignore-count is set to N, the breakpoint will be ignored the next N times it is hit.
   A new breakpoint can be set by clicking the New Breakpoint button and typing in a location of the form "L:<linenumber>". The breakpoint location can refer to an already loaded script, or one that has not been loaded yet.

2. **Code window:**

   The code window shows the JavaScript code of the currently executed script. The widget displays an arrow in the left margin, marking the code line that is being executed. Clicking in the margin of a line will cause a breakpoint to be toggled at that line. A breakpoint has to be set on a line that contains an actual statement in order to be useful. When an uncaught script exception occurs, the offending line will be shown with a red background.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

The code window is read-only; it cannot currently be used to edit and (re)evaluate scripts.

3. **Locals: Maybe the most important window.**

   This window is extremely useful when debugging a transformation Script.
   I cannot live without it anymore.

   This window lists of all the defined variables (in the current stack frame). Only the variables that have been declared using the "***var***" keyword will be shown. For primitive variables (double, string, pure JavaScript objects), you can directly see the value of the variable. Objects can be expanded, so that their properties can be examined, recursively. Properties whose value has changed are shown in bold font.

   Properties that are not read-only can be edited. Double-click on the value and type in the new value; the value can be an arbitrary expression. The expression will be evaluated in the associated stack frame. While typing, you can press the TAB key to get possible completions for the expression.

4. **Stack:**

   This list all the function calls that have been made to arrive to the currently executed line.

   The stack widget shows a backtrace of the script execution state. Each row represents one frame in the stack. A row contains the frame index (0 being the inner-most frame), the name of the script function, and the location (line number). To select a particular stack frame to inspect, click on its row.

5. **Error log,**
6. **Debug Output:**

   These are less important windows. The "debug output" window is only used by the "*printRow(rowObject)*" command.

7. **Console: The second most important window!**

   The Console window allows you to interactively run parts of your scripts: You can call your own functions, change value of variables, make nearly any computations that you want, etc…

   The console widget provides a command-line interface to the debugger's functionality, and also serves as an interactive script interpreter. The set of commands and their syntax is inspired by GDB, the GNU Debugger. Commands and script variables are auto-completed through the TAB key.

   Any console command that causes a change in the debugger or debugger target's state will immediately be reflected in the other debugger components (e.g. breakpoints or local variables changed).

   The console provides a simple and powerful way of manipulating the script environment. For example, typing "x" and hitting enter will evaluate "x" in the current stack frame and display the result. Typing "x = 123" will assign the value 123 to the variable x in the current scope (or

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

create a global variable x if there isn't one -- scripts evaluated through the console can have arbitrary side effects, so be careful).

Furthermore, Anatella has 4 special script functions that are very useful inside the console (see next page for more information about these functions):

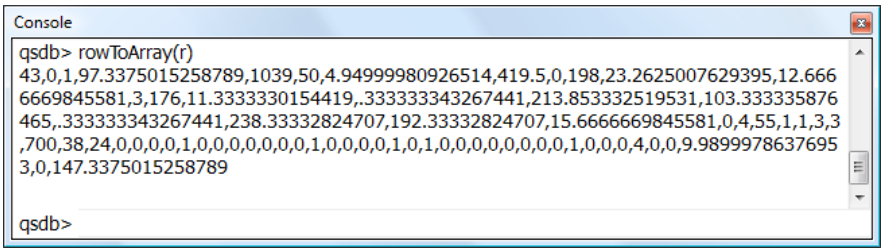
rowToArray(rowObject)
rowToObject(rowObject)
printRow (rowObject)
clearLog()

The script execution can be resumed in one of the following ways (These are standard commands and shortcuts also available inside the well-known Microsoft-Visual-Studio debugger):

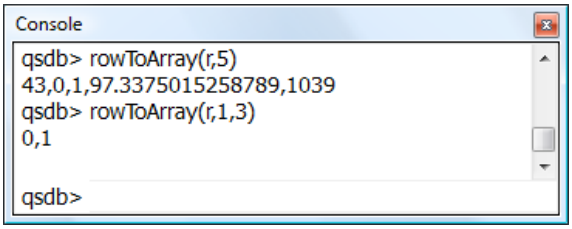
- **Continue (- F5)**: Evaluation will resume normally.
- **Step Into (- F11)**: Evaluation will resume until the next statement is reached.
- **Step Over (- F10)**: Evaluation will resume until the next statement is reached; but if the current statement is a function call, the debugger will treat it as a single statement.
- **Step Out (- Shift-F11)**: Evaluation will resume until the current function exits and the next statement is reached.
- **Run to Cursor (- Ctrl-F10)**: Run until the statement at the cursor is reached.

Press 3 times on the button (or press 3 times F10): you can see, in the "locals" window, the value of the "acc" variable updating as the Anatella-script is executed step-by-step.

In the "locals" window, you cannot directly see the content of the "r" object because behind this apparently simple JavaScript object, there is, in reality, a complex, highly optimized, C++ class. To see the content of the row object "r", write inside the Console "*rowToArray(r)*": you will see the whole row "r", where each column is separated by a comma:

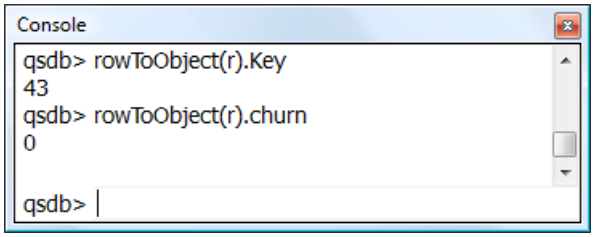


If you only want to see the first 5 columns of the row "r", you type "*rowToArray(r,5)*". If you only want to see the columns 1 (included) to 3 (excluded) of the row "r", you type "*rowToArray(r,1,3)*". For example:

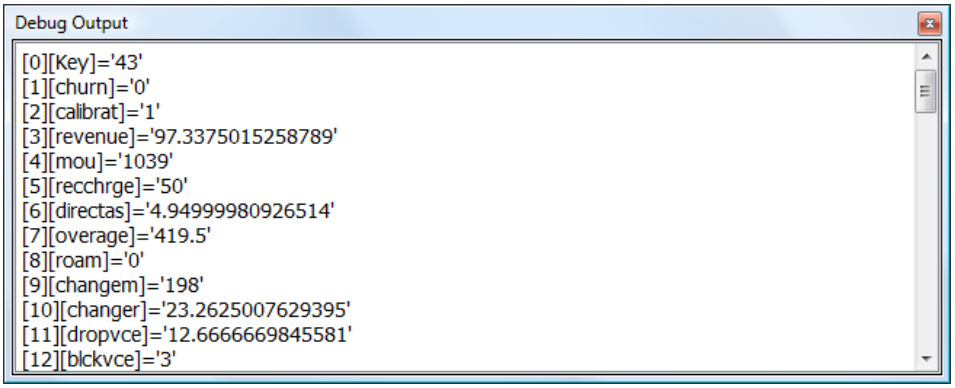


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

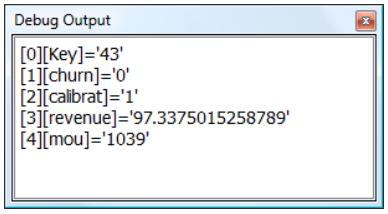
If you only want to see the column named “Key” of the row “r”, or the column named “churn” of the row “r”, you type in the Console window:



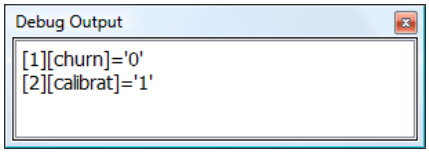
Finally, if you want to see a nice display of the row object inside the “debug output” window, you type in the Console window “*printRow(r)*” …and you obtain in the following “debug output” window:



Once again, additional parameters to the “*printRow(r)*” function allow you to select the exact portion of the row that you want to see. For example: “*printRow(r,5)*” shows:



Another example: “*printRow(r,1,3)*” shows:



9.10.3. Stopping the debugger

One Anatella feature that is really nice: When stopping the debugger, the position of your breakpoints is saved and the next time you use the debugger, you will retrieve your breakpoints exactly where you left them!

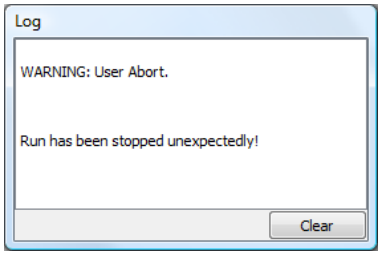
Usually, once you have detected your bug, you want to:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1. Stop the execution of the Anatella-Graph: press F4 and thereafter (possibly several times) F5.
2. Correct your bug (edit the JavaScript code)
3. Re-run the Anatella-Graph: press F5 to run your Anatella-Graph without the debugger or F6 to run with the debugger

To abort the execution of your graph, press the button (or Press F4). Unfortunately, the graph execution does not abort immediately: The Anatella Engine is only able to stop the graph execution when it's not executing any JavaScript code. You must thus exit the JavaScript code (and return to the C/C++ code) to allow Anatella to stop the graph execution. The simplest way to exit the JavaScript code execution is to press (possibly several times) F5: This will **Continue** the JavaScript code execution up-to-the-point that the execution control goes back to the C/C++ code: At that precise moment, Anatella is able to completely abort the graph execution.

Once the graph execution is completely stopped, you will see inside the Anatella log window the message "Run has been stopped unexpectedly!". For example:



Once you see this message, you can be sure that the graph execution is successfully aborted (but not before).

To summarize: To stop the debugger: press F4 and thereafter (possibly several times) F5.
Thereafter, to re-run your graph (in debug-mode), press F6.

10. FAQ

10.1. Help! I can't open my .anatella files anymore!?

There might be several reasons for that:

10.1.1. You need to update your Anatella software.

You receive an error message "*Cannot read file … check log window for more information.*" and the log window contains: "*The .anatella script 'xxx' has been created for Anatella v??? and you are using Anatella v???. Please update your Anatella software*".

The easiest solution is to update your Anatella software (as suggested).

If you are in a hurry, you can modify your .anatella file using an UTF-16 text editor (such as EMEditor, Textpad, EditPadLite, etc.) so that Anatella accepts to open it. Such manual modification might cause errors and erratic behavior inside Anatella, so try to avoid it. To remove the version-check:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

1. Open the file in an UTF-16 text editor.
2. Search for “version” attribute and replace it with 2.08:
   For example: Before modification:

```
<?xml version="1.0" encoding="utf-16"?>
<ANATELLA version='2.72>
<GlobalParameters …
```

After modification:

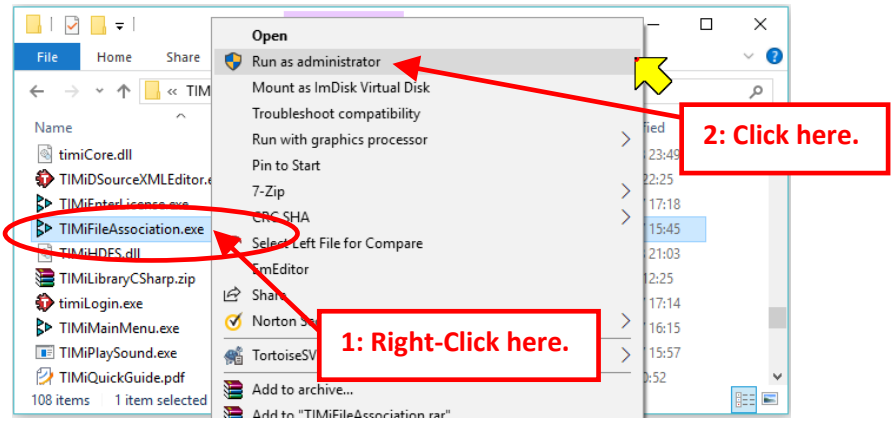
```
<?xml version="1.0" encoding="utf-16"?>
<ANATELLA version='2.08'>
<GlobalParameters …
```

3. Save your .anatella file. You can now open it inside Anatella.

WARNING: If your .anatella file is encrypted you cannot use the above “trick” (because the Text Editor will damage the binary code located after the file header) (But, you can still use a Hexadecimal Editor, if you really want to).

10.1.2. Broken .anatella file or Broken TIMi file association.

The MSWindows file association with .anatella files might be broken. To repair the file associations, run “TIMiFileAssociation.exe” (located inside the subdirectory “bin” inside your installation directory). Sometime, this is not enough and you need to run “TIMiFileAssociation.exe” with “elevated” privileges to restore the file association:



10.1.3. Encrypted .anatella file

You receive an error message “*Cannot read file … check log window for more information.*” and the log window contains: “*User-Right Error*” (optionally followed by: “*Your password has expired. Please re-login.*”.

This means that the .anatella file that you try to open is encrypted. You need to login before opening the file. See section 4.6.3. (about Security) for more information about this subject.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.2. Help! I am seeing NaN everywhere!

"*When I look at the DataTable view or inside my exported text file, I see 'NaN' in the columns where there should actually be numbers!*"

"NaN" means "*Not a Number*". It usually means that you tried to compute a sum, a mean, or any mathematical expression using some columns that did not contain any number in the very first place. All Numbers in Anatella are represented in "English Notation" (standard or scientific notation are also accepted): The decimal separator is the dot ('.') and there are no "thousands" separators. In particular, the "French notation" for numbers is not (directly) considered by Anatella as correct numbers and will lead to "NaN" results.

Here are some examples of Correct&Wrong number formatting:

GOOD	BAD
1528.7	1.528,7 (usage of "French" notation)
1.5287E3	1,528.7 (usage of a "thousand" separator)

You can use the ChangeDataType Action to "correct" the numbers that are in the wrong format.

More precisely, the ChangeDataType Action will not only "correct" the numbers, it will also change the meta-data of the column to reflect that the column contains floating-point values.

10.3. Help! I just made an error and I want to undo it!

Inside Anattella, the undo functionality is available in all text fields: Simply press CTRL-Z to undo (and CTRL-Y to redo).

If you made a mistake while editing the data-transformation-graph in itself, simply close the Window that contains your .anatella file (use the button) and re-open your .anatella file: See section 8.5. on how to quickly close and re-open Anatella graphs.

By default, Anatella saves your .anatella files each time you run them. It means that, if you did a mistake while editing your graph, you should close it (and therafter re-open it) **BEFORE running it**.

Also, you can ask Anatella to keep backup copies of your graphs. Using a previous backup of your .anatella file allows you to "go back" several steps "in the past". You can define the quantity of backups. A large number of backups allows you to keep track of all your modifications and editing over a long period of time: See section 7.1. about .anatella file backup.

Also, if you want to "go back" even more in the past, you can use a versioning system (such as Git) to keep track of all the versions of your .anatella files over the course of several years: See section 8.9. about versioning and collaborative work.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.4. Anatella is accessing the computer network. Is it normal?

The Anatella software in-itself does not access the network.

However, some ODBC drivers (for example: the MySQL ODBC drivers) and some OleDB drivers are accessing their database through a network connection. For these reasons, you should configure your firewall to allow Anatella to access your network (or at least, to allow access to the required database).

There are also some specific Anatella Actions that are accessing the network when used (This is a normal and expected behavior): Such Actions are:

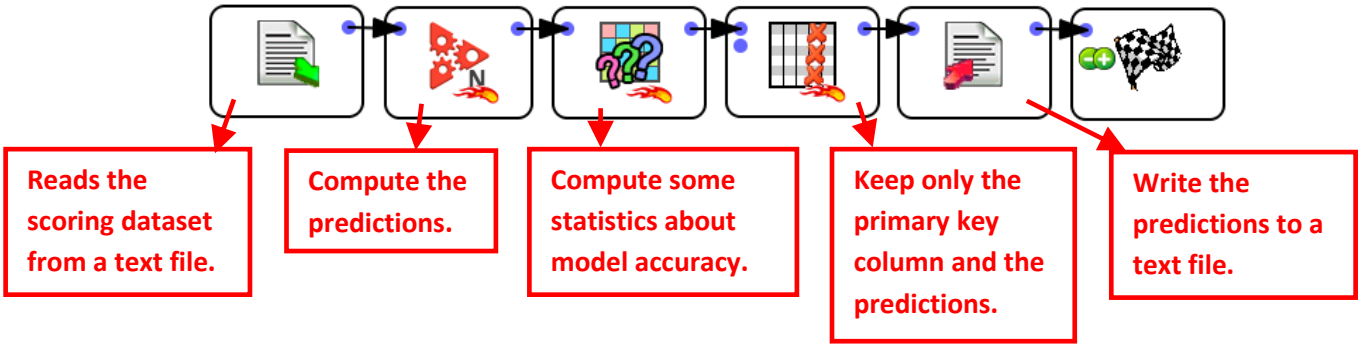
Query SMS Delivery Status, Geocoding & Reverse Geocoding, TCP-IP Receive table, Get Email, InfoOnIp, Send Email, TCP-IP Send table, Send SMS HTTP, Send SMS SMPP, Hadoop HDFS storage.

Anatella also uses cURL to run the following actions: These actions are accessing different Cloud services using the cURL:

The DownloadAndUpload action (see section 5.23.1), The Multiple DownloadAndUpload action (see section 5.23.2), The generic REST API call action (see section 5.23.3), The VAT check action (see section 5.23.4), The Tableau Publish action (see section 5.23.5), The S3ListFilesInBucket action (see section 5.23.6), The S3DownloadFile action (see section 5.23.8), The S3UploadFile action (see section 5.23.9), Send a Message to Slack

10.5. I can't find anymore the "Model Merger" module from TIMi?

It has been replaced with a combination of the TIMiUseModels Action (see section 5.7.1.) and the ConfusionMatrix Action (see section 5.7.2). The equivalent of the "Model Merger" module from TIMi is the following simple Anatella graph:

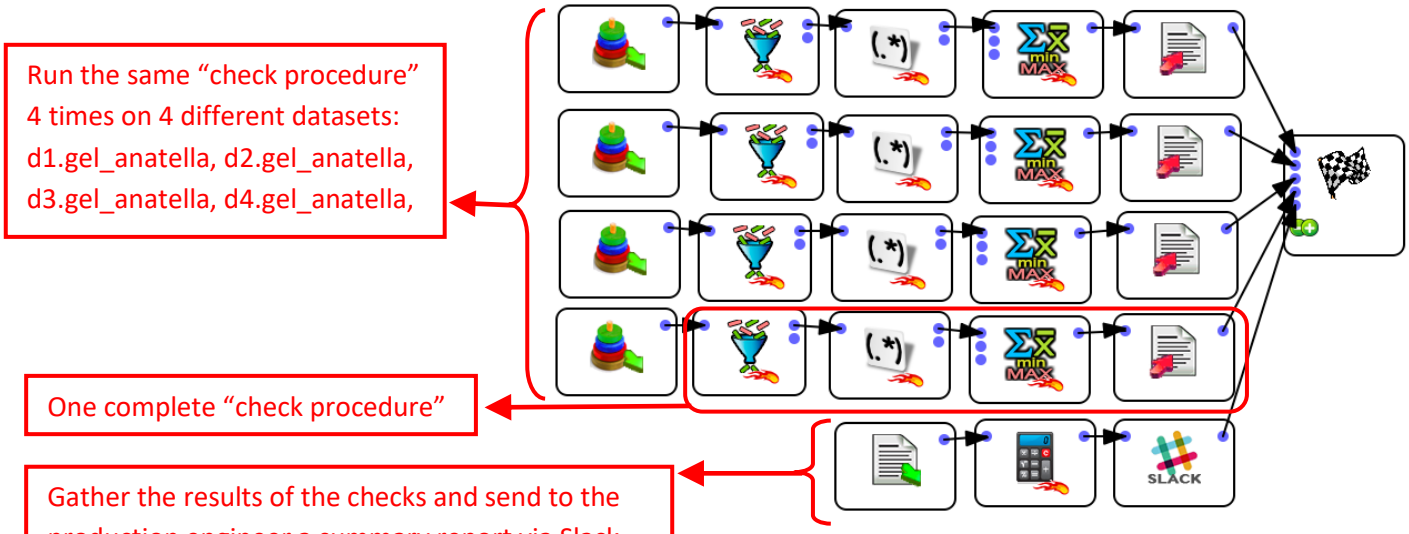


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

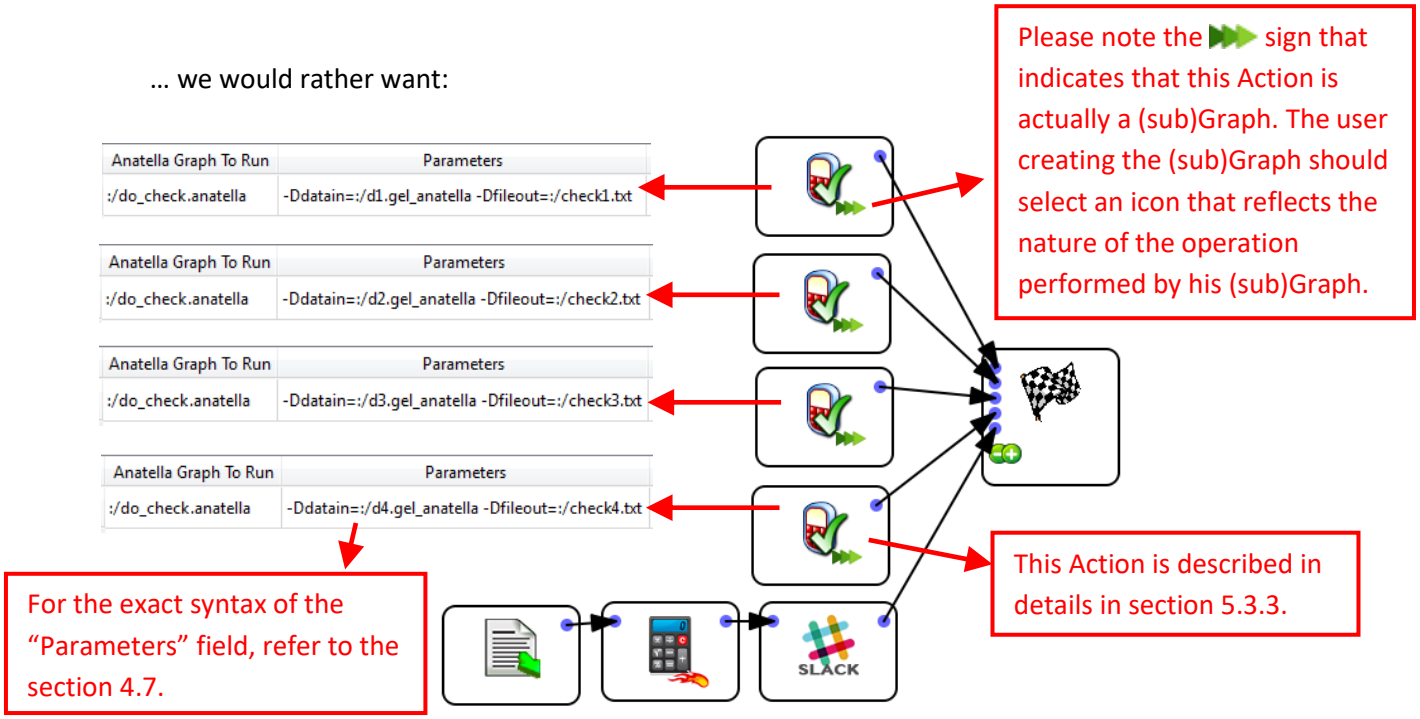
10.6. How to group several Actions into one unique Action?

Or, to rephrase the question: "*How to create a (sub) graph, parametrize the (sub)Graph and associate it with a new Action (i.e. with a new Icon)?*"

Let's take a concrete example: Let's assume that we have a serie of Actions that peforms various quality checks on your data. We want to isolate this serie of Actions inside a (sub)Graph and run the (sub)Graph on different datasets. So, instead of having:



… we would rather want:

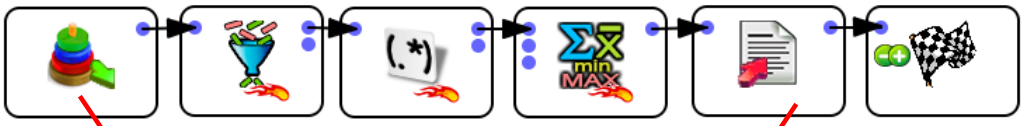


…where we isolated inside a graph named "do_check.anatella" the code (i.e. the Actions) that actually performs the check. We can see that the graph named "do_check.anatella" has two "Global Parameters":

- data*in*
- file*out*

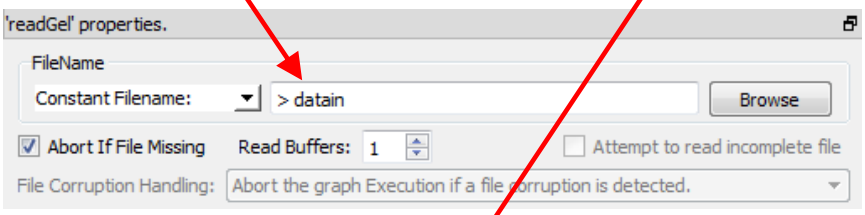
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Let's have a look at the graph named "do_check.anatella":

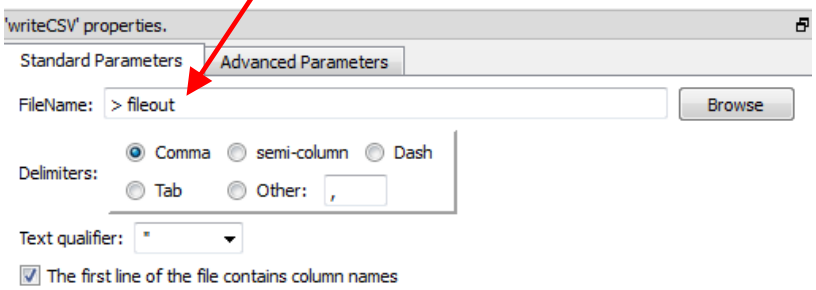


To create the above graph, we simply copied the original .anatella file and, thereafter, we edited the copy: i.e. we removed all the uninteresting Actions, to keep only the Actions that are performing the check.

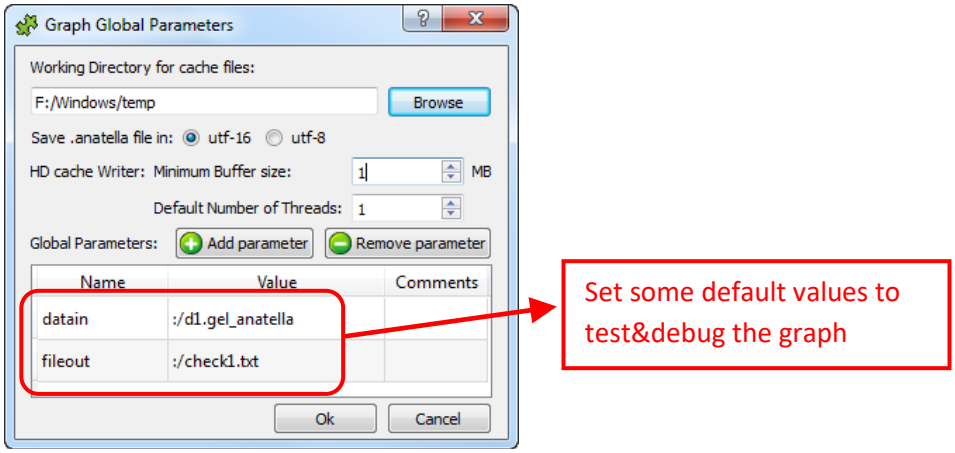
The name of the .gel_anatella file to check is defined using the "Global Parameter" named "datain":



The name of the text file that contains the outcome of the check is defined using the "Global Parameter" named "fileout":

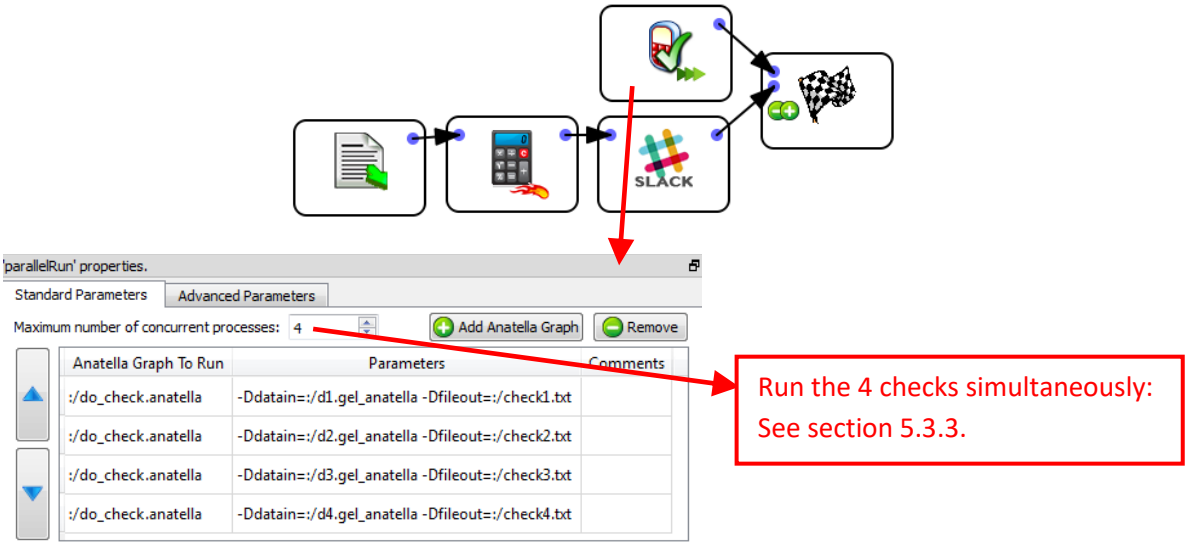


During the time required to develop&test the graph "do_check.anatella", we can assign to the two Global Parameters named "data*in*" and "file*out*" some default values. These values will be very useful for testing the graph:



We can now even go one step further: To get a higher execution speed, we could execute all the checks simultaneously (i.e. in parallel) in this way:

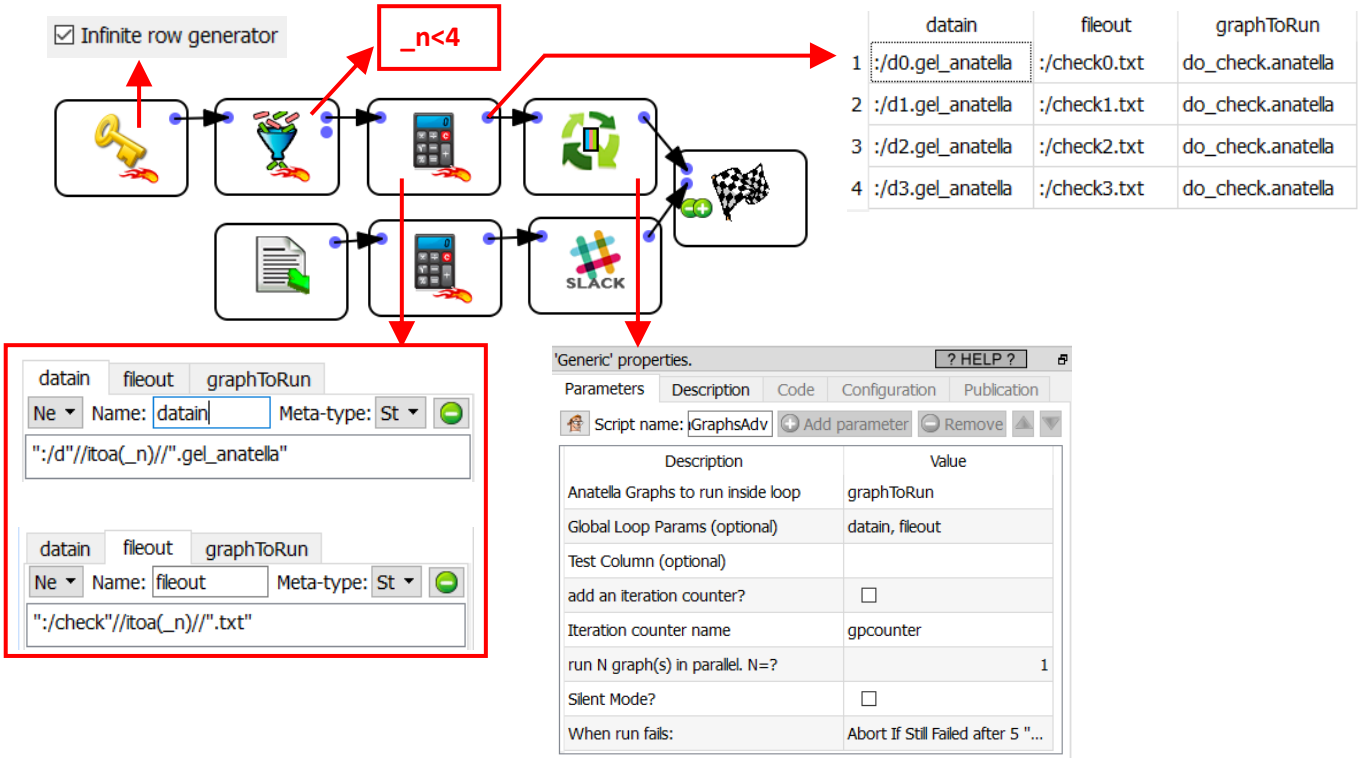
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



10.7. How to create "Loops" inside Anatella?

Let's re-start from the example given in the previous section (i.e. the section 10.6).

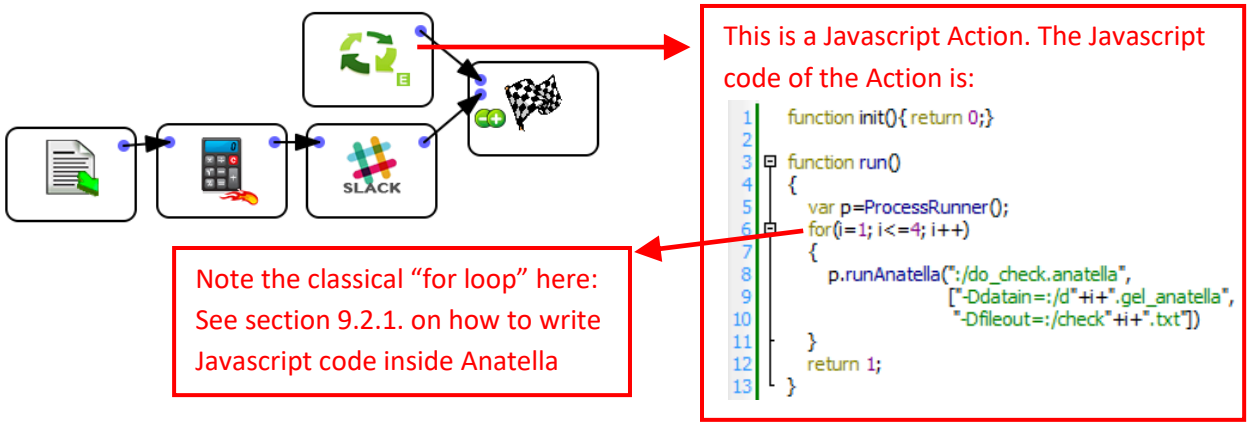
In the example from the previous section, we only had 4 datafiles to check. When we have many files to check, we can embed the call to the (sub)Graphs inside a loop. More precisely, we'll have:



For another example of usage of the loopAnatellaGraphAdv action (to create "loops"), see the section 5.21.6.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

If you prefer to code a little bit in Javascript, the following graph is equivalent to the above graph:



In the above example (with the Javascript "Loop"), the 4 "checks" (i.e. the four subGraphs) are executed in a sequential order. If we want to execute the 4 graphs simultaneously (to divide the computation time by 4), we'd rather have the following Javascript code:

```
function init(){ return 0;}

function run()
{
  var p=ProcessRunner();
  for(i=1; i<=4; i++)
  {
    p.addAnatellaJob(":/do_check.anatella",
                     ["-Ddatain=:/d"+i+".gel_anatella",
                      "-Dfileout=:/check"+i+".txt"])
  }
  p.run(4);
  return 1;
}
```

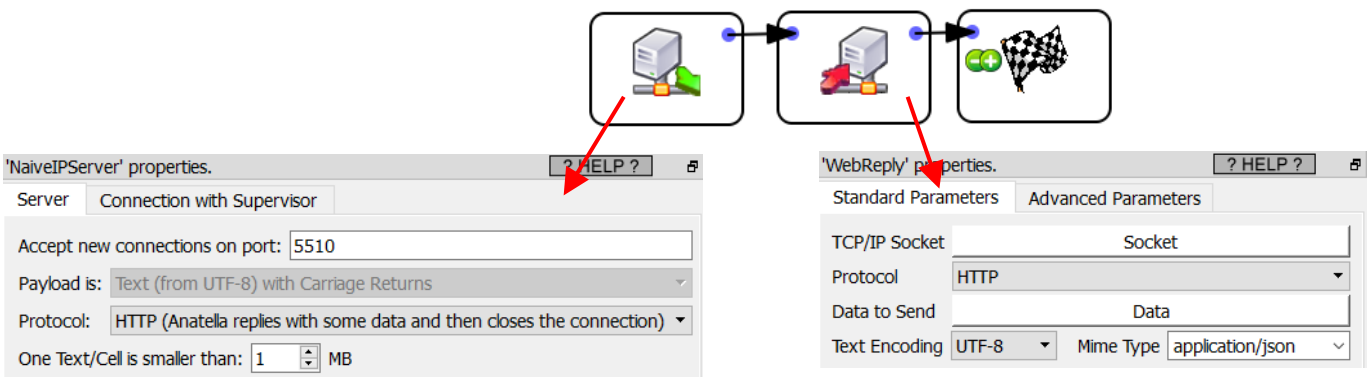
You'll find the documentation of the "ProcessRunner" Javascript Object inside the section 11.6.

10.8. Web Service, Real-time Data Streaming and Master Data Management

There are, basically, 3 technologies inside Anatella that allows to create a Web Service (for Master Data Management). These 3 technologies are described in the next 3 sections.

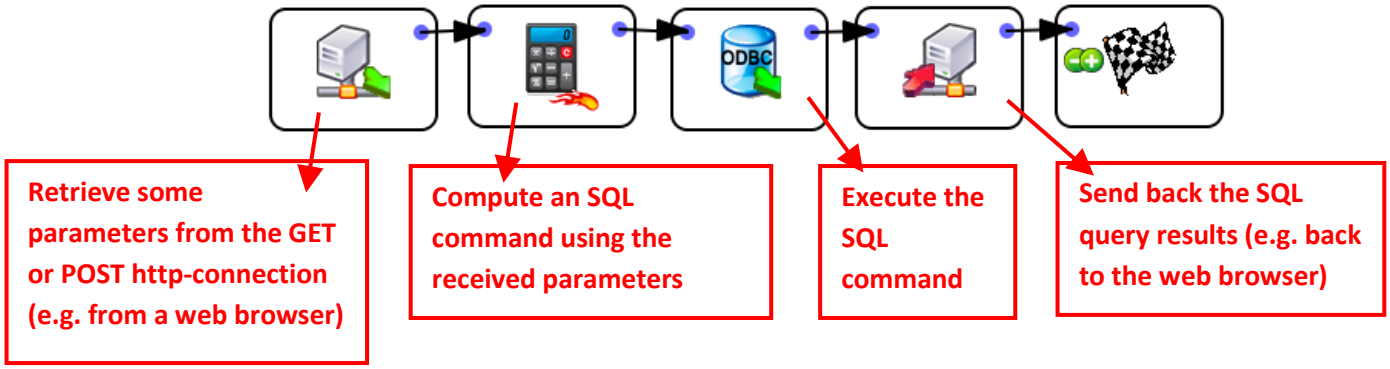
10.8.1. Streaming data processing using an HTTP source/sink

This is a very simple way to create a small webservice. For example, this Anatella graph creates a webservice that "echoes" the data received through a POST Http connection:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

This webservice queries a database and return the query results to the web browser:



As soon as a new TCP/IP connection is initiated, Anatella generate a new row of data (containing the received GET and/or POST parameters) and "forwards" this row through the graph. This means that you'll get a very quick response (in a few milliseconds) because the process to handle a new connection is is very "light weight" (i.e. it's just creating a new row and sending it it through the graph).

There are some limitations on the set of Actions that can be used inside an Anatella graph using a NaïveIPServer Action as the data source: See the section 10.8.4. for more details.

10.8.2. Streaming data processing using an IoT/MQTT source/sink

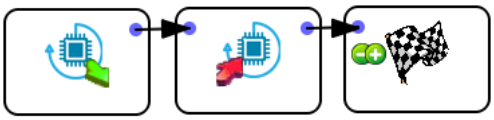
An Alternative to the NaïveIPServer action to stream data in real-time within Anatella is the MQTSubscribe action. MQTT is the most common IoT protocol and is 100% supported by Anatella. Here are the main differences between the MQTT protocol (supported by the MQTSubscribe action) and the HTTP protocol (supported by the NaïveIPServer action):

Topic	MQTT Protocol	HTTP Protocol	Comments
Infrastructure	**Complex**	**Simple**	When using MQTT, you need to manage and maintain a MQTT broker. HTTP is much simpler and does not require a third party software.
Resilience	**High**	**Medium**	MQTT supports the Level 3 **Q**uality **o**f **S**ervice (Level 3 QoS) that offers strong guarantees on messages delivery.
Protocol Overhead	**Tiny Headers**	**Big Headers**	The header size of the MQTT protocol is just from 2 to 4 bytes while the HTTP headers are much bigger (typically, from 150 to 250 bytes).
Easy to use	**Medium**	**High**	There now exists plenty of low-level libraries that allows to use the MQTT protocol but nothing beats in simplicity the simple usage of HTTP (e.g. using cURL or any browser to send HTTP messages is too easy!).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

From the point-of-view of Anatella, using the MQTT or HTPP protocol is equivalent: Everything that you can do with one protocol, you can do with the other:

For example, this Anatella graph creates a MQTT-based webservice that "echoes" the received data:



There are some limitations on the set of Actions that can be used inside an Anatella graph using a MQTSubscribe Action as the data source: See the section 10.8.4. for more details.

10.8.3. General Purpose Web Service

You can transform any Anatella graph into a REST-full web service using the procedure described in this section. This procedure involves a Web **PHP** server.

A web service is, typically, a small process that:
1. Waits for an incoming HTTP connection (typically initiated using your Browser or with cURL).
2. Extract some parameters from the HTTP connection. These parameters are typically stored in the URL part of the HTTP header (in the case of a GET http request) or stored in the data part of the HTTP request (in the case of a POST http request).
3. Compute some response based on the parameters received.
4. Send back the response through the same connection (i.e. the same socket) that was used initially to received the parameters. This response is usually formatted as a JSON structure.

You'll need a web PHP server to run the example given in this section. You can easily get such a web PHP sever by installing Kibella on your machine (since Kibella uses many PHP libraries).

We'll use a small example to explain how to create a REST-full web service using Anatella&PHP.
Let's now assume that:

1. We have a big .gel_anatella file that describes all the phone-calls given on a given day. This .gel_anatella file contains the following columns:
   a) The calling phone number: i.e. The column "A".
   b) The called phone number: i.e. The column "B".
   c) The duration of the call (`chargedduration`)
   d) The price of the call (`chargedamt`)

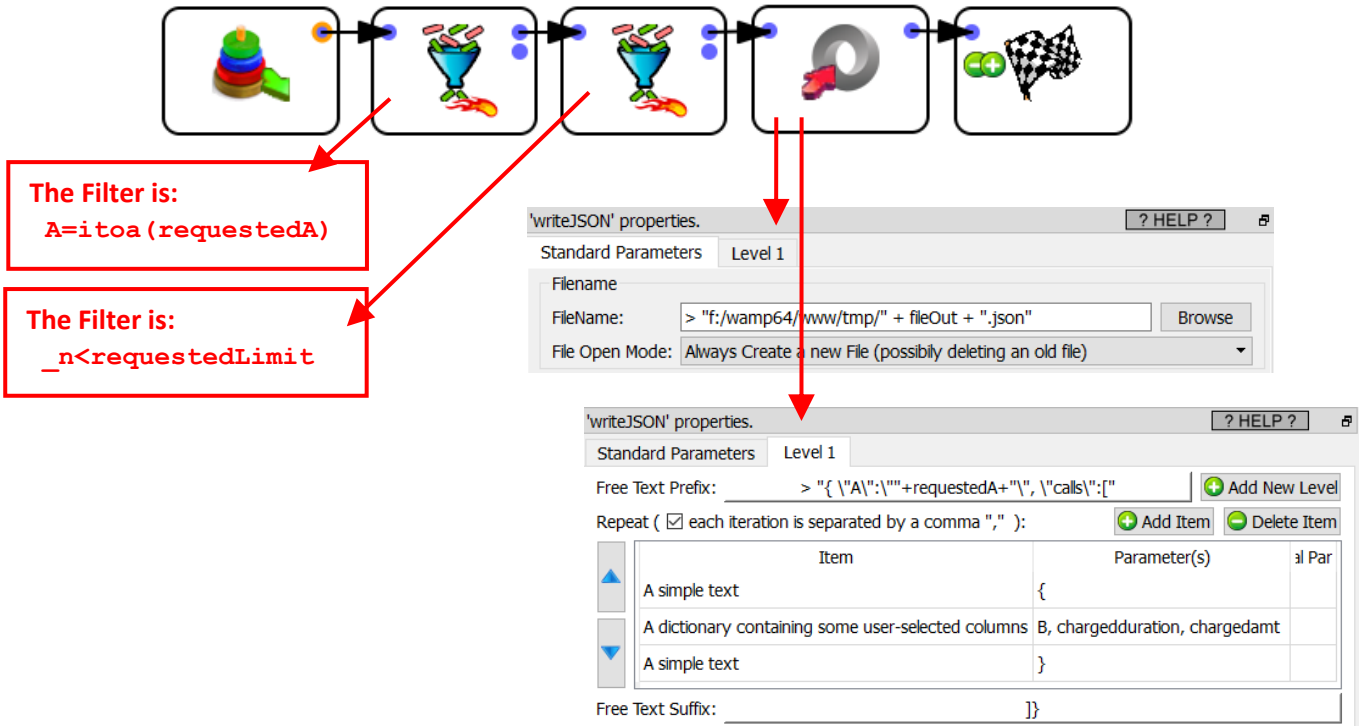
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. We want to create a REST-full web service that returns (inside a .json structure) (nearly) all the available data that we have about the phone-calls emitted by a user-given phone number (we'll name this user-given phone number as the “requestedA”). To limit the size of the returned .json structure, we'll limit the number of returned rows of information to a user-given upper limit (we'll name this user-given limit as the “requestedLimit”)..

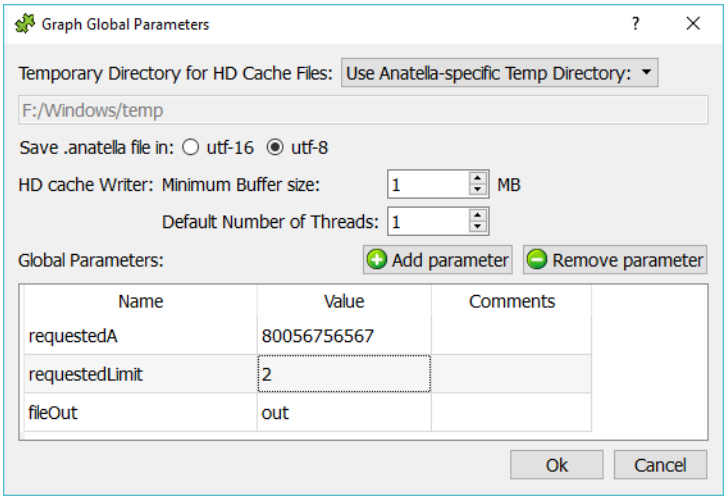
The first step to create the requested REST service is to create an .anatella graph that:

1. Uses as “global parameters” the 2 user-given parameters:
   a) the user-given phone-number named as “requestedA”.
   b) this user-given limit named as “requestedLimit”.
2. Uses a “global parameter” named “fileOut” that is the name of the JSON file that contains the results.
3. Extract the required rows of data and save them as a .json file.

Here is the requested .anatella graph:



The 3 “global parameters” used inside the above .anatella graph are:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Here is an example of JSON output file:

```
{
  "A": "80056756567",
  "calls":
  [
    {
      "B": "57876545456",
      "chargedduration": 45,
      "chargedamt": 3870
    },
    {
      "B": "343214556873",
      "chargedduration": 28,
      "chargedamt": 1926
    }
  ]
}
```

The second step to create the requested REST service is to create a PHP code that:

1. Extract from the GET or POST http data the two user-given parameters: “A” and “Limit”.
2. Run the .anatella graph (“p.anatella”) that computes the JSON result file.
3. Return the JSON result file to the webservice calling process (i.e. to the web browser or to cURL).

Here is the requested PHP code:

```
1  <?php
2  if (array_key_exists('A', $_REQUEST) == FALSE || array_key_exists('Limit', $_REQUEST) == FALSE) {
3         echo "<pre>The following variables were not found in the data received ".
4                                                          "from the browser: 'A', 'Limit'</pre>";
5         exit(-1);
6  }
7
8  $a = $_REQUEST['A'];
9  $nmax = $_REQUEST['Limit'];
10 date_default_timezone_set("UTC")
11 $id=uniqid("webservice_".date("Ymd_"));
12
13 exec('"c:/soft/TIMi/bin/AnatellaConsole" "f:/wamp64/www/p.anatella" '.
14                       escapeshellarg("-DrequestedA=$a")." -DrequestedLimit=$nmax -DfileOut=$id");
15
16 header('Content-type: application/json');
17 @readfile("f:/wamp64/www/tmp/$id.json");
18 ?>
```

Here are more details and explanations about the above PHP code:

- Rows 2-6: Check if the HTTP parameters “A” and “Limit” are present inside the HTTP request.
- Rows 8-9: Copy the HTTP parameters “A” and “Limit” into the PHP variables $a and $nmax
- Row 10-11: Compute the file name of the result JSON file.
- Rows 13-14: Run the Anatella graph named “p.anatella” with the “correct” values for the “global parameters” named “requestedA”, “requestedLimit” and “fileOut”.
  WARNING: The executable to run (i.e. AnatellaConsole.exe) must be written between double-qotes (and not between simple quotes!!) .
- Rows 16-17: Return the JSON result file to the webservice caller (i.e. to the web browser or to cURL).

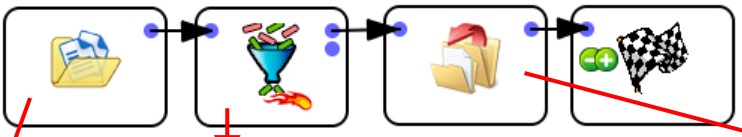
In the above PHP code, you can replace `$_REQUEST` with `$_POST` or `$_GET` to retrieve the values from the POST or GET http parameter exclusively.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We can call our webservice using the following cURL command-line:

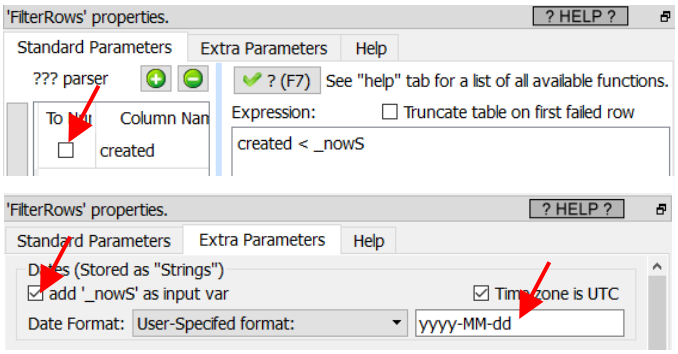
```
"c:\soft\TIMi\curl\curl" "http://localhost/testAnatella.php?A=80056756567&Limit=2"
```

At each call, our webservice creates one new temporary file (inside the directory "`f:/wamp64/www/tmp`"). To prevent the undesired accumulation of these temporary files, you should run a "cleaning" job once per day (i.e. schedule an Anatella job with Jenkins that runs daily). This "cleaning job" is the following:

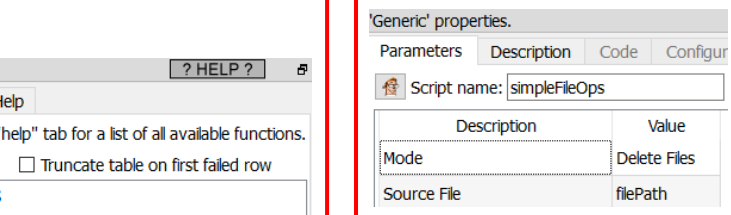


List all the webservice_*.json files located inside "f:/wamp64/www/tmp"

A filter that keeps all the "old" .json files (that are not from today).



Delete the old .json files:



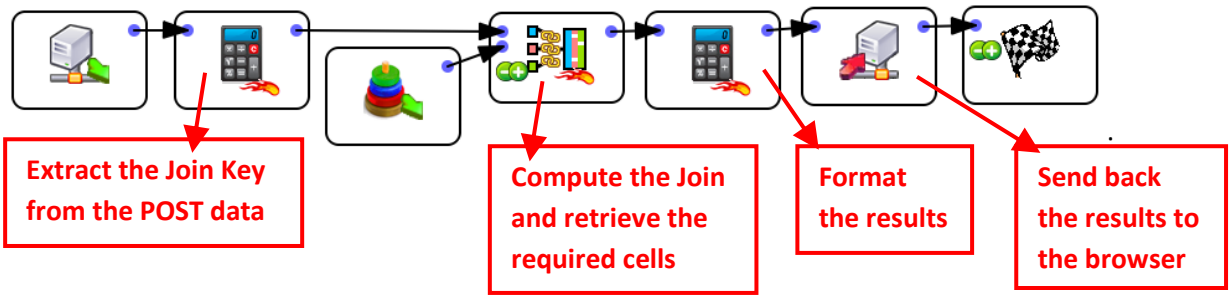
Here is a comparison with the solutions exposed in the 2 previous sections:

Subject	Real-Time Data Streaming (using the NaïveIPServer action or the MQTSubscribe action)	PHP-based solution (as exposed in this section)	Comments
Typical response Time	**A few milli-seconds**	**Half a second to several seconds**	The PHP-based solution must start a new (Anatella) process at each connection: This is much slower than simply creating a new Row in memory.
Infrastructure	**None** (at least for the NaïveIPServer action)	**Requires a web PHP server**	The web PHP server is not really a burden since it's really easy to administrate (and you most certainly already have it for other reason: e.g. Kibella)
Versatility	**Only a subset of all the Actions can be used** (See the next section 10.8.4. for more details)	**All the actions are available**	

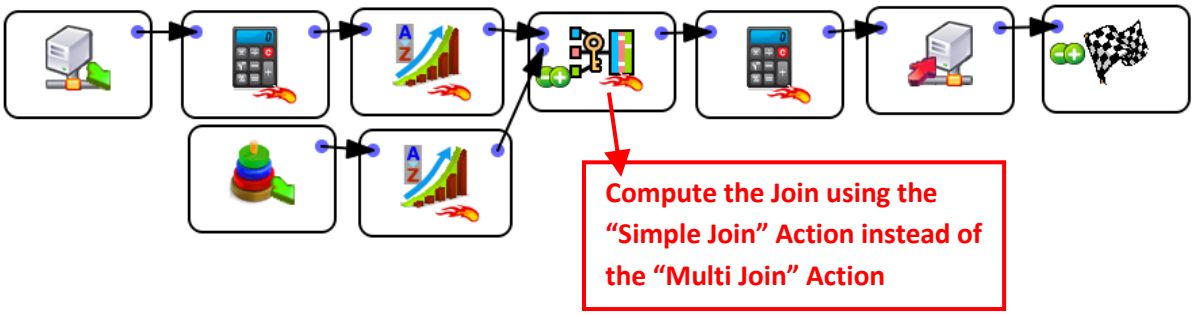
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.8.4. Limitations to the Real-Time Data Streaming in Anatella

The NaïveIPServer action and the MQTSubscribe action are streaming data in real-time within Anatella. At each new connection, Anatella generates a new data-row and "forwards" it through the data-transformation-graph: This is nice because it's a very fast & light weight procedure. However, this way of working has some limitations: For example, this is working:



…but this equivalent graph is not working (because of the "Sort" Action):



Why is this second graph not working? This is because the "Sort" action does not process the data row-by-row: i.e. the "Sort" action must accumulate all the rows from the input table before emitting its first output row. This "accumulation" never terminates (because the "NaïveIPServer" Action never stops and always continues to create new rows at each new connection) and the whole process "gets stuck".

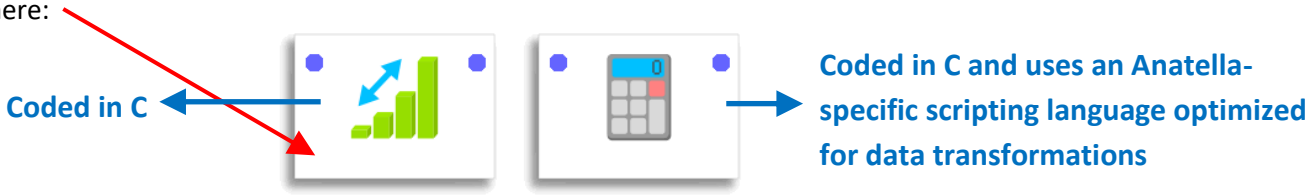
Inside a real-time data streaming Anatella graph, only the Actions that are working row-by-row will work as expected: Hopefully, this represents most of the commonly used Actions. To test if your graph works as expected, just add a few RowCounter Actions: At each new connection, you must see all the RowCounters that increase by one unit everywhere.

If you can't create a real-time Anatella graph (e.g. because you need to use an Action that is not allowed in real-time data streaming mode), you can still always use the solution exposed in the previous section (i.e. in the section 10.8.3.): This alternative solution is slightly slower but it allows you to use all the Anatella Actions without any restriction.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

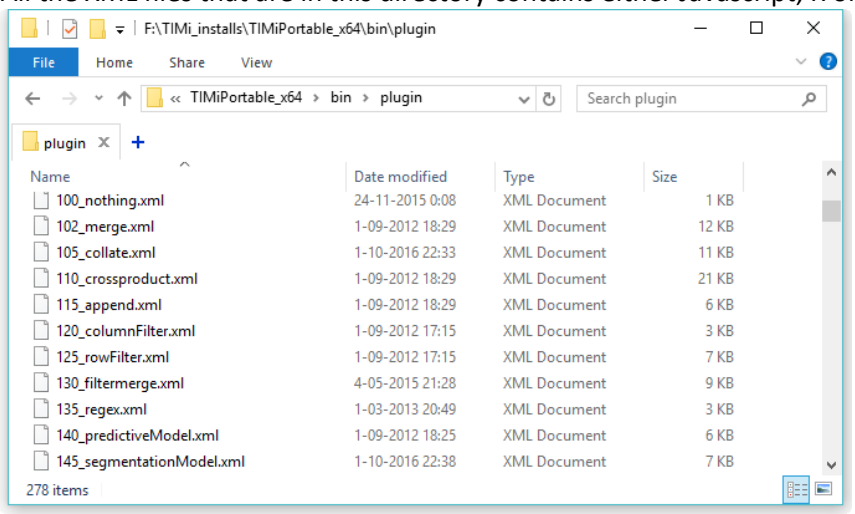
10.9. Where is the "*open source*" source code?

There are many Actions (i.e. the little "boxes") inside Anatella. Some Actions are coded in C/C++, other are coded in Javascript, R or Python. Only the Actions coded in Javascript, R or Python are open-source (and not all of them). The Actions that are created using the C/C++ language are closed source: You can recognize these Actions becayse they don't have any Javascript, R or Python logo here:

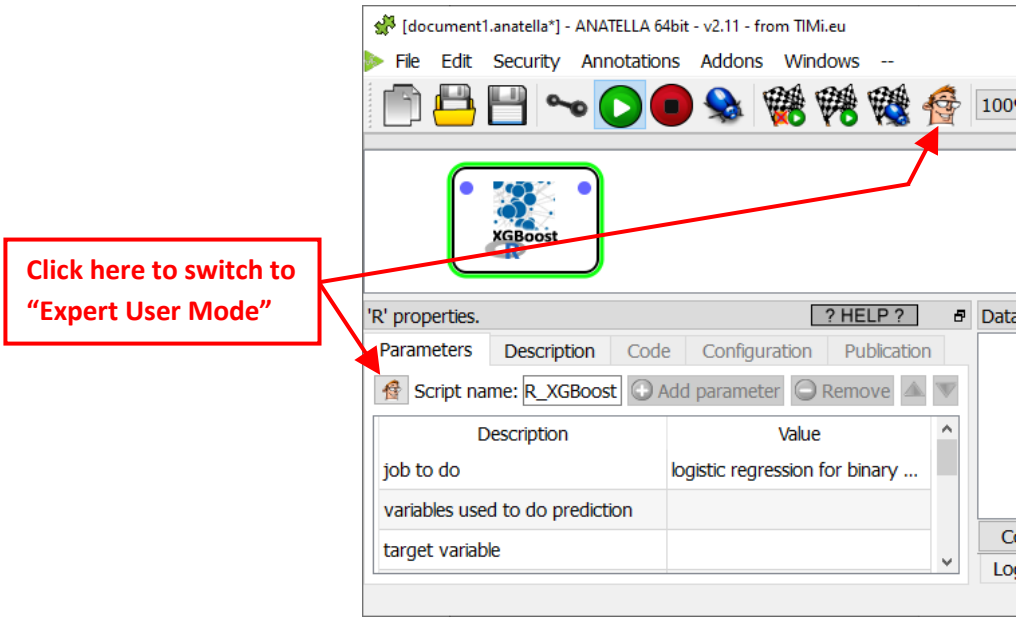


There are two ways to see the "open source" source code:

1. The first way is to open the "bin/plugin" subdirectory inside the Anatella installation directory: All the XML files that are in this directory contains either Javascript, R or Python code:

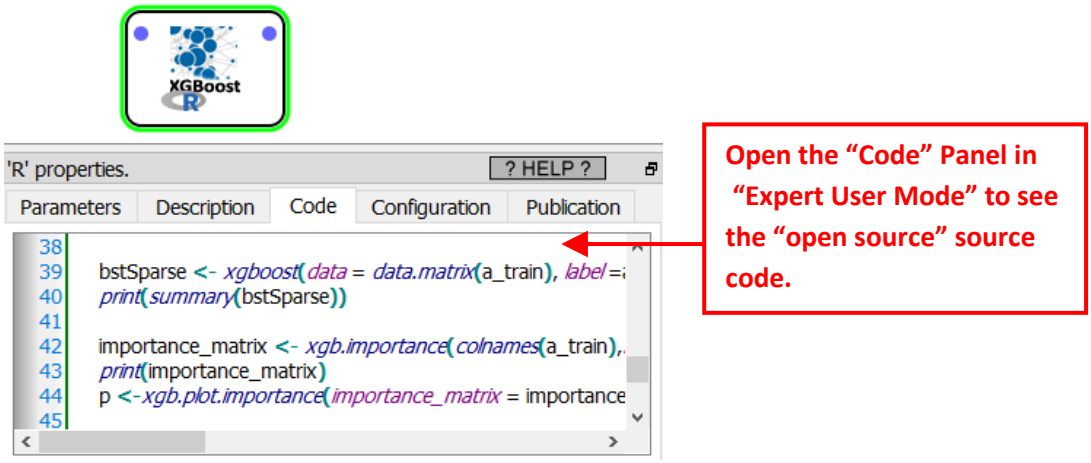


2. Let's assume that you want to see the source code of the Action named "R_XGBoost": Follow these steps:
   2.1. Place the Action "R_ XGBoost" inside the graph and open its property window.
   2.2. Switch to "Expert User Mode":



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2.3. The “open source” source code is visible inside the “Code” Panel here:



10.10. Working with Databases

10.10.1. Introduction: Use the right tool for the right job.

Common database engines (such as Oracle, Teradata, MS-SQL-Server, etc.) have been optimized to be very efficient and fast when you need to:

- Find a particular cell in a large table containing billions of cells.
- Update a few cells in a large table.

These optimizations are adapted to the typical workloads that a database needs to run: Here are some examples of workloads:

- You make a money transfer from your bank account to another bank account. This involves updating (at least) two cells:
  - The first cell contains your current “bank account balance”: i.e. you need to substract from the value stored in this cell the transferred money amount.
  - The second cell contains the destination “bank account balance”: i.e. you need to add the transferred amount to the value that is stored in this cell.
- You purchase a MP3 from a webstore. This also involves updating a very small number of cells/rows. Typically, we’ll have something like:
  - Update the cell that contains your money balance to substract the value of the purchased MP3.
  - Add one row inside the table that lists which MP3 belongs to which user.
  - …and some other small modifications to the webshop database.
- You want to get a list of all the money transfers executed over the last minutes (or seconds). This involves finding a few rows out of a table containing (possibly) billions of rows.

To find the “right” cells to display/process at a very high speed, the database engines are using a data-structure that is named “INDEX”. Typically, if you forgot to add an INDEX to a table inside a database, all the operations on this table will be extremely slow because the database engine will be forced to always do “*full table scans*” to find the cells/rows required to perform your commands. What’s a “*full table scan*”? A “*full table scan*” means that the database engine will be forced to read the whole table (i.e. all the rows) from top to bottom in order to find the few cells/rows required to perform your command(s). More precisely, the database engine might be forced to read hundreds of Terabytes out

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

of the hard drive, just to find the one or two cells that needs to be displayed/updated. This is terrible! In opposition, when an INDEX exists, the database uses the INDEX structure to directly find the location (e.g. the offset inside a file containing your table inside your hard drive) of the few cells that are required to perform your commands. Once this location is known, the database engine can then read and/or update the required cells at a very high speed. More precisely, the database engine will read or update a few hundred bytes inside your hard drive. This is much faster than a "*full table scan*" that involves reading Terabytes of data.

Basically, an INDEX allows you to respond extremely quickly to questions such as: "*What are the rows that contains the data from the customer with the ID="Bob"*?". ..and the answer will be: "*These are the row number 101,102,103*". Then, assuming that all the rows in the table have the same size (that we'll name <RowSize>), the database engine can quickly find the byte-offset of the row 101: The byte-offset is: 101 x <RowSize>. This means that, most of the time, to really benefit from the INDEXing mechanism, the database engine must use a constant "row size": i.e. All the rows must be stored using the same quantity of bytes. In this way, it's very easy to find the location/offset of a row inside a file, just knowing the row number.

The "constant row size" is annoying because it means that a column that is declared as a VARCHAR(20) (i.e. a column that contains a string of maximum 20 character long) will always use 20 bytes of storage (we assumed here the "Latin1" encoding, to have 1 byte=1 character), whatever the actual content of the cells. There can be some cells that contains shorter strings (e.g. strings that only have 2 characters) but we still use 20 bytes to store these "short" strings nevertheless (despite the fact that we could store a 2 characters string using only 2 bytes). **This obligation of "constant row size" thus produces larger files on the hard drive**. On the other hand, this "constant row size" is very handy when you need to update one cell: For example, Let's assume that:

- I have an "PostalAddress" table that have a column named "StreetName" that is declared as a VARCHAR(20).
- I just moved my home address: My "StreetName" changed from "Baker Street" to "Pennsylvania Avenue".

My new "StreetName" contains more characters (19>12) but, since I declared "StreetName" as a VARCHAR(20), I still have enough space left to "overwrite" my old "StreetName" with the new one (because 19<20). This means that the "update" operation will be very quick and straightforward (i.e. I just need to overwrite the old data with the new).

To summarize, a "classical" database engine:

- Uses INDEXes on your tables to quickly find the rows/cells required to perform your (SQL) commands.
- To really benefit from the INDEXing mechanism, the database engine must use a constant row size (i.e. all rows must use the same quantity of bytes for storage). A constant row size has some PRO's and CON's:

PRO	CON
○ Faster INDEXing ○ "Updates" operations are fast & straight forward	○ Larger files (filled with zero's). ○ No data compression algorithms (because, then, each row has a different size, due to the compression).

INSERTing new rows inside a table that has an INDEX structure is very slow. Indeed, each time you insert a new row inside a table, you must update the INDEX structure to reflect the presence of this new row: This is *very* slow! Please refer to the section 5.27.4.1. about the different "tricks" that you can use to avoid losing time when you need to insert rows inside a table that contains an INDEX (because always updating this INDEX structure after each insertion of one new row consumes a large amount of time).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

In opposition, the Anatella engine doesn't keep an INDEX structure on the hard drive (when required, Anatella can actually re-builds an INDEX structure in memory "on-the-fly": For example, this is what happens inside the MultiJoin Action or the "In-memory" Aggregate Action). This means that, most of the time, the Anatella engine will be forced to run "*full-table-scans*". …but the Anatella engine has been speed-optimized for this situation. When Anatella runs a "*full-table-scan*" on a table, Anatella need to extract from the Hard drive all the rows from the whole table. So, to go faster, we reduced to the minimum the quantity of bytes required to store the table (extracting less bytes from the hard drive means higher speed): i.e. Inside Anatella, the size of the .gel_anatella files or the .cgel_anatella files is reduced as much as possible: All the cells are as densely "packed" together as possible (in opposition to a database, inside .gel_anatella files, there are no "holes"). Anatella uses many different proprietary compression algorithms to reduce the data size. This means that, inside Anatella, it's not possible to update one cell in the middle of a table (if you want to do that, you need to re-create the whole table: i.e. you need to copy the complete .gel_anatella file with a one cell modified). An engine (such as Anatella) that do not have an INDEX structure might seem limited but this has several advantages:

- Since the Anatella engine has no INDEX structure, you can add new rows inside your tables (i.e. inside your .gel_anatella or .cgel_anatella files) at a very high speed (i.e. inside Anatella, on a small laptop, you can "insert" in a table several millions rows per second. This is, at least, hundreds times faster than the best databases and, very often, thousands times faster) because you don't need to maintain and update your INDEX structure at each "INSERT". It's thus possible to create very large tables very quickly.
- The overall storage space of your tables is much reduced: Typically, a table stored inside Anatella is from 20 to 100 times smaller than the same table stored inside a classical database because Anatella can "pack & store" the data as efficiently as possible (using advanced compression algorithms) and because Anatella does not lose space to store the INDEX structure.

The Anatella engine compensates the absence of INDEX by its raw reading (and writing) speed. Indeed, on a common-grade 2000€ laptop, the Anatella engine reads the .gel_anatella files at a speed of about 300 MB/sec (before decompression) or 1GB/sec (after decompressing the data contained inside the file). As a comparison, most databases read their data at a speed of only 10 to 20 MB/sec (because they are forced to read uncompressed data full of "holes"). This makes Anatella run from 50 to 100 times faster than a "classical" database when it comes to pure "reading speed".

So, despite the fact that the Anatella engine might read out of the hard drive a few more **un**necessary rows (because it doesn't have an INDEX allowing it to avoid reading these rows), it still able to reach very high processing speed because it reads these rows at a very high speed.

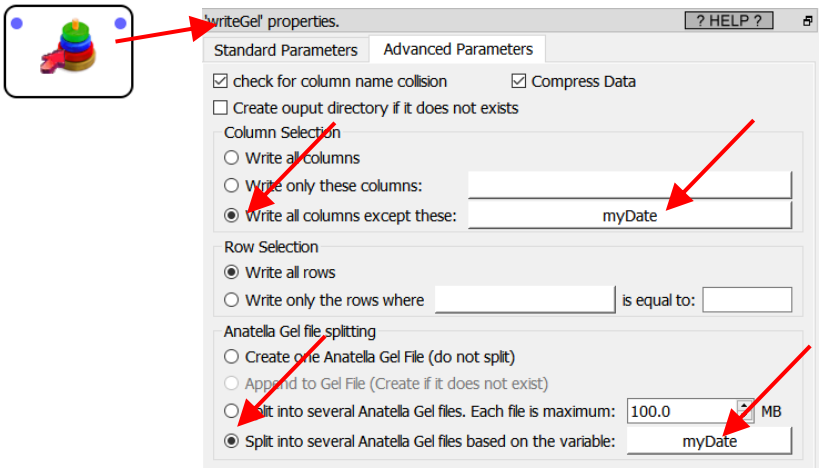
This means that Anatella is especially good for jobs that involve reading complete tables (or a very high quantity of row, such as over 90% of the rows of the tables): i.e. Anatella is especially good for jobs that involve "*full-table-scans*" because these jobs will, typically, run at a speed that is 100 times faster inside Anatella than inside a database.

Since the Anatella engine do not have any INDEXing mechanism, we'll have to use some "tricks" to avoid reading large tables completely from start to bottom. One very common "trick" is to **split** our table into many different .gel_anatella files (or .cgel_anatella files). And, thereafter, when you access your table, you extract out of the hard drive only the required files (and we don't extract/read *all*

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

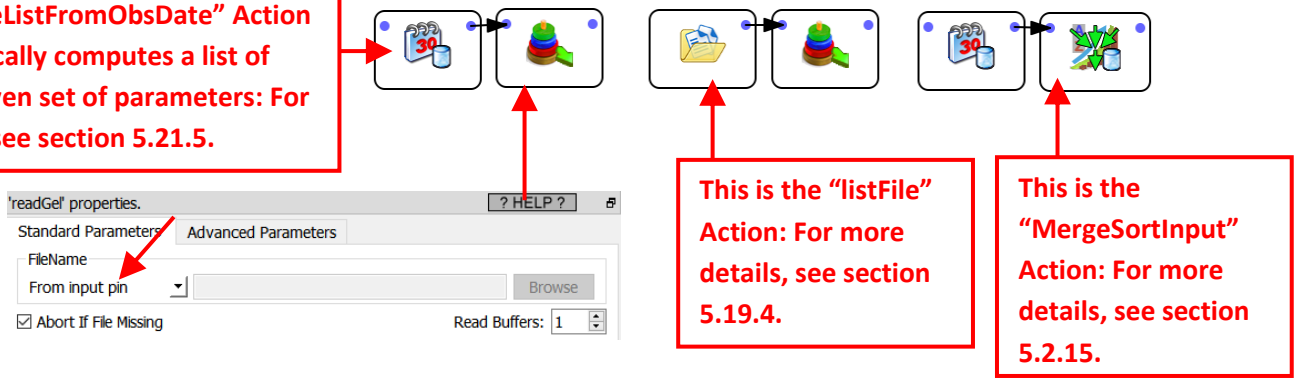
the small files that are composing the table, otherwise there is no speed gain). In practice, this is done this way:

- **TIME-based Splitting:**
  We'll assume that each different .gel_anatella file must store the data for a specific day (i.e. we split "by day"). We have inside our original dataset a column named "myDate" that contains the exact day each data-row is linked to. When writing your table on the hard drive, we'll use the "SPLIT" option of the "writeGel" action in this way:



When reading-back the table from the hard drive, Anatella must "assemble" all the different .gel_anatella (one file per day) "as if" there is only one .gel_anatella file (but, in reality, Anatella is concatenating the content of several .gel_anatella). To do so, you can use:

This is the "fileListFromObsDate" Action that automatically computes a list of files from a given set of parameters: For more details, see section 5.21.5.



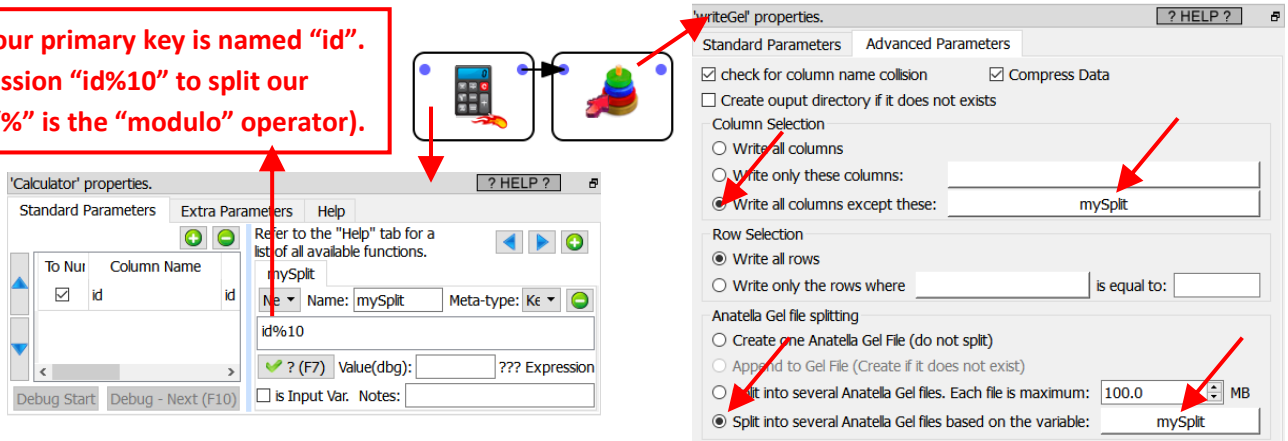
This is the "listFile" Action: For more details, see section 5.19.4.

This is the "MergeSortInput" Action: For more details, see section 5.2.15.

See also the section 5.21.6. for a complete example of "TIME-based Splitting" and databases interaction.

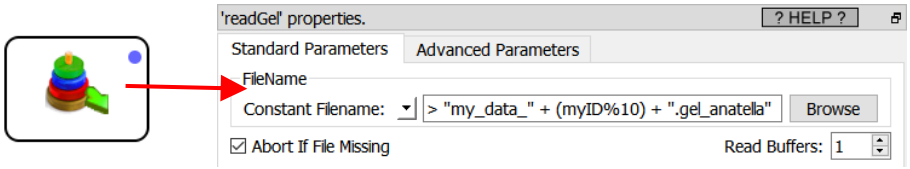
- **Key-based splitting**
  We'll split our table on a primary key inside your table. This is done this way:

We assume that our primary key is named "id". We use the expression "id%10" to split our table in 10 files ("%" is the "modulo" operator).



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Thereafter, to read the .gel_anatella file that contains the data for a specific "myID" ("myID" is a "graph global parameter": see section 5.1.5. about "graph global parameters"), we'll have something like:



As you can guess, the Anatella engine is very good when you have to do "*full-table-scans*". "*full-table-scans*" happens all the time when you do these tasks:

- **"Business-Intelligence" Tasks or "Strategic Queries":**
  For example: What's the sum of the revenue for the last 3 months?

  We'll use the "trick" explained hereabove to split the Revenue Table "on time" and only extract/read from the hard drive the .gel_anatella files that are required to compute the sum over the last 3 months. This means that we won't extract from the hard drive any unnecessary rows (because we only read the .gel_anatella files that matches the requested 3 months and nothing more): i.e. we run a high-speed "*full-table-scan*" on these files.

  To execute "Busines-Intelligence" tasks, the best solution is almost always to run a "*full-table-scan*". This means that Anatella is technically the tool that will deliver the highest performances for this type of tasks. In opposition, databases will have comparatively poorer performances (again for this type of "Busines-Intelligence" tasks). This means that Anatella is very often used in conjunction with visualization tools for "Business-Intelligence" (because it's the fastest&cheapest tool for such tasks). More precisely, Anatella is used to run the heavy aggregations, the heavy "joins", etc. Once the data size is reduced (i.e. aggregated and cleaned by Anatella) to a smaller, easily manageable size, then you can use practically any vizualisation tool to plot your data. For your convenience, Anatella can directly inject datasets into the most popular data visualization/BI tools:
    - Anatella directly creates .hyper files for Tableau (see section 5.27.16)
    - Anatella directly creates (and reads back) .qvx/qvd files for Qlik (see section 5.27.17)
    - Anatella directly creates .sqlite files for Kibella (see section 5.27.18)
    - Anatella directly creates .json files for Kibana (see section 5.27.13)

  If your vizualization tool is not listed hereabove, you'll need to copy your "reduced" datasets inside a classical database for "inter-operability" reasons (because all vizualization tools can access data stored inside a database). See the next section 10.10.2. to know how to make this "copy" procedure as fast as possible. If you are lucky enough to have your vizualization tool listed hereabove, I strongly suggest you to avoid "*going through a database*" to exchange data with your vizualization tool because the native Anatella connectors are much, much faster (from 10 to 100 times faster).

- **"Analytical" Tasks**
  Very often, the first step of any "Analytical" job is just to create a "Customer View" (also sometime named "CAR : Customer Analytical Record").

A "customer view" is a table where each row contain data about a different customer and the columns contains as much as possible different informations about your customers.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Amongst other things, the "Customer View" is used to create optimized marketing campaigns (through predictive analytical and machine learning) and to run many different types of analytics inside a "customer-centric" organization. It's also used extensively by your CRM system.

Typically, inside a "customer view", you'll have a column such as "*Number of Puchase over the last Month for this Customer*". To compute this aggregate, you'll need to read the (very large) "Transaction Table". And, once again, this translates in technical term to a "*full-table scan*" (once we used the "trick" explained hereabove to split the Transaction Table "*on time*"). Actually, to compute all the columns inside the "customer view", you'll need to run *many* "*full-table scan*". This means that, once again, Anatella is the best tool because it's technically the tool that will deliver the highest performances because it's optimized to run "*full-table scans*".

Once the "customer view" table is computed, you need to copy it inside your data base, so that other tools can use it (e.g. your CRM system might use your "customer view"). See the next section 10.10.2. to know how to make this "copy" procedure as fast as possible.

- **Machine learning, Text Mining, Graph Mining tasks:**
  These tasks involve creating large tables (that are named in technical terms "Learning datasets", "Scoring datasets", etc.). The complete computation of some new tables always involves a "*full-table scans*" and Anatella is, again, the best technical solution.

  The last step of a Machine Learning, Text Mining, Graph Mining, etc. tasks is nearly always to save the new results inside data base (for inter-operability reasons: So that other tools can use these results). See the next section 10.10.2. to know how to make this "copy" procedure as fast as possible.

10.10.2. Upload a fresh copy of a whole table

It's very common to need to upload inside the database an "updated" copy of some result table.

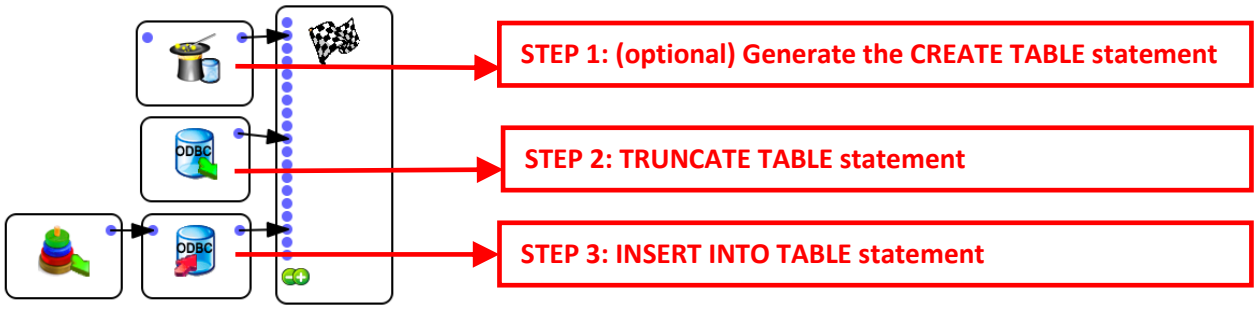
The first question is the following: Are you using a database as an "inter-operability" platform to quickly exchange data with a CRM/Visualization tool that has its own storage system? If that's the case, I'd suggest you to use a SQLite database. Indeed, the "INSERT" speed (which is usually the only bottleneck when working with a database) inside SQLite databases is from 10 to 1000 times higher than inside a "normal" database. To create/update a SQLite database, use the SQLiteWriter Action: See the section 5.27.6 (and 5.2.4.) for more details.

The typical steps to "*upload a fresh copy of a table*" are:

1. Optionally: Create the table using a "CREATE TABLE" statement.
2. Truncate the table (to remove any "old" rows).
3. (If required: Drop all indexes)
4. Insert the new rows inside the table.
5. (If required: Re-create the indexes)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We'll have:



In such very simple scenario, you should:

- …avoid using any "UPDATE" statements (because databases are usually extremely slow when processing "UPDATE" statements): i.e. it's faster to use "TRUNCATE" and then "INSERT".
- …avoid any INDEX on your table (because databases are much slower when inserting rows inside an INDEXed table).

You can use the ODBCReader Action to execute any SQL statement (here above: a "TRUNCATE" statement)

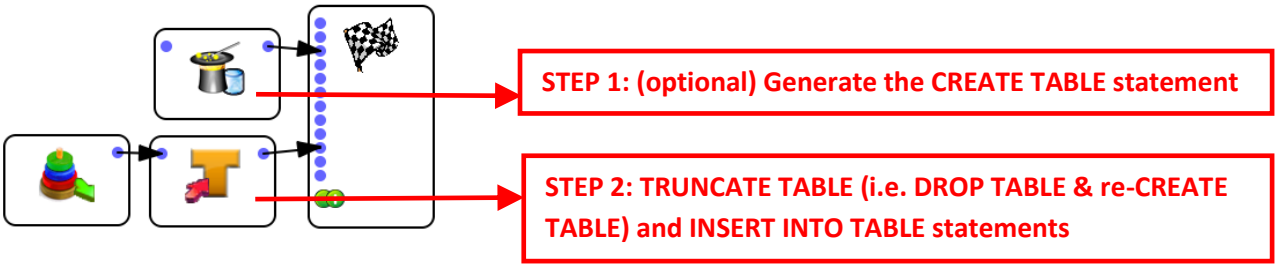
See section 5.2.2. about the ODBCReader Action.

See section 5.27.3. about the Upsert Action.

See section 5.27.4. about the CreateTable Action.

See section 5.27.19. about the Teradata Writer Action.

If we are using Teradata, we'll have:



To get an even higher "INSERT" speed, instead of using the Upsert Action, you might be tempted to use a "bulk upload" tool. Indeed, many database vendors offer specialized tools called "bulk upload" that allow you to copy a text file into the database at a very high speed: See the next section for some comments about these "bulk upload" tools.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.10.3. About Bulk Upload Tools

Typically, the “INSERT” operations are very slow inside a database and incurs a high load on the database system. So, most database vendors offer specialized tools called "bulk upload" that allow you to do many “INSERT” at very high speed and at a lower processing cost. Typically, these "bulk upload" tools take as input large **text** files and copy them into the database. There exists two different problems when working with text files:

1. Text file cannot store the “NULL” value (that is represented inside the Anatella-data-preview-window by an empty cell with a *red* background): Text files can only store strings of zero-length (that is represented inside the Anatella-data-preview-window by an empty cell with a *white* background). Thus, you might lose some valuable information here (because, for example, from the point-of-view of a predictive model, the NULL value might represent a concept fundamentally different than the “” value).

In opposition to classical “Bulk Upload” tools, the Upsert Action and the TeradataWriter Action are both able to safely send to the database NULL values.

2. Storing floating-point numbers inside a text file is the most common source of many large “rounding errors”: i.e. You’ll get a slight loss of precision or accuracy when storing floating-point numbers inside text files because of the decimal↔binary conversion: See the next paragraph for more details on this subject. When these "rounding errors" accumulate, they can represent a large quantity of Euros/Dollars (There was even a movie about this subject: i.e. A coder that recovered the money losts inside the "rounding errors" (several millions euros) and put them on his own bank account)

Let’s now talk about how the floating-point numbers are stored inside a computer. Typically, the floating-point numbers are stored & manipulated in **binary** notation: 010011101 (more details about this subject here and here: IEEE 754-1985). Unfortunately, in text files, these same numbers are stored in **decimal** notation: e.g. the number Pi is 3.14159265458, 10, and so on. Converting fractional (i.e. non-integer) numbers from decimal notation to binary notation (and in the other direction: from binary notation to decimal notation) is a (significant) source of rounding error.

The most well-known example of "rounding error" due to the conversion from decimal to binary notation is "1-0.9-0.1" that does not give 0 as output as expected (it gives 2e-17 instead).

The conversion from decimal notation (used by humans to represent numbers inside a **text**) to binary notations (the way the numbers are manipulated and stored inside a computer) is usually the most common and most important source of "rounding errors" in a computer program. Hopefully, for **integer** numbers, we don't loose any precision during the conversion.

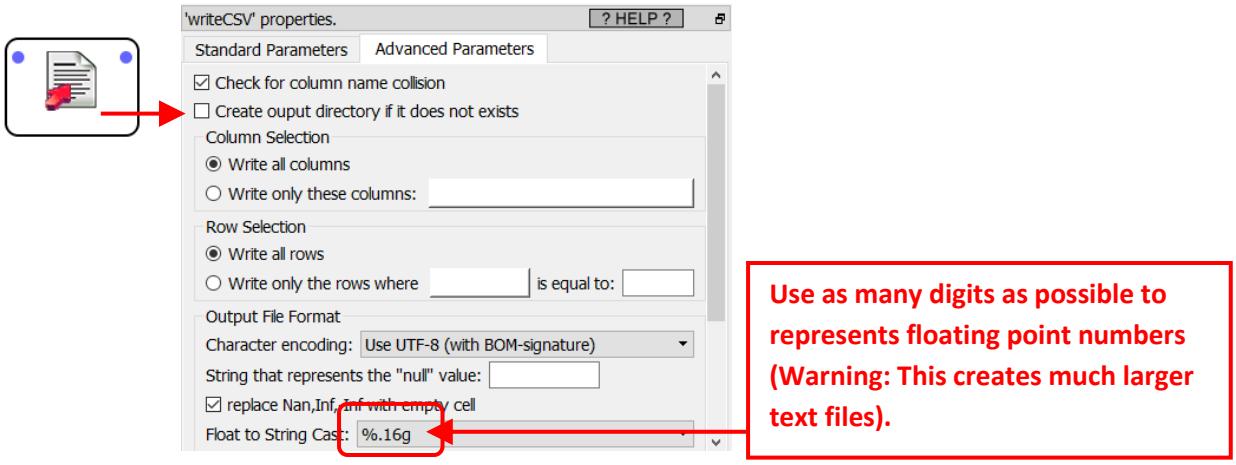
Anatella tries its best to avoid any "rounding errors" and keeps the floating-point numbers stored in binary representation at all the time. For example, when Anatella connects to a database through ODBC or OleDB, it always tries to receive the floating-point numbers directly in binary notation (to avoid the "bad" conversion from/to decimal notation). In the same spirit, when Anatella make

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

some "INSERT" into a database (using the ODBC Upsert Action or the TeradataWriter Action), it also sends to the database the floating-point numbers in Binary notation, as bits&bytes (without converting them to text: i.e. there is no conversion to decimal notation).

This means that, with Anatella, you can safely "copy" floating-point numbers from one database to another database without any loss of accuracy or precision because all the data is processed "in binary" (i.e. there is no conversion to decimal notation) (i.e. in the data preview window, the columns are all on a "blue background"). This ensure a total precision: i.e. 0% loss of accuracy. With most other ETLs (e.g. ETL in Java), you always have a loss of accuracy.

When you are using standard "bulk upload" tools, you'll lose some precision on the floating-point numbers because these tools expect some text files as input (and inside the text files, all the floating-point numbers are converted to decimal notation). To reduce to the minimum this loss of precision, you should instruct Anatella to use as many digits as possible to represents floating-point numbers in the text file: This is done using this parameter inside the writeCSV Action:



10.10.4. Securely adding rows to a database table

Let's assume that we want to add a few rows in a table (i.e. we will not refresh the entire table as in the section 10.10.2. but just add a few rows). Let's also assume that the computer might crash (or power off) while uploading these new rows into the database. In such situation, there will only be a small part of the rows that will be loaded into the database and it's difficult to properly restart the upload procedure: i.e. If we simply restart from scratch the upload process a second time, we will have some duplicated rows. This section offers some simple solution that guarantee that no rows will ever be lost or duplicated, even in the case of computer crash (or power off).

Let's categorize this "upload procedure" into 2 different problems:

1. There exist a Primary Key (that is an increasing number) inside the table to upload: See the next section 10.10.4.1.
2. There are no "usable" Primary Key inside the table to upload: See section 10.10.4.2.

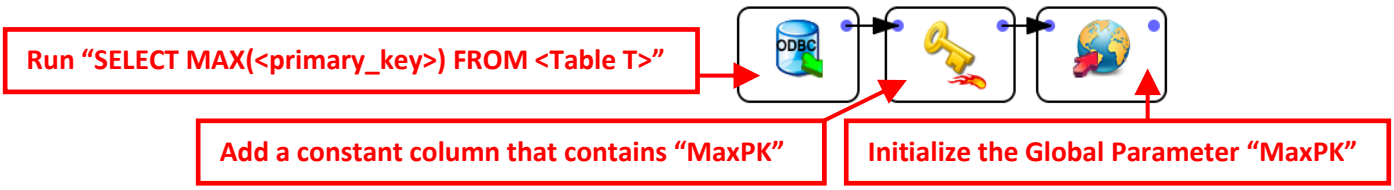
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.10.4.1. Incremental Table Upload

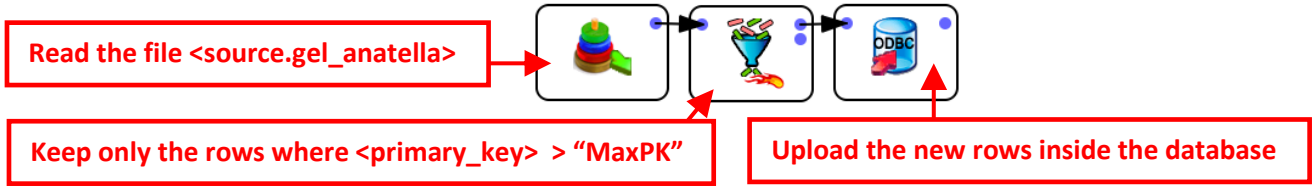
The procedure given in this section guarantees to have no duplicated or lost rows when uploading rows inside a database. This procedure also allows for "Incremental Upload" where we only upload inside the database the new rows that were not there yet.

To add a a few rows (available inside a file named <source.gel_anatella>) inside the "final" Table T, we will run the following steps:

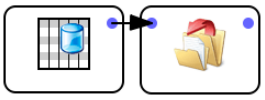
1. Run a "SELECT MAX(<primary_key>) FROM <Table T>" and save the result inside a "Global Parameter" named "MaxPK": This is done this way:



2. Run the following:



3. (Optional:) Delete the (local) file <source.gel_anatella>:



10.10.4.2. Secure Database Table Upload without Primary Key

To always guarantee to have no duplicated or lost rows, we will use an intermediary table.

To add a a few rows (available inside a file named <source.gel_anatella>) inside the "final" Table T, we will run the following steps:

1. (optional) CREATE an intermediary table (that has no INDEX)
2. Empty the intermediary table (i.e. run a "TRUNCATE TABLE" SQL command)
3. INSERT all the new rows from <source.gel_anatella> inside the intermediary table (See the previous section for more details about this INSERT procedure).
4. Delete the (local) file <source.gel_anatella>
5. Run a "INSERT INTO <Table T> SELECT * FROM <Intermediary Table>" SQL command (to copy the rows from the intermediary table to the final table T).
6. Empty the intermediary table (i.e. run a "TRUNCATE TABLE" SQL command)

With this 6-step procedure (via an intermediary table), you are 99% sure to have no "duplicate lines", even in the event of a machine crash (or power off). If the machine crashes, just restart the "complete" Anatella process. This procedure is 99% safe unless the PC crashes during step 4 (in this case, there will be some "lost" raws) but this is very unlikely because the step 4 is very short. In the case of machine crash (or power off), just restart the whole procedure from scratch.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We can also move the step 4 after the step 5 but then, in case of machine crash during step 5, there will be some "duplicated" rows. Furthermore, this might not be a good idea because the step 5 is very long.

To get a 100% safe procedure, we can use this slightly more complex procedure:

1. One time per day (or each time we get a new <source.gel_anatella> to upload inside the database): Save the result of a "SELECT COUNT(*) FROM <Table T>" into the local Anatella variable "Count_Table_T" (more precisely: store the variable "Count_Table_T" inside a text file)
2. Compare the result of a "SELECT COUNT(*) FROM <Table T>" to the local Anatella variable "Count_Table_T" (that has been read from the text file). if these 2 numbers are equal, run the upload procedure (i.e. run the steps 2.1. to 2.3.):

2.1. Empty the intermediary table (i.e. run a "TRUNCATE TABLE" SQL command)

2.2. INSERT all the new rows from <source.gel_anatella> inside the intermediary table (See the section 10.10.2 for more details about this INSERT procedure).

2.3. Run a "INSERT INTO <Table T> SELECT * FROM <Intermediary Table>" SQL command (to copy the rows from the intermediary table to the final table T).
3. Delete the (local) file <source.gel_anatella>

4. Empty the intermediary table (i.e. run a "TRUNCATE TABLE" SQL command)

The above procedure is 100% safe and guarantees that there will never be any lost or duplicated rows, even in the case of a computer crash (or power off). If a computer crash happens, just restart the whole procedure (excluding step 1) a second time.

To make things easier, you can save this whole procedure inside one "parametrized" Anatella graph (actually, you'll need 2 or 3 graphs and not one). In this way, you'll only need to call your graph (using the ParallelRun Action) to upload safely your rows to the database. Your "parametrized" Anatella graph will be using the 2 global parameters "<source_gel_file>" and "<destination_table>": See the section 5.1.5. for more information about "Graph Global parameters".

The use of an intermediary table when making large quantities of "INSERT" is very common for other reasons: e.g. when the final table T has an INDEX, it's usually much better to "go through" an Intermediary Table (to avoid losing time updating the INDEX): You'll get more details about the subject of "*Uploading rows inside a table with an INDEX*" inside section 5.27.4.1.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Because of the distributed nature of Teradata database, the update of the INDEX structure is very slow. This means that, 99% of the time, you must use an Intermediary Table (see the section 5.27.4.1. for more details about this subject). Also, to guarantee that no rows are ever duplicated or lost, you must also use an Intermediary Table (as explained in this section). This means that, when using Teradata, 99% of the time, we'll use an Intermediary Table to upload new rows inside the database.

This is why, when we created the TeradataWriter Action, we decided to use the "FastLoad" tool as the back-end (that is working "behind the scene" in the TeradataWriter Action). At first sight, the "FastLoad" tool might seem limited because it doesn't allow you to add rows inside a table that has an INDEX (and this means that we'll be forced to use an Intermediary Table to be able to add rows to a Table with an INDEX). …But since, we'll be using *anyway* 99% of the time an Intermediary Table (for the different reasons explained hereabove), the "FastLoad" tool becomes the most efficient choice (in 99% of the situations).

10.10.5. Saving Historical Data

It happens very often that you need to see the content of a database "as if" you were at a specific date in the past. This type of request happens mostly…

- …when you are creating predictive models: i.e. You need to create a learning dataset "as if" you were at specific observation date in the past.
- …for compliance reasons.

This is very easy to do when your table (inside your database) contains the two columns named "RECORD_VALIDITY_FROM" and "RECORD_VALIDITY_TO". Then, if you want to "see" a table "as-if" you were at a specific "Observation_Date", you'd simply write:

```
SELECT * FROM <My_Table>
WHERE (RECORD_VALIDITY_FROM < Observation_Date) AND (Observation_Date< RECORD_VALIDITY_TO)
```

Unfortunately, most tables/databases do not possess such columns. In such (common) situation, to still be able to safeguard all the "history" of your database (to be able to "go back" in time), you'll need to run everyday an Anatella graph that looks at the current content of your database and saves (inside different .gel_anatella files) all the changes made over the last day. More precisely, this Anatella graph needs to: For each Table T inside your database:

- Find inside the table T all the rows that changed.
- Add only the changed rows inside a <TableT.gel_anatella> file.
- Create (and thereafter, update) the columns "RECORD_VALIDITY_FROM" and "RECORD_VALIDITY_TO" inside the <TableT.gel_anatella> file.
- (optional): Create (and thereafter, update) the column "RECORD_CURRENT_FLAG" inside the <TableT.gel_anatella> file.

If you are lucky and you have a column named "LAST_UPDATE_TIME" inside your database (that contains the last time a row has been updated), then go to the next section 10.10.5.1. Otherwise, go to the section 10.10.5.2.

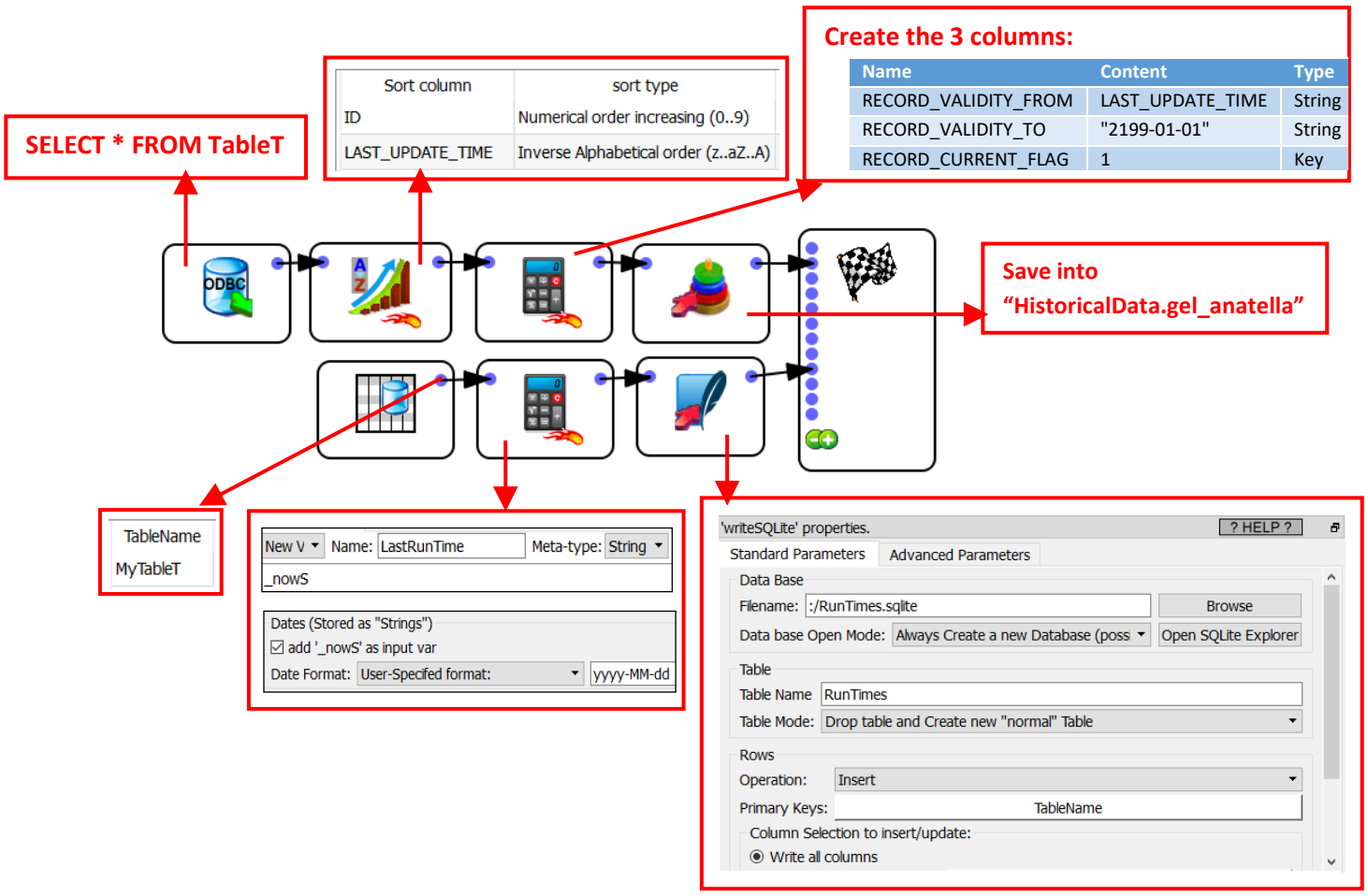
©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

10.10.5.1. You have a column named "LAST_UPDATE_TIME"

Inside this section, we'll assume that:

- You want to save all the changes made to a table named "TableT" located inside your database
- There is a column named "LAST_UPDATE_TIME" inside the "TableT" (that contains the last time the row has been updated).
- No row is ever deleted from the source database (i.e. there are only INSERT and UPDATE operations).
- There is a column named "ID" inside the "TableT" (that contains the primary key)
- All the historical data from the "TableT" will be saved inside the file "HistoricalData.gel_anatella"
- The file "HistoricalData.gel_anatella" contains 3 more additional columns (in addition to the columns from the Table "TableT") that are named "RECORD_VALIDITY_FROM", "RECORD_VALIDITY_TO" and "RECORD_CURRENT_FLAG".

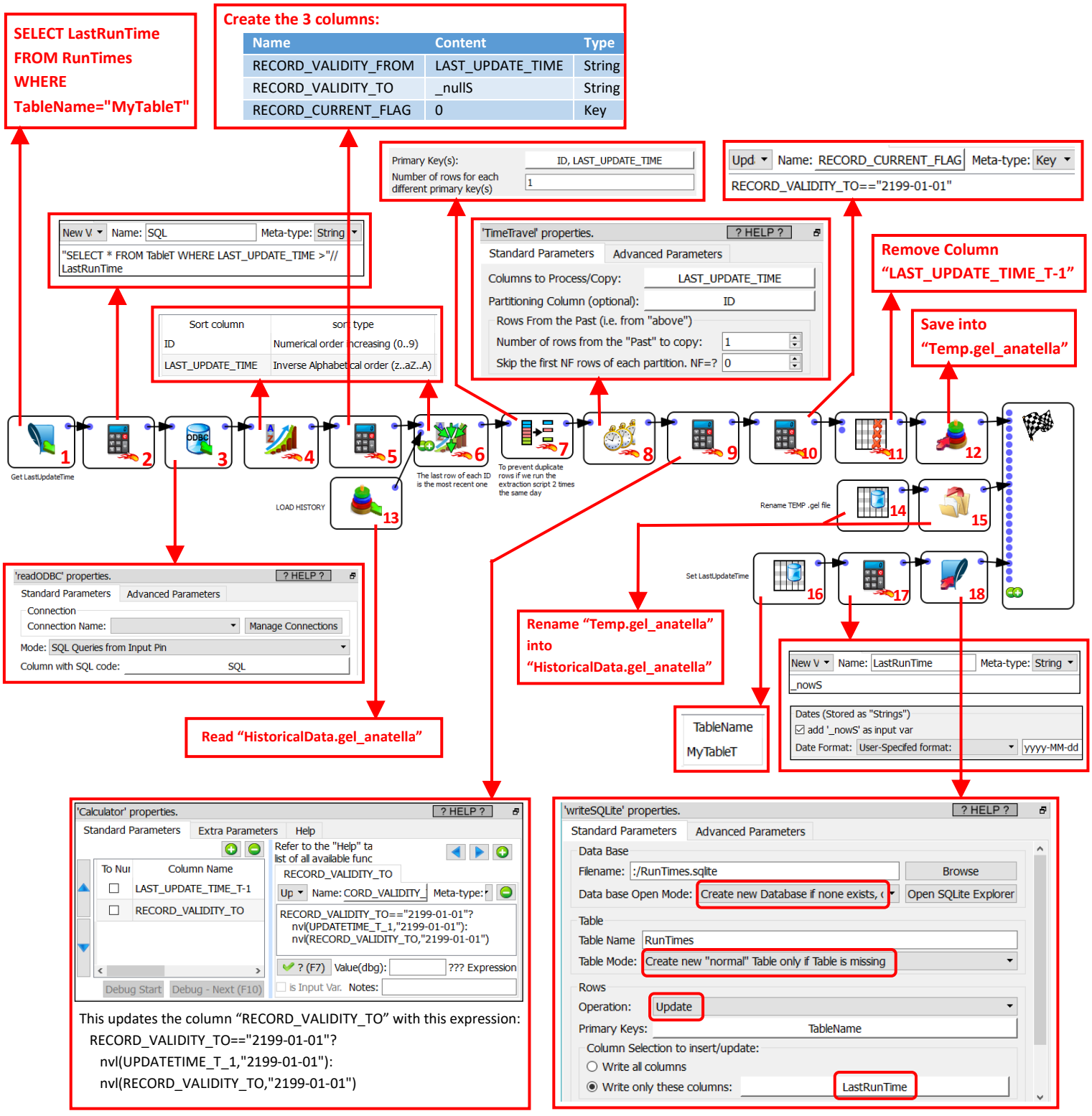
Here is the initial data extraction graph. You only run this graph one time to create the initial version of "HistoricalData.gel_anatella":



Please note that we used herabove a small SQLite file to keep track of the last time that we are running the extraction procedure.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

You'll find on the next page the "incremental" extraction procedure that we run everyday. This is a quite straightforward Anatella graph.



Inside the above graph, we gave a small ID number to each Action. Here is a small summary of the objective of each Actions:

- Actions 1,2: This computes the SQL query to run inside your database taking into account the date of the last execution of the extraction procedure.
- Action 3: This actually extract from the database all the new rows.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Actions 4,5,6,13: This merges the new rows (that we just extracted) with the "old" rows (that were extracted a long time ago and that are stored inside "HistoricalData.gel_anatella"). In particular, the new rows (coming from the data base) are missing the 3 columns named "RECORD_VALIDITY_FROM", "RECORD_VALIDITY_TO" and "RECORD_CURRENT_FLAG", so, (using the Action 5) we add them.
- Action 7: Just a small "RowDeduplicate" Action to prevent duplicate rows if we run the extraction script 2 times on the same day
- Action 8,9: On some rows, we must update the column "RECORD_VALIDITY_TO". The update expression only working because the table is sorted in a very specific way (see the Action 6 to know the exact Sort Order).
- Action 10: Update the column "RECORD_CURRENT_FLAG".
- Actions 11,12,14,15: self-explaining.
- Actions 16,17,18: We save inside the SQLite file the date from today as the "last extraction date" (i.e. the last date when we ran the extraction procedure).

The above "incremental" extraction procedure is very efficient because, each day, it only sorts the "new" rows (i.e. it does not sort the whole historical data). Still, it's able to properly update the column "RECORD_VALIDITY_TO" over the whole historical table.

10.10.5.2. You have no column named "LAST_UPDATE_TIME" inside your database

In the previous section we could extract from the database only the rows that have changed by simply executing the SQL command:

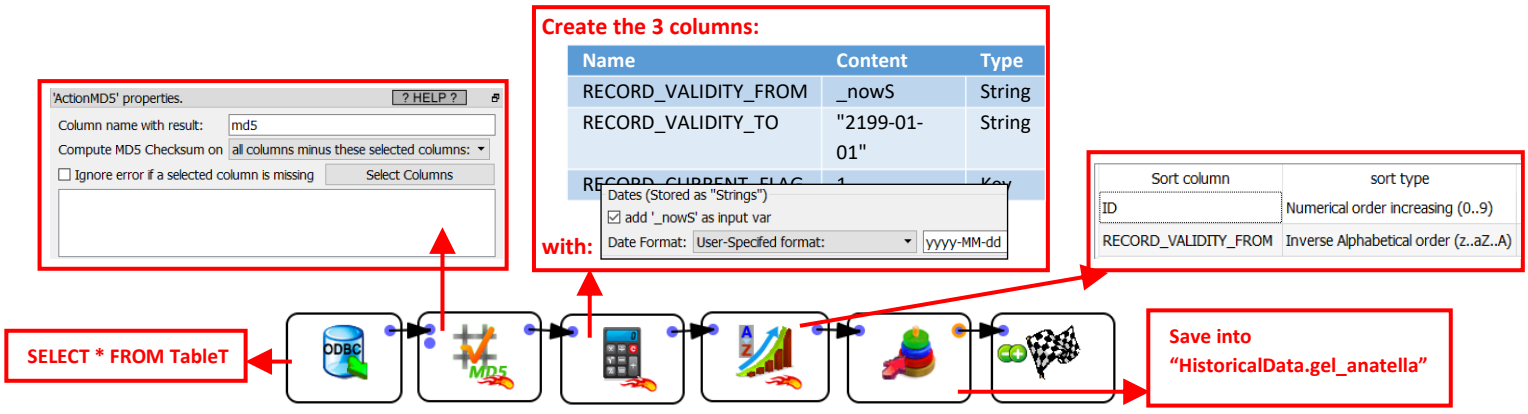
SELECT * FROM TableT WHERE LAST_UPDATE_TIME > <*LastRunTime*>

Since the column "LAST_UPDATE_TIME" is not available anymore, we'll have to extract every day the *whole* table from the database and resort to other ways to detect the rows that have been updated.

One efficient way to construct this "detection" mechanism is to use the MD5 Action (see section 5.19.5. for more information about the MD5 Action). To summarize:

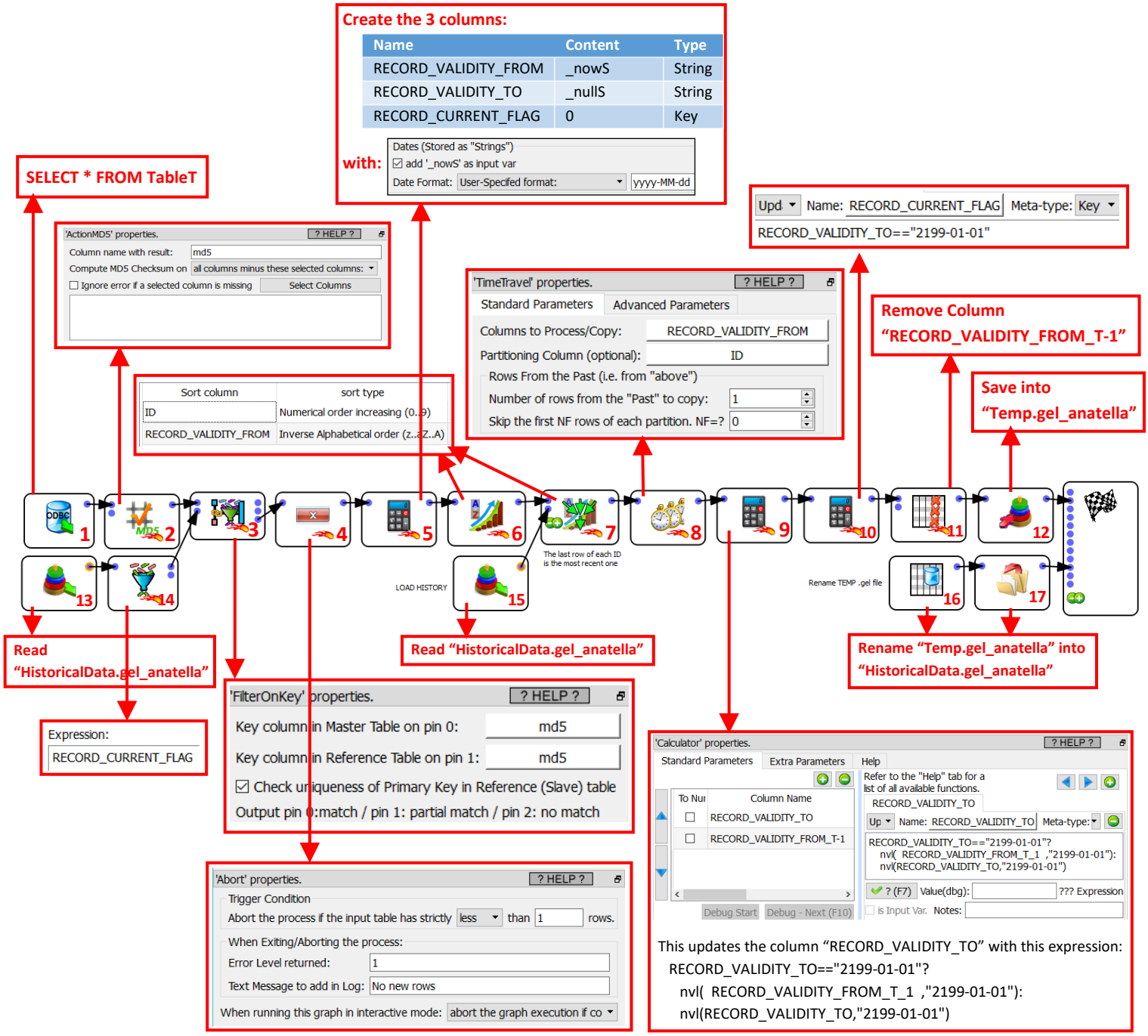
- the MD5 action compute a checksum that summarizes the content of the row.
- If the MD5 checksum on a row has "changed" between two extractions, it means that somewhere inside that row there is one cell that has changed: i.e. We can use the MD5 checksum as a detection mechanism to detect the rows that have been updated.

Here is the initial data extraction graph. You only run this graph one time to create the initial version of "HistoricalData.gel_anatella":



On the next page, you'll find the "incremental" extraction procedure that we run everyday. This is a quite "straight forward" Anatella graph.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



Inside the above graph, we gave a small ID number to each Action. Here is a small summary of the objective of each Actions:

- Action 1: This actually extract from the database all the rows.
- Actions 2,3,13,14: This filter only keeps the "new" rows (that are not inside the "HistoricalData.gel_anatella" file).
- Action 4: Stop the process if there are no new rows.
- Actions 5,6,7,15: This merges the new rows (that we just extracted) with the "old" rows (that were extracted a long time ago and that are stored inside "HistoricalData.gel_anatella"). In particular, the new rows (coming from the data base) are missing the 3 columns named "RECORD_VALIDITY_FROM", "RECORD_VALIDITY_TO" and "RECORD_CURRENT_FLAG", so, (using the Action 5) we add them.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

- Action 8,9: On some rows, we must update the column “RECORD_VALIDITY_TO”. The update expression is only working because the table is sorted in a very specific way (see the Action 7 to know the exact Sort Order).
- Action 10: Update the column “RECORD_CURRENT_FLAG”.
- Actions 11,12,16,17: self-explaining.

10.11. How to install both the 32-bit and the 64-bit versions of Anatella simultaneously?

You can install on the same PC, at the same time, both the “Anatella 32-bit” and the “Anatella 64-bit”. This means that you can still use the fastest “Anatella 64-bit” to do all your data transformations and only use the Anatella 32-bit to run the tasks that requires a 32-bit executable (e.g. to connect to a 32-bit Oracle database or to connect to a 32-bit MS-Outlook).

Both the Anatella 32-bit and the Anatella 64-bit are able to read the same .gel_anatella file format (and the same .cgel_anatella file format): these are universal formats. This means that you can use some simple .gel_anatella files to exchange data between Anatella 32-bit and Anatella 64-bit.

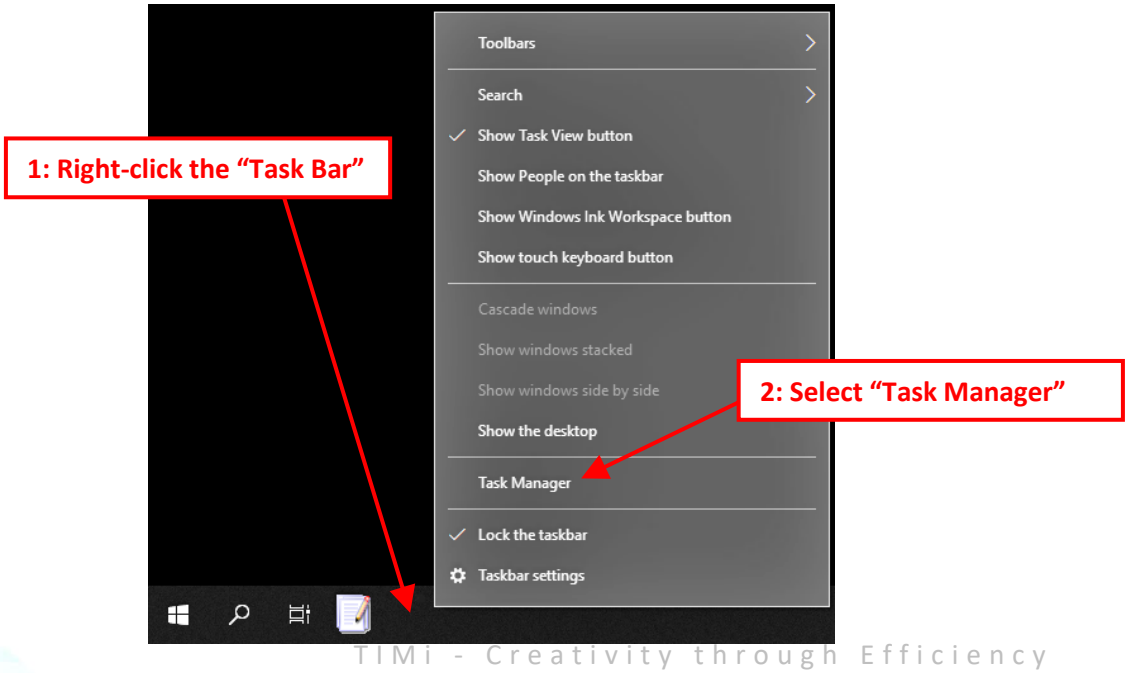
When you double-click on a .anatella file inside a MS-FileExplorer window, Windows automatically opens your .anatella file using Anatella (this mechanism is usually named “file association”). Most of the time, you want Windows to use the 64-bit version of Anatella to open your .anatella file, and never use the slower 32-bit version.

To be sure that MS-Windows always uses the 64-bit version of Anatella to open your .anatella file, you need to go inside the “bin” directory of the 32-bit Anatella installation and to delete the following executables:

“AnatellaServer.exe”
“TIMiServer.exe”
“TIMiFileAssociation.exe”

If these executables are already currently running, MS-Windows will prevent you to delete them. So, before deleting these executables, you might need to, first, stop them. To do so:

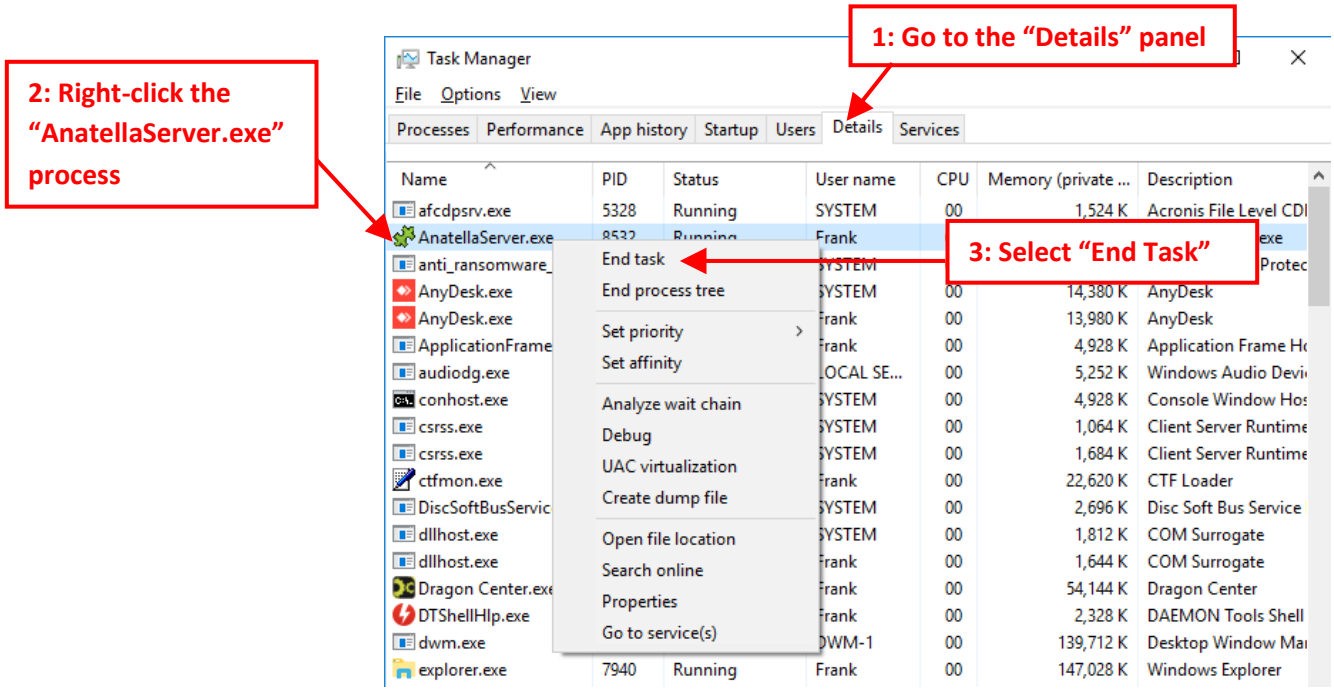
1. Open the MS-Window Task Manager: Right-click the “Task Bar” at the bottom of your screen and select “Task Manager”:



©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

2. Inside the “Task Manager” window, click “Mode details”:

3. Inside the “Details” panel of the “Task Manager” window, right-click the process “AnatellaServer.exe”, select “End task”. Then, confirm the stop of the process by clicking the “End Process” button in the confirmation dialog.



4. Follow the same procedure as above to stop the process “TIMiServer.exe”: i.e. Right-click the process “TIMiServer.exe”, select “End task”. Then, confirm the stop of the process by clicking the “End Process” button in the confirmation dialog.

To open a .anatella file inside an “Anatella 32-bit”, you need to:

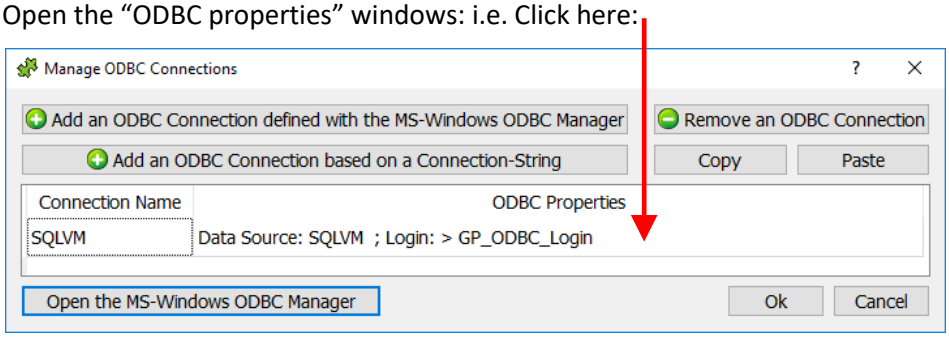
1. Run manually the “Anatella 32-bit” executable: i.e. double-click “Anatella.exe” inside your 32-bit installation directory.
2. Inside the “File” drop-down menu, select “Open” and browse for the desired .anatella file. Alternatively, you can also drag&drop a .anatella file from the MS-Window File Explorer window to the “Anatella 32-bit” window.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

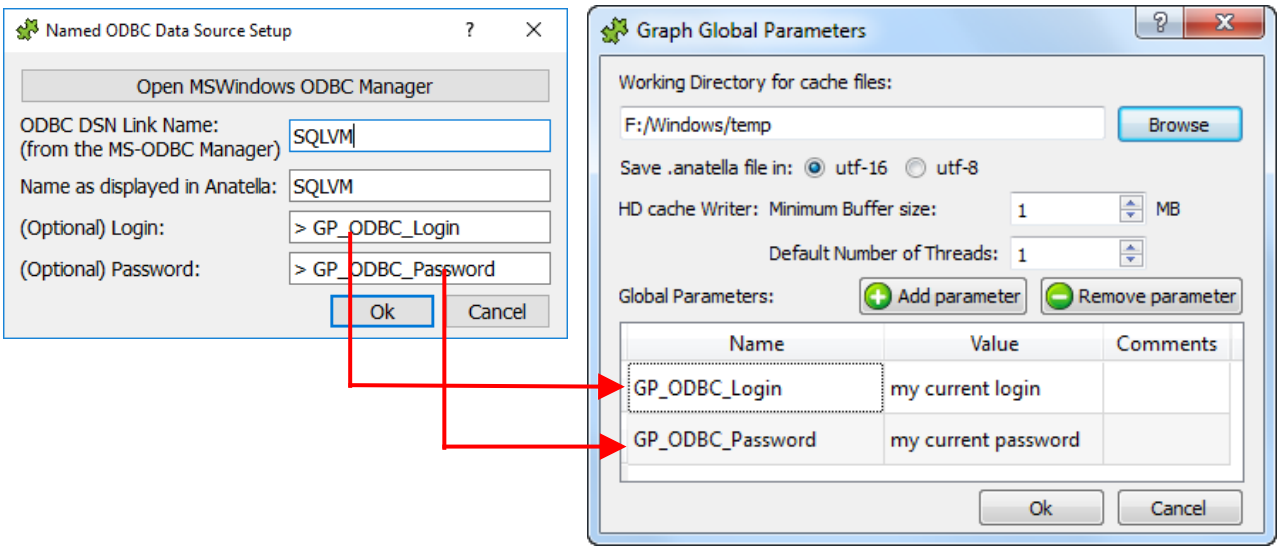
10.12. Automatic bulk change of ODBC connection parameters

There are two ways to change the ODBC connection parameters at regular interval:

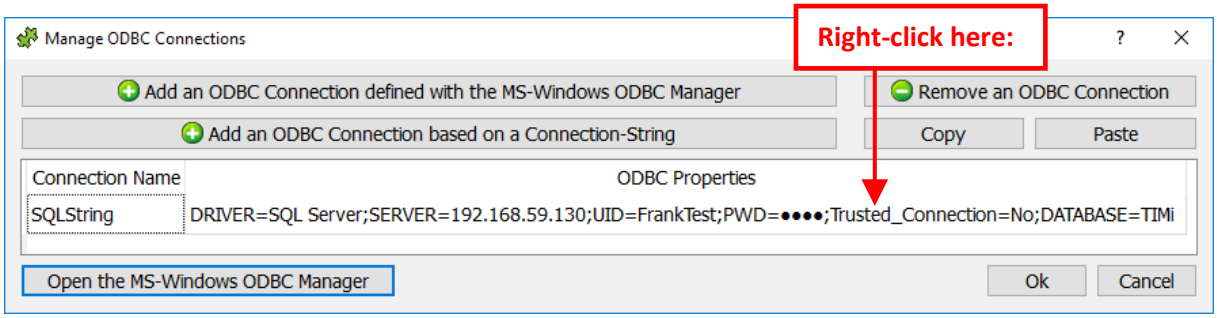
1. You can use the "Global Parameters" to define the ODBC connection parameters. Here are two examples:
   - **For a Type-1 ODBC connection:**
     Open the "ODBC properties" windows: i.e. Click here:



     Inside the "ODBC properties" windows, use the ">" character to specify "Global Parameters": For example, this is valid:



   - **For a Type-2 ODBC connection:**
     Right-click the ODBC connection string to edit it directly:



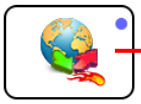
     Then, use the ">" character to specify "Global Parameters": For example, this is valid:

```
> "DRIVER=SQL Server;SERVER=192.168.59.130;UID="+GP_ODBC_Login+";PWD="+GP_ODBC_Password+";Trusted_Connection=No;DATABASE=TIMi"
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

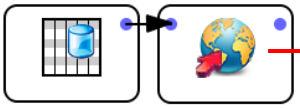
Hereabove we used some "Global Parameters" to define the ODBC connection parameters. Now, we need to update these "Global Parameters" to the right values, each time that the ODBC connection parameters are changed: You can do that in different ways:

- …using the command-line arguments: See section 4.7.1.
- … using this simple Action:



Update the "Global Parameters" using the connection parameters obtained from a .cfg file.

- … using these two Actions (this must be the first two actions executed in the graph):



Update the "Global Parameters" with the connection parameters obtained from the in-line Table.

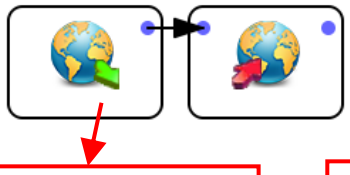
- …using these three Actions (this must be the first three actions executed in the graph):

Set the "Global Parameters" to the correct (i.e. decrypted) connection parameters

Read some encrypted connection parameters from a text file

Decrypt the connection parameters

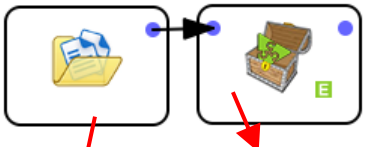
- … using these two Actions (this must be the first two actions executed in the graph): This is **deprecated**:



Read the connection parameters from a small configuration text file. (this is a deprecated Action)

Update the "Global Parameters" with the connection parameters obtained from the file.

2. You can use the Javascript Object named "AnatellaGraph" to update "in bulk" all the parameters (including the ODBC connection parameters) inside all your graphs: For example:



List all the .anatella files that you want to update.

Loop over all selected .anatella files.

At each iteration: open another .anatella file, update it, and save it.

```
function run()
{
   for(;;)
   {
      var r=getNextRow();
      if (r.isNull) break;

      var a=AnatellaGraph(r.col(0));
      a.setODBCPassword("my connection name","my new password");
      if (a.hasChanged()) a.save();
   }
   return 1;
}
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

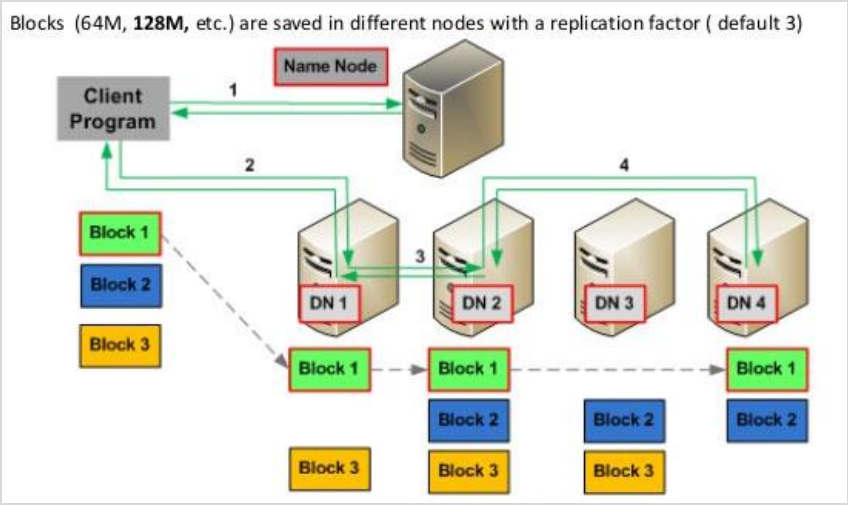
10.13. Where is the Hadoop integration within TIMi&Anatella?

If you used Anatella from 2017 to 2023, you might have noticed inside the Anatella GUI, some configuration screens that allowed to setup an access to an HDFS drive directly from Anatella. Starting with version v3.43, we decided to completely remove from Anatella the integration with HDFS. This section is an attempt at explaining why we took this difficult decision.

Back in 2017, our objective was to make Anatella the most efficient solution to work with data stored inside Hadoop. Indeed, Anatella is the only tool that can manipulate the famous .parquet files (using the readParquet Action and the writeParquet Action) using very efficient C/C++ code. Also, at that time, Anatella was amongst the very few tools that could natively access (read&write) the HDFS drive using efficient C/C++ code (and to the best of my knowledge, it still is?).

What's an HDFS drive? An HDFS drive is a logical drive that is built "on top" of several real hard drives that are physically placed in several (usually cheap) servers. In Hadoop terminology, these servers (that contains your data) are named "data nodes". An HDFS drive is composed of several "data nodes" (that contain data) and one "name node" (that is indexing the data contained in the HDFS drive: i.e. the name node contains an index that allows to know on which "data node" is stored each chunk/block of data).

Here is an illustration:



Despite the fact that we spend almost two years of development time to code our own highly-optmized proprietary Hadoop librairies in the very efficient C/C++ language (these libraries were used inside Anatella to read/write data from/to HDFS), the speed of the resulting end-system was still rather disapointing. I would even dare to say: catastrophically slow. More precisely, the read speeds from HDFS were limited to 5 to 15 MB/Sec on a common infrastructure. This is to be put in comparison with the common read-speed from the current NVMe SSD drives (2500 MB/Sec).

The speed limitation from HDFS originates from a very obvious bottleneck: the Java code running on the "Data node" servers. Anatella can read and process data at a speed around 1000 MB/sec (on 1

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

server equipped with common hardware) BUT, if the "Data Node" servers are not sending the data at a rate faster than 15MB/sec, it's pointless.

For a brief amount of time, it was our hope that, the once striving Hadoop ecosystem, would evolve to a point were the "Data Nodes" do not represent such a terrible "bottleneck". We now know that it will never happen. Indeed, the few remaining projects inside the Hadoop ecosystem have all dropped the HDFS protocol in favor of the S3 protocol. ..And this makes sense. The S3 protocol is much more cleverly designed, has better security features (all headers are "signed") and allow for a much higher throughput. Furthermore, there are now dozens of "open source" projects implementing the S3 protocol and practically all of them are faster and more robust than HDFS ever was. Indeed, it's our own experience that an HDFS cluster required hundreds of skilled technicians, just keep it *"up and running*" without doing anything with the data. And even with these many skilled technicians, it was very common to arrive in a company to see the HDFS cluster "down" for, on average, 1.5 day per week, each week. Basically, HDFS was never the robust solution that we hoped for.

HDFS was also a terrible security time-bomb waiting to explode. Just have a look at the table below that details all the Internet TCP-IP ports that needs to be opened to operate an HDFS cluster! Terrible!

TOOL	Port
HDFS WebUI for NameNode	9870/9871
HDFS Metadata service (NameNode)	hdfs://hdp-master:19000
HDFS Data Node	9864/9865 (https) and 9866 and 9867 (ipc)
HDFS Secondary NameNode	9868/9869
HDFS JournalNode	8485 (ipc) and 8480/8481 (http/https)
HDFS Aliasmap Server	50200
HDFS Namenode	Cloudera/HDP: 8020
HDFS Datanode (2)	50010 and 50020
HttpFS	14000
WebHDFS	Cloudera/HDP: 50070
YARN Resourcemanager	Cloudera: 8032 / HDP: 8050
YARN JobTracker	Cloudera/HDP: 8021
Hive Metastore (optional)	9083
YARN Resourcemanager Scheduler	8030
YARN collector-service.address	8048
YARN localizer.address	8040 (ipc)
YARN Resourcemanager Admin	8033

TOOL	Port
YARN Resourcemanager WebApp	8088
YARN Nodemanager WebApp	8042/8044 (http/https)
YARN sharedcache	8788
YARN sharedcache uploader server	8046
YARN sharedcache client server	8045
YARN amrm proxy	8049
YARN Timeline Service	10200 (ipc) or 8188/8190 (http/https)
YARN webapp	8089/8091 (http/https)
HDFS WebUI for NameNode	9870/9871
HDFS Metadata service (NameNode)	hdfs://hdp-master:19000
HDFS Data Node	9864/9865 (https) and 9866 and 9867 (ipc)
HDFS Secondary NameNode	9868/9869
HDFS JournalNode	8485 (ipc) and 8480/8481 (http/https)
HDFS Aliasmap Server	50200
MapReduce Job History	10020
MapReduce Job History UI	19888/19890 (http/https)
MapReduce History server admin	10033 (ipc)
HiveServer2 (optional)	TCP connection: 10000 / HTTP connection: 10001

There are no less than 57 TCP-IP ports required to operate an HDFS cluster! Crazy! And, as a consequence, we have seen one of our major customer being the target of a **ransom-ware**, twice in a time-span of only 6 months. The hackers managed to infiltrate the pc network through a vulnerability inside the HDFS servers.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

What's a ransom-ware? That's a little software the operates silently behind your back on your servers. It does three things:
* It reads the data from the local SSD and write it back on the local SSD in an encrypted form. The encryption is performed using an encryption key that is only known from the hacker. Depending on the size of the SSD, it can take from 1 day to 1 week to completely encrypt all the files from the SSD.
* It intercepts all the low-level "read" attempts on the SSD and decrypt on-the-fly the data coming from the SSD using the secret encryption key only known from the hacker. Basically, all the software running on the OS are totally unaware that some decryption occurred at some point because they just receive their data as usual, each time that they read something from the SSD.
* It attempts to "infect" neighboring machines located on the same local network.

Typically, the hacker waits around one week (to be sure all the SSD's are fully encrypted) before sending a signal to the ransom-ware tool that stops the "*on-the-fly decryptions*" of all the files. At that point, you must pay the hacker a "*ransom*" to get the encryption keys that allows you to decrypt everything to get access back to the data stored on your SSD's again. For a large company, this "*ransom*" is usually in the order of the million of euros/dollars.

The HDFS project did also undergo several code-breaking changes that were designed to break compatibility between different versions of HDFS. These code-breaking changes were engineered on-purpose to allow Databricks to remain the only player commercially active inside the Hadoop ecosystem (more on this subject in the gray box juste here below).

For those of you interested in the chronology of events that lead to the current position of Databricks being the only player active in the Hadoop ecosystem, here is a little timeline:
*** 2006:** Google published a seminal research paper where they describe their own version of HDFS named "Google big table" (long before HDFS existed). Some Java developers decided to code in Java something that is similar to "Google big table". They name it HDFS (Hadoop File System). The HDFS project is accepted in the Apache foundation.
*** 2011:** The Hadoop ecosystem is getting traction and the company HortonWorks is created to cash-in on the large investments made by the bankers that were lured to invest their money in this "promising technology".
*** 2012:** There are now three major players in the Hadoop ecosystem: HortonWorks, Cloudera and MapR. HortonWorks is majoritarly composed of Java coders: they produce 90% of the Hadoop code. Cloudera is is majoritarly composed of marketeers and resellers: they produce almost no code but they sell like crazy. Their promise is simple: "*Add more machine, and you get higher speed.*". We will see later that this is in fact a plain and stupid lie: This is 100% bullshit. MapR is the only one trying to make Hadoop efficient (they do C code instead of Java code).
*** 2012:** The Cloudera company buys controlling shares of the O'Reilley company. This purchase is kept secret. The O'Reilley company is a trusted publisher of good books related to coding since 1978: They have the trust of millions of coders all over the world. Every coder on the planet opened at some point a book by O'Reilley! The O'Reilley company starts organizing early events that are making the promotion of the Hadoop ecosystem, and of Cloudera in particular. These events are immense. Coders from all around the planet are flooding in to listen to the falacious marketing arguments from Cloudera (under the disguise of the good old O'Reilly). Personnaly, I think that this was the best single move that Cloudera did over all these years. Through this acquisition of O'Reilly, they were able to control the narrative around Hadoop: They were able to silence all dissenting voices.
*** 2013:** The Company DataIku is created. The mission from this company is to simplify the usage of Hadoop by providing an easy interface that minimizes the quantity of line of codes that you must write to operate your Hadoop system. In essence, the promise of Dataiku is simple: "*Do you want to do big data with Hadoop? Will make it run for you.*"

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

*** 2014:** TeraData (a famous company in the database world, known for operating the largest databases in the world) integrates all his database systems with Hadoop and become a major official Hadoop reseller.
*** 2015:** At that point some people start to suspect that the promise made by Cloudera and HortonWorks is bullshit. To remind you, their promise is: "*Add more machines, and you get higher speed".* ..but this is a lie. To avoid getting trapped in their lie, Cloudera invents a new lie: "*You did't get more speed (as we promised) because you didn't add enough machines: Go to AWS to get more machines.*". AWS joins the party. This a marriage made in heaven. AWS sells their machines like crazy and decide to promote Hadoop using insane marketing budgets: Literally, hundreds of millions of euros/dollars are poured to promote this non-working technology.
*** 2016:** Hadoop is at its peak. The marketing machine from AWS, Cloudera, O'Reilley, HortonWorks, IBM works at full regime. Afterwards, that's the downfall.
*** 2017:** (June) IBM saw through the lies of Cloudera. IBM gets rid of all the business around Hadoop. They abandon their own Hadoop platorm named "IBM Big Insight" and they sell all their Hadoop business-lines to HortonWorks.
*** 2018 (October):** Cloudera buys HortonWorks.
*** 2018:** Teradata officially abandon the Hadoop Ecosystem. According to Teradata, Hadoop is a total failure. And we know why: Hadoop actually never delivered on their promise "*Add more machines, and you get higher speed*". It was a lie all along. More details in a blog entry from Teradata here (blog entry from June 2019).
*** 2018:** I have a private discussion with a friend that codes for Cloudera. He explains me that the code from Hadoop (i.e. the code of their main tool : Spark) is not designed for speed or efficiency: This code is designed with only one goal in mind: ***Obscursification***. Only the coders from Cloudera have the "*Rule Book*" that allows them to "see through" the layers of "Code Obscursifications". In other words, Hadoop might be an "open source" project and you can try to read the code from Hadoop but you won't be able to understand it: Only the coders from Cloudera can do that. Hadoop/Cloudera only wears the disguise of "open source" but, in reality, Cloudera/Hadoop is a totally closed solution, as closed as any major player in the Big Data space (TIMi, Teradata, Oracle, Google Big Table, etc.).
*** 2019:** Many times over the 2019 year, Cloudera makes breaking changes after breaking changes inside the official Hadoop distribution. All the tools inside the Hadoop eco-system just stop working because of all these breaking changes. All the commercial players active in the Hadoop ecosystem are dying, one after the others.
*** 2019 (August):** MapR declares banckrupcy. Now, the only player left is Cloudera. ..But the whole Hadoop eco-system is pissed against them because Cloudera killed everybody (on purpose). They managed to kill everybody by engineering many code-breaking changes to break every Hadoop systems, except theirs.
*** 2022:** DataIku changes their mission to avoid the banckurpcy. Because of Cloudera, they cannot make Hadoop work properly anymore (or, at least, it requires hundreds of engineers to barely keep the Hadoop system running), so they announce: "*We won't help you do Big Data with Hadoop anymore like we did in the past. Now, our mission is to help you do Little Data (with anything but Hadoop).*". This is a complete betrayal of their core value and of their core mission. Only time will tell if they manage to survive such a betrayal! You can see it here:

https://youtu.be/7e54BbFmR34?t=19

This is the video where they made the announcement of their betrayal (in French).
Here are 3 screenshots of this Youtube video:

Big Data, for me, it's dead.

So, it is no longer Big but Small

What comes to my mind is Small Data

*** 2022:** Cloudera is sold to an American private equity firm named "Clayton, Dubilier & Rice and KKR". This sale effectively allows Cloudera to change their name (since it acquired such a bad reputation after the events from 2019) but don't be fooled: They are still there under the name "DataBricks".
*** 2023+:** Databricks (formely Cloudera) is now the sole player in the Hadoop ecosystem. Databricks is a closed-source solution that operates under the disguise of "open source" (through code obscurcification). They still maintain the same false promise: "*Add more machines, and you get higher speed*". To undertand why this is a lie, I invite you to watch this youtube video:
https://www.youtube.com/watch?v=rgITj1N6wBo
This video explains that the speed of Spark (that is the fastest, most well-known tool inside the Hadoop ecosystem) does not increase when you add more machines: This was all a lie from the start. This is the consequence of a mathematical law named "Amdhal's law".

This means that a single Anatella server is faster than a Hadoop/Spark cluster of any size (i.e. with any quantity of "nodes"). This is due to the relatively high incompressible time of Hadoop/Spark: you find more details on this subject here: https://timi.eu/blog/cloud/

So, the HDFS project inside the Hadoop ecosystem is a dead-end. …But what about the other projects available inside this ecosystem? They are equally as dead as HDFS. For example, the parquet file-format has also undergo several code-breaking changes. This means that there are great chances that the Anatella Actions that read&writes parquet files won't be compatible with the parquet files from your Databricks system. If you are using the old "official" code repository for Hadoop from the Apache foundation, then it should work.

That was our hope that Hadoop would evolve into a striving ecosystem. We invested nearly 2 years of development to get the best solution available inside this ecosystem: e.g. the only solution coded 100% in C and assembler. ..But we are now forced to recognize that the whole ecosystem is a dead-end. We lost 2 years of development for nothing. ..But we might consider ourselves lucky: Many data-related companies did not survive the downfall of Hadoop. And for those companies that are still using Hadoop now and that refuse to aknowledge the basic truth that Cloudera/Databricks lied about their promise, they are in a very dangerous position where they"ll pay sooner or later a heavy price for their missguided trust…

What about the future after Hadoop?
As you might have guessed, we won't invest more energy and time in making TIMi/Anatella more compatible with Hadoop. Since Hadoop is now dead or dying, there do no exists anymore many solutions left that are really focused on "Big Data". TIMi is one of them. Teradata Vintage is one.

TIMi has several advantages compared to the other solutions in the "Big Data" space:
1. TIMi is much faster and much more scalable than any other solution currently on the market.
2. TIMi can do distributed computing (e.g. see the Action "LoopJenkins" in section 5.22.1). But, 99% of the time, one TIMi server will be more than enough to handle all your data processing needs, since TIMi is so efficient and fast.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

3. TIMi can store all its data inside a distributed/cloud storage that is S3-compatible.
   TIMi is compatible with many free solutions that do exactly that. For example: MinIO, Ceph, OpenIO, OpenStack Swift, etc.
   This means that you can manage with TIMi peta-byte-scale datasets. …But, 99% of the time, one TIMi server will be more than enough to store all your data, since TIMi uses its own proprietary compressed file format that is very efficient at reducing data size. For example, a Teradata database that "weights" 2.5 TBytes is stored using only 25GB inside the proprietary file format from TIMi.
4. Thanks to a unique "streaming" approach, the R&Python engines included inside TIMi can bypass the limitations on the volumetry for the datasets stored in RAM memory. This allows to use R&Python on massive datasets with hundreds of billions of rows without falling into "*memory allocation errors*" issues.

11. Appendix

11.1. Appendix A: Introduction to the Standard JavaScript Language

This section presents a brief introduction to the JavaScript language so that you can understand the code snippets presented in the rest of the guide (mainly in section 6), so that you can start writing your own scripts.

Inside Anatella, you can write new Actions using Javascript (and also R, Python or C/C++). The "official" name of the JavaScript-language used in Anatella is ECMAScript, the standardized version of JavaScript. ECMAScript forms the basis of JavaScript (Mozilla), JScript (Microsoft), and ActionScript (Adobe). Although the language's syntax is superficially similar to C++ and Java, the underlying concepts are somewhat different.

We will now quickly review the fundamental of the JavaScript language. A summary of all standard functionalities offered by any JavaScript/ECMAScript compliant language is given in appendix E. The special extentions of Javascript provided by Anatella are listed in appendix F.

The Mozilla Foundation's web site hosts a more complete tutorial at http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Guide, and David Flanagan's "*JavaScript: The Definitive Guide (O'Reilly, 2006)*" is recommended both as a tutorial and as a reference manual. The official JavaScript / ECMAScript specification is available online at http://www.ecmainternational.org/publications/standards/Ecma-262.htm.

The basic JavaScript control structures—if statements, for loops, and while loops—are the same as in C++ and Java. JavaScript also provides more or less the same assignment, relational, and arithmetic operators. JavaScript strings support concatenation with + and appending with +=.

To get a feel for the JavaScript syntax, we will start by a small example:

```
function square(x)
{
    return x * x;
}

function sumOfSquares(array)
{
    var result = 0;
    for (var i = 0; i < array.length; ++i)
        result += square(array[i]);
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
        return result;
    }

    var array = new Array(100);
    for (var i = 0; i < array.length; ++i)
        array[i] = (i * 257) % 101;

    print(sumOfSquares(array));
```

This will print “336234” in the console.

From a classical Java or C++ programmer's perspective, probably the most striking feature of JavaScript is that variables are not explicitly typed; the var keyword is all that is required to declare a variable. Read-only variables are declared with const instead of var. Another noteworthy feature of the preceding program is that there is no main() function. Instead, the code that is located outside any function is executed immediately, starting from the top of the file and working down to the bottom.

Unlike in Java or C++, semicolons at the end of statements are generally optional in JavaScript. Using sophisticated rules, the interpreter can insert most missing semicolons itself. Despite this, typing semicolons ourselves is recommended, to avoid unpleasant surprises.

If we don't provide an initial value when declaring a variable with var, the default value is undefined, a special value of type Undefined. We can later assign any value of any type to the variable using the assignment operator (=). Consider the following examples:

```
var x;
typeof x;           // returns "undefined"

x = null;
typeof x;           // returns "null"

x = true;
typeof x;           // returns "boolean"

x = 5;
typeof x;           // returns "number"

x = "Hello";
typeof x;           // returns "string"
```

The typeof operator returns a lowercase string representation of the data type associated with the value stored in a variable. JavaScript defines five primitive data types: Undefined, Null, Boolean, Number, and String. The Undefined and Null types are special types for the undefined and null constants, respectively. The Boolean type consists of two values, true and false. The Number type stores floating-point numbers. The String type stores Unicode strings.

Variables can also store objects and functions, corresponding to the data types Object and Function. For example:

```
x = new Array(10);
typeof x;           // returns "object"

x = print;
typeof x;           // returns "function"
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Like Java, JavaScript distinguishes between primitive data types and object types. Primitive data types behave like C++ value types, such as int and double. These are created without the new operator and are copied by value. In contrast, object types must be created using the new operator, and variables of these types store only a reference (a pointer) to the object. When allocating objects with new, we do not need to worry about releasing their memory, since the garbage collector does this automatically.

If we assign a value to a variable without declaring it first using the "var" keyword, the variable will be created as a global variable. And if we try to read the value of a variable that doesn't exist, we get a ReferenceError exception. We can catch the exception using a try ... catch statement, as follows:

```
try {
    print(y);
} catch (e) {
    print(e.name + ": " + e.message);
}
```

If the variable "y" does not exist, the message "`ReferenceError: y is not defined`" is printed on the console.

If undefined variables can cause havoc in our programs, so can variables that are defined but that hold the undefined constant—the default value if no initializer is provided when declaring a variable using var. To test for undefined, we can use the strict comparison operators === or !==. For example:

```
var x;
...
var y = 0;
if (x !== undefined)
    y = x;
```

The familiar == and != comparison operators are also available in ECMAScript, but unlike === and !==, they sometimes return true when the compared values have different types. For example, 24 == "24" and null == undefined return true, whereas 24 === "24" and null === undefined return false.

Let's go back to our first example. This first program illustrates how to define our own functions in JavaScript:

```
function square(x)
{
    return x * x;
}

function sumOfSquares(array)
{
    var result = 0;
    for (var i = 0; i < array.length; ++i)
        result += square(array[i]);
    return result;
}

var array = new Array(100);
for (var i = 0; i < array.length; ++i)
    array[i] = (i * 257) % 101;

print(sumOfSquares(array));
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Functions are defined using the "`function`" keyword. In keeping with JavaScript's dynamic nature, the parameters are declared with no type, and the function has no explicit return type.

By looking at the code, we can guess that the "*square()*" function should be called with a Number and that the "*sumOfSquares()*" function should be called with an Array object, but this doesn't have to be the case. For example, square("7") will return 49, because JavaScript's multiplication operator will convert strings to numbers in a numeric context.

Similarly, the sumOfSquare() function will work not only for Array objects but also for other objects that have a similar interface.

In general, JavaScript applies the duck typing principle: "*If it walks like a duck and quacks like a duck, it must be a duck*". This stands in contrast to the strong typing used by C++ and Java, where parameter types must be declared and arguments must match the declared types.

Instead of defining an array variable, we can pass an array literal:

```
print(sumOfSquares ([4, 8, 11, 15]));
```

JavaScript lets us supply more arguments to a function than there are parameters declared. The extra arguments are accessible through the arguments array. Consider the following example:

```
function sum()
{
    var result = 0;
    for (var i = 0; i < arguments.length; ++i)
        result += arguments[i];
    return result;
}

print(sum(1, 2, 3, 4, 5, 6));
```

Here, the sum() function is defined to take a variable number of arguments. The arguments are the numbers that we want to sum.

The arguments array can be used to overload the behavior of functions based on the types of the arguments or on their number.

More information about advanced features of Javascript is given in the next appendix (Appendix B). A summary of all standard functionalities offered by any JavaScript / ECMAScript compliant language is given in appendix E. The special extentions of Javascript provided by Anatella are listed in appendix F.

11.2. Appendix B: Advanced JavaScript tutorial.

It's not required to read this section in order to understand how to create new powerful Script-based-Anatella Actions.

You can safely skip this section, if you are not interested in advanced & specific notions related to the JavaScript language.

For Java and C++ programmers, arguably the most difficult aspect of JavaScript is its object model. JavaScript is an object-based object-oriented language, setting it apart from C++, C#, Java, Simula, and

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Smalltalk, which are all class-based. Instead of a class concept, JavaScript provides us with lower-level mechanisms that let us achieve the same results.

The first mechanism that lets us implement classes in JavaScript is that of a constructor. A constructor is a function that can be invoked using the new operator. For example, here is a constructor for a Shape object:

```
function Shape(x, y) {
    this.x = x;
    this.y = y;
}
```

The Shape constructor has two parameters and initializes the new object's x and y properties (member variables) based on the values passed to the constructor. The this keyword refers to the object being created. In JavaScript, an object is essentially a collection of properties; properties can be added, removed, or modified at any time. A property is created the first time it is set, so when we assign to this.x and this.y in the constructor, the x and y properties are created as a result.

A common mistake for C++ and Java developers is to forget the this keyword when accessing object properties. In the preceding example, this would have been unlikely, because a statement such as x = x would have looked very suspicious, but in other examples this would have led to the creation of spurious additional variables.

To create (or “instantiate”) a new Shape object, we use the new operator as follows:

```
var shape = new Shape(10, 20);
```

If we use the typeof operator on the shape variable, we obtain Object, not Shape, as the data type. If we want to determine whether an object has been created using the Shape constructor, we can use the instanceof operator:

```
var array = new Array(100);
array instanceof Shape;             // returns false

var shape = new Shape(10, 20);
shape instanceof Shape;             // returns true
```

JavaScript lets any function serve as a constructor. However, if the function doesn't perform any modifications to the “this” object, it doesn't make much sense to invoke it as a constructor. Conversely, a constructor can be invoked as a plain function, but again this rarely makes sense.

In addition to the primitive data types, JavaScript provides built-in constructors that let us instantiate fundamental object types, notably Array, Date, and RegExp. Other constructors correspond to the primitive data types, allowing us to create objects that store primitive values. The valueOf() member function lets us retrieve the primitive value stored in the object. For example:

```
var boolObj = new Boolean(true);
typeof boolObj;                     // returns "object"

var boolVal = boolObj.valueOf();
typeof boolVal;                     // returns "boolean"
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

We have seen how to define a constructor in JavaScript and how to add member variables to the constructed object. Normally, we also want to define member functions. Because functions are treated as first-class citizens in JavaScript, this turns out to be surprisingly easy. Here's a new version of the Shape constructor, this time with two member functions, "*manhattanPos()*" and "*translate()*":

```
function Shape(x, y) {
    this.x = x;
    this.y = y;
    this.manhattanPos = function() {
        return Math.abs(this.x) + Math.abs(this.y);
    };
    this.translate = function(dx, dy) {
        this.x += dx;
        this.y += dy;
    };
}
```

We can then invoke the member functions using the . (dot) operator:

```
var shape = new Shape(10, 20);
shape.translate(100, 100);
print(shape.x + ", " + shape.y + " (" + shape.manhattanPos() + ")");
```

With this approach, each Shape instance has its own "*manhattanPos*" and "*translate*" properties.

Our "shape" object possesses several properties (the "manhattanPos" and "translate" functions for example). These properties should be identical for all Shape instances, it is desirable to store them only once rather than in every instance. JavaScript lets us achieve this by using a prototype. A prototype is an object that serves as a fallback for other objects, providing an initial set of properties. One advantage of this approach is that it is possible to change the prototype object at any time and the changes are immediately reflected in all objects that were created with that prototype. Consider the following example:

```
function Shape(x, y) {
    this.x = x;
    this.y = y;
}

Shape.prototype.manhattanPos = function() {
    return Math.abs(this.x) + Math.abs(this.y);
};

Shape.prototype.translate = function(dx, dy) {
    this.x += dx;
    this.y += dy;
};
```

In this version of Shape, we create the "*manhattanPos*" and "*translate*" properties outside the constructor, as properties of the "*Shape.prototype*" object. When we instantiate a Shape, the newly created object keeps an internal pointer back to "*Shape.prototype*". Whenever we retrieve the value of a property that doesn't exist in our Shape object, the property is looked up in the prototype as a

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

fallback. Thus, the Shape prototype is the ideal place to put member functions, which should be shared by all Shape instances.

It might be tempting to put all sorts of properties that we want to share between Shape instances in the prototype, similar to C++'s static member variables or Java's class variables. This idiom works for read-only properties (including member functions) because the prototype acts as a fallback when we retrieve the value of a property. However, it doesn't work as expected when we try to assign a new value to the shared variable; instead, a fresh variable is created directly in the Shape object, shadowing any property of the same name in the prototype. This asymmetry between read and write access to a variable is a frequent source of confusion for novice JavaScript programmers.

In class-based languages such as C++ and Java, we can use class inheritance to create specialized object types. For example, we would define a Shape class and then derive Triangle, Square, and Circle from Shape. In JavaScript, a similar effect can be achieved using prototypes. The following example shows how to define Circle objects that are also Shape instances:

```
function Shape(x, y) {
    this.x = x;
    this.y = y;
}

Shape.prototype.area = function() { return 0; };

function Circle(x, y, radius) {
    Shape.call(this, x, y);
    this.radius = radius;
}

Circle.prototype = new Shape;
Circle.prototype.area = function() {
    return Math.PI * this.radius * this.radius;
};
```

We start by defining a Shape constructor and associate an "*area()*" function with it, which always returns 0. Then we define a Circle constructor, which calls the "base class" constructor using the "*call()*" function defined for all function objects (including constructors), and we add a radius property. Outside the constructor, we set the Circle's prototype to be a Shape object, and we override Shape's "*area()*" function with a Circle-specific implementation. This corresponds to the following C++ code:

```
class Shape
{
public:
    Shape(double x, double y) {
        this->x = x;
        this->y = y;
    }
    virtual double area() const { return 0; }
    double x,y;
};

class Circle : public Shape
{
public:
    Circle(double x, double y, double radius) : Shape(x, y)
    {
```


©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

```
        this->radius = radius;
    }
    double area() const { return M_PI * radius * radius; }
    double radius;
};
```

This corresponds to the following Java code:

```
class Shape
{
    public Shape(double _x, double _y) {
        x = _x;
        y = _y;
    }
    public double area() { return 0; }
    double x,y;
};

class Circle extends Shape
{
    Circle(double _x, double _y, double _radius)
    {
        super(_x,_y);
        radius = _radius;
    }

    double area(){ return 3.1415 * radius * radius; }

    double radius;
};
```

The `instanceof` operator walks through the prototype chain to determine which constructors have been invoked. As a consequence, instances of a subclass are also considered to be instances of the base class:

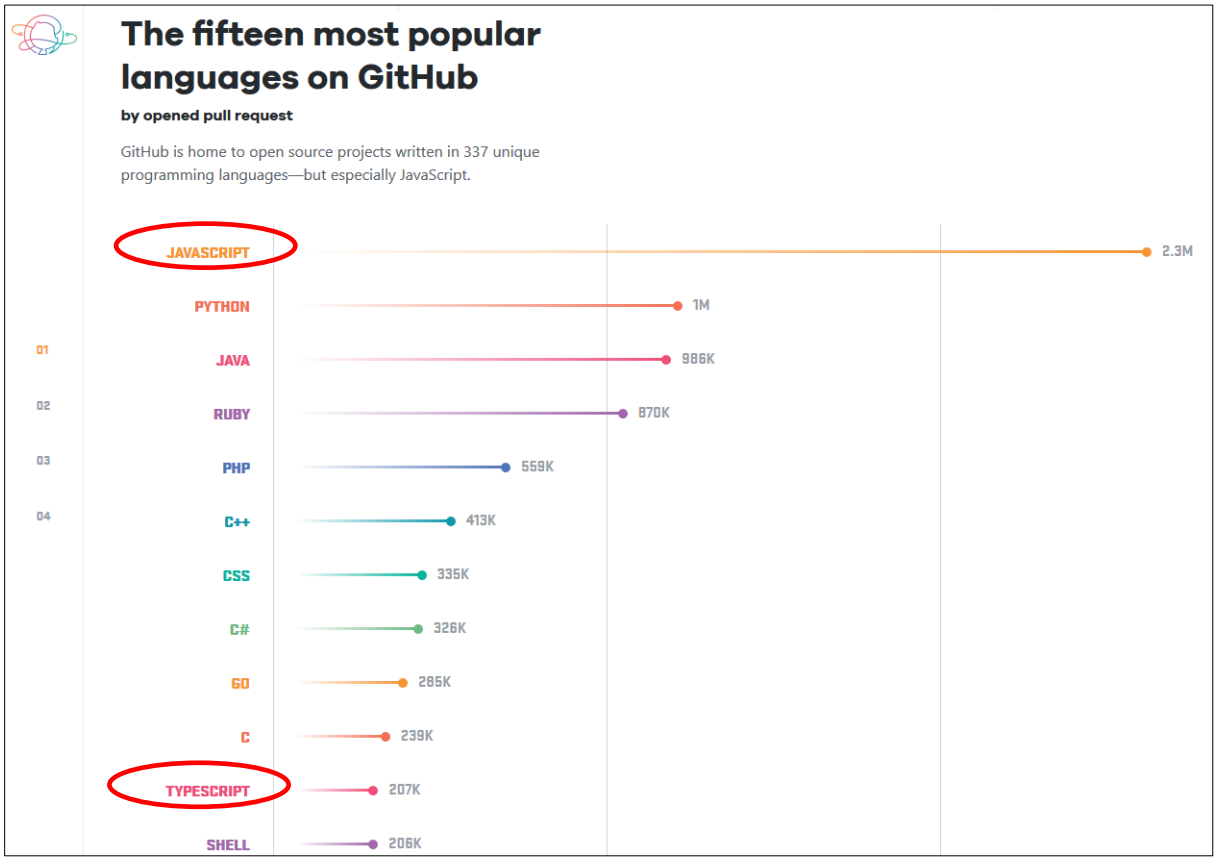
```
var circle = new Circle(0, 0, 50);
circle instanceof Circle;          // returns true
circle instanceof Shape;           // returns true
circle instanceof Object;          // returns true
circle instanceof Array;           // returns false
```

David Flanagan's "*JavaScript: The Definitive Guide (O'Reilly, 2006)*" is recommended both as a tutorial and as a reference manual.

11.3. Appendix C: JavaScript popularity

Following the website "https://octoverse.github.com/", the most popular programming languages are (May 2018):

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



JavaScript is on the 1st position, but also on the 11th position since Typescript is just a dialect of Javascript optimized to run outside a web browser.

From this Table, we can see that the JavaScript language is the most popular scripting language of all the **scripting** languages. "Scripting language" are languages designed so that everybody can use them: you don't need to be an experienced programmer to use a "Scripting language": nearly anybody can do it. "Scripting languages" are a lot easier to use than a "normal programming language" like C or C++.

Python, C++, C and R (the 4 other programming languages used inside Anatalla) have also a dominant position.

It seems that improving anyone's JavaScript-skills is a good investment for the future because this scripting-language is extremely popular & standard and it's "here to stay".

The popularity of Javascript is bound to increase even more over the forecoming years because Google decided to invest a huge amount of cash to improve the speed, portability and popularity of Javascript because Javascript is the main and only programming language behind the "Google Computer" (i.e. "Chromebook").

11.4. Appendix D: Is Anatella Fast?

The software design of Anatella has been carefully though to obtain one of the fastest ETL tool (maybe the fastest) on "simple commodity hardware".

There are currently three main different approaches when designing a new ETL tool based on "small graphical boxes". The main difference between these three approaches resides in the technique used

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

to “transfer” the data-stream from one “little box” to the next one. The common three different approaches are:

1. Description: The different boxes are communicating through a file on the harddrive. Each operator/box reads a file in input and writes a file as output.

   Advantages: Each operator can be a totally different executable. When developing the ETL tools, if you assign one developer per “executable/operator”, it’s very easy to find which developer to blame when an operator crashes. Thus the management of the development team that is developing the ETL tool is very easy. This approach allows a fast and cheap development of the ETL tool. This is, of course, a very common approach. Only low-grade ETL’s are using this approach.

   Disadvantages: this technique is very slow because the speed of any ETL software is usually (inversely) proportional to the number of disk I/O performed (at least for Analytical tasks). Since you need to write and read the whole data-tables between each box, it generates an enormous amount of I/O, leading to an extremely slow ETL.

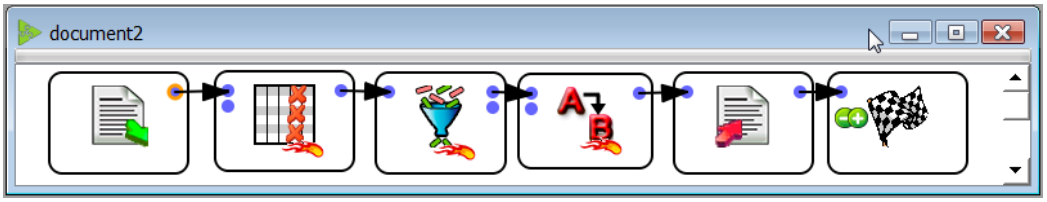
   Example of ETL tool using this approach: SAS enterprise guide (not SAS base), Amadea.

2. Description: Each box has its own thread. Each arrow between two boxes is a “queued buffer”. Each “thread/box” reads new “rows” on its input queue and writes “processed rows” on the output queue (See section 5.3.2.9. for more information about this “multithreaded” approach). This approach can be described as a “push” approach: each “box” processes a row and there after pushes it in the queue for the next box on the right. The classical consumer/producer problem occurs in each input/output queue. The queues must thus have a mechanism (usually based on semaphore and mutex locks) to ensure consistency between threads.

   Advantages: All operators are running in parallel, exploiting the multiple CPU’s available on your computer. By default, for simple “row operations”, there are no disk I/O to perform and thus this approach is far more efficient than the previous one.

   Disadvantages:
   - There are as many threads running in parallel as the number of “boxes” in your transformation script. The number of active thread inside the ETL tool can thus be quite large. In such situation, the main CPU of your PC will spend a large amount of time in “context switching”: it will lose time switching from one thread to the other. See section 5.3.2.9. for more information about this subject.
   - There exists a large number of “queued buffer” (one for each arrow in the transformation graph). For efficiency reason, each of these buffers must contain several hundreds of rows. If we are manipulating “big” rows (that contains thousands of columns), each of these buffers can requires as much as 100 MB each. This imposes a limit of 20 arrows in the graph in a “common PC hardware” (for example a windows XP 32 bit.) This approach can thus lead to serious memory limitations.
   - This approach requires to perform a deep copy of each Row in the output “queued buffer” that exists between each “box”. This means that for a simple script like this one:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu



... all the rows are "deep-copied" 5 times in the 5 "queued buffers". If we are manipulating "big" rows, this "deep-copy" requires a huge amount of CPU power and this slows down significantly the computations.

Example of ETL tool using this approach: CloverETL, Kettle.

3. Description: There is only one thread that runs the whole transformation graph. When a "box" requires a new row to perform its computation, it asks a row to the previous "box". This approach can be described as a "pull" approach: each "box" pulls a row from the "previous box on the left" and thereafter process it.

   Advantages: By default, for simple "row operations", there are no disk I/O to perform and thus this approach is very efficient. Since there exists only one unique thread to do the whole processing of the whole transformation graph, there is no need for "queued buffers" between each box. This approach is thus a lot less "memory hungry" compared to the previous one. Since there are no "queued buffers" at all, it means that we can also avoid to perform all the deep-copy of all the rows. Since there are no deep copy anymore, the stress on the CPU is greatly minimized and the CPU can be wittingly used to run the "useful" computations: i.e. the CPU is actually used to run the transformations taking place inside the different "boxes".

   Disadvantages: This approach does not directly use all the CPU's available on your computer. Thus, depending on the type of transformation, it can be slower than the previous approach. In particular, if you have several database extractions to perform, this approach is not efficient at all because it runs each extraction sequentially (instead of simultaneously for the previous approach). This approach is nevertheless far more efficient than the first one.

   Example of ETL tool using this approach: Old ETL tools.

4. Description: This last approach combines the best of the 2 previous approach: it allows multithreaded execution (where you have the exact control of how many CPU are used) and mono-threaded execution (so that each CPU runs at peak efficiency). Anatella uses this approach. Inside Anatella, you have:
   - All the Actions inside the same Multithreaded section run on 1 CPU (i.e. we have mono-threaded execution)
   - We have also multithreaded execution because each Multithreaded section (defined using the Multithread Action) runs on a different CPU.

   Advantages: All advantages of the two previous approaches.

   Disadvantages: This approach does not directly use all the CPU's available on your computer. To use all the available CPU's, you must manually add some Multithread

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Actions (or run several different transformations graphs "in parallel", using the ParallelRun Action). Usually, the real speed bottleneck in Anatella is the CPU (all the disk I/O operations are very efficient inside Anatella). To remove this bottleneck, use carefully some Multithread Actions. The proper usage of the Multithread Actions can very often multiply the speed of the data-transformation process by 8 on simple commodity hardware.

Example of ETL tool using this approach: Anatella (and no other!)

Anatella is one of the very few modern "ETL tools" developed in C. The recent C compilers (like the excellent Microsoft Visual Studio) are astonishing: they are producing executables that usually runs 2 orders of magnitudes faster than any other programming language (except maybe Fortran). This means that all the transformation scripts based on the C/C++ operators available in Anatella are unmatched in terms of execution speed. The execution speed of the operators based on the "JavaScripts language" (similar to JavaScript) is also extremely high, thanks to the usage of a highly tuned JIT (just-in-time compiler) named "SquirrelFish Extreme". The speed of the operators based on "JavaScripts language" is still unfortunately largely inferior to the C operators (but the research in the field of "JavaScript compilers" is still very active and an improved version of the "SquirrelFish Extreme" JIT might appear very soon).

We have invested a great deal of time designing and developing one of the most flexible and fastest data transformation tool available on the market. Indeed the processing speed reached by Anatella on simple commodity hardware is impressive. Thanks to a unique software design, Anatella is able to flawlessly process in a few seconds extremely large tables containing thousands of columns. This ability is vital when working on analytical tasks, such as predictive datamining.

11.5. Appendix E: The standard functionalities offered by any JavaScript /ECMAScript compliant engine

The next pages list the built-in global constants, functions, and objects provided by any standard JavaScript compliant language. In the next appendix (section 11.6), we will see the built-in functionalities that are specific to Anatella.

Constants of the global object	
NaN	IEEE 754 Not-a-Number (NaN) value
Infinity	Positive infinity (+∞)
undefined	Default value for uninitialized variables

Functions of the global object	
print(x)	Prints a value in the Anatella log window
eval(str)	Executes an JavaScript program
parseInt(str, base)	Converts a string to an integer value
parseFloat(str)	Converts a string to a floating-point value
isNaN(n)	Returns true if n is NaN
isFinite(n)	Returns true if n is a number other than NaN, +∞, or -∞
decodeURI(str)	Converts an 8-bit-encoded URI to Unicode

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Functions of the global object	
decodeURIComponent(str)	Converts an 8-bit-encoded URI component to Unicode
encodeURI(str)	Converts a Unicode URI to an 8-bit-encoded URI
encodeURIComponent(str)	Converts a Unicode URI component to 8-bit-encoded
Escape()	Encodes a string
Unescape()	Decodes an encoded string
Number()	Converts an object's value to a number
String()	Converts an object's value to a string

Classes (Constructors)	
Object	Provides functionality common to all objects
Function	Encapsulates an JavaScript function
Array	Represents a resizable vector of items
String	Stores a Unicode string
Boolean	Stores a Boolean value (true or false)
Number	Stores a floating-point number. Provide conversion between Strings and the primitive numeric value.
Date	Stores a date and time
RegExp	Provides regular expression pattern matching
Error	Base type for error types
EvalError	Raised when using eval() wrongly
RangeError	Raised when a numeric value is outside the legal range
ReferenceError	Raised when trying to access an undefined variable
SyntaxError	Raised when a syntax error is detected by eval()
TypeError	Raised when an argument has the wrong type
URIError	Raised when URI parsing fails

Static Object (always accessible)	
Math	Provides mathematical constants and functions

Properties of the Math Object	
E	Returns Euler's number (approx. 2.718)
LN2	Returns the natural logarithm of 2 (approx. 0.693)
LN10	Returns the natural logarithm of 10 (approx. 2.302)
LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
PI	Returns PI (approx. 3.14159)
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
SQRT2	Returns the square root of 2 (approx. 1.414)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the Math Object	
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
asin(x)	Returns the arcsine of x, in radians
atan(x)	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
atan2(y,x)	Returns the arctangent of the quotient of its arguments
ceil(x)	Returns x, rounded upwards to the nearest integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of E^x
floor(x)	Returns x, rounded downwards to the nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x,y,z,…,n)	Returns the number with the highest value
min(x,y,z,…,n)	Returns the number with the lowest value
pow(x,y)	Returns the value of x to the power of y
random	Returns a random number between 0 and 1
round(x)	Rounds x to the nearest integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle

Properties of the Number Class	
MAX_VALUE	Returns the largest number possible in JavaScript
MIN_VALUE	Returns the smallest number possible in JavaScript
NEGATIVE_INFINITY	Represents negative infinity (returned on overflow)
POSITIVE_INFINITY	Represents infinity (returned on overflow)

Methods of the Number Class	
toExponential(x)	Converts a number into an exponential notation
toFixed(x)	Formats a number with x numbers of digits after the decimal point
toprecision(x)	Formats a number to x length
toString()	Converts a Number object to a string
valueOf()	Returns the primitive value of a Number object

Methods of the Date Class	
getDate()	Returns the day of the month (from 1-31)
getDay()	Returns the day of the week (from 0-6)
getFullYear()	Returns the year (four digits)
getHours()	Returns the hour (from 0-23)
getMilliseconds()	Returns the milliseconds (from 0-999)
getMinutes()	Returns the minutes (from 0-59)
getMonth()	Returns the month (from 0-11)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

getSeconds()	Returns the seconds (from 0-59)
getTime()	Returns the number of milliseconds since midnight Jan 1, 1970
getTimezoneOffset()	Returns the time difference between GMT and local time, in minutes
getUTCDate()	Returns the day of the month, according to universal time (from 1-31)
getUTCDay()	Returns the day of the week, according to universal time (from 0-6)
getUTCFullYear()	Returns the year, according to universal time (four digits)
getUTCHours()	Returns the hour, according to universal time (from 0-23)
getUTCMilliseconds()	Returns the milliseconds, according to universal time (from 0-999)
getUTCMinutes()	Returns the minutes, according to universal time (from 0-59)
getUTCMonth()	Returns the month, according to universal time (from 0-11)
getUTCSeconds()	Returns the seconds, according to universal time (from 0-59)
getYear()	Deprecated. Use the getFullYear() method instead
parse()	Parses a date string and returns the number of milliseconds since midnight of January 1, 1970
setDate()	Sets the day of the month (from 1-31)
setFullYear()	Sets the year (four digits)
setHours()	Sets the hour (from 0-23)
setMilliseconds()	Sets the milliseconds (from 0-999)
setMinutes()	Set the minutes (from 0-59)
setMonth()	Sets the month (from 0-11)
setSeconds()	Sets the seconds (from 0-59)
setTime()	Sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970
setUTCDate()	Sets the day of the month, according to universal time (from 1-31)
setUTCFullYear()	Sets the year, according to universal time (four digits)
setUTCHours()	Sets the hour, according to universal time (from 0-23)
setUTCMilliseconds()	Sets the milliseconds, according to universal time (from 0-999)
setUTCMinutes()	Set the minutes, according to universal time (from 0-59)
setUTCMonth()	Sets the month, according to universal time (from 0-11)
setUTCSeconds()	Set the seconds, according to universal time (from 0-59)
setYear()	Deprecated. Use the setFullYear() method instead
toDateString()	Converts the date portion of a Date object into a readable string
toGMTString()	Deprecated. Use the toUTCString() method instead
toLocaleDateString()	Returns the date portion of a Date object as a string, using locale conventions
toLocaleTimeString()	Returns the time portion of a Date object as a string, using locale conventions
toLocaleString()	Converts a Date object to a string, using locale conventions
toString()	Converts a Date object to a string
toTimeString()	Converts the time portion of a Date object to a string
toUTCString()	Converts a Date object to a string, according to universal time
UTC()	Returns the number of milliseconds in a date string since midnight of January 1, 1970, according to universal time
valueOf()	Returns the primitive value of a Date object

Properties of the Array Class	
length	Sets or returns the number of elements in an array

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the Array class	
concat()	Joins two or more arrays, and returns a copy of the joined arrays
join()	Joins all elements of an array into a string
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes the first element of an array, and returns that element
slice()	Selects a part of an array, and returns the new array
sort()	Sorts the elements of an array
splice()	Adds/Removes elements from an array
toString()	Converts an array to a string, and returns the result
unshift()	Adds new elements to the beginning of an array, and returns the new length
valueOf()	Returns the primitive value of an array

Properties of the String Class	
length	Sets or returns the number of elements in an array

Methods of the String class	
charAt()	Returns the character at the specified index
charCodeAt()	Returns the Unicode of the character at the specified index
concat()	Joins two or more strings, and returns a copy of the joined strings
fromCharCode()	Converts Unicode values to characters
indexOf()	Returns the position of the first found occurrence of a specified value in a string
lastIndexOf()	Returns the position of the last found occurrence of a specified value in a string
match()	Searches for a match between a regular expression and a string, and returns the matches
replace()	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring
search()	Searches for a match between a regular expression and a string, and returns the position of the match
slice()	Extracts a part of a string and returns a new string
split()	Splits a string into an array of substrings
substr()	Extracts the characters from a string, beginning at a specified start position, and through the specified number of character
substring()	Extracts the characters from a string, between two specified indices
toLowerCase()	Converts a string to lowercase letters
toUpperCase()	Converts a string to uppercase letters
valueOf()	Returns the primitive value of a String object

Properties of the RegExp Class	
global	Specifies if the "g" modifier is set
ignoreCase	Specifies if the "i" modifier is set
lastIndex	The index at which to start the next match

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Properties of the RegExp Class	
multiline	Specifies if the "m" modifier is set
source	The text of the RegExp pattern

Methods of the RegExp Class	
compile()	Compiles a regular expression
exec()	Tests for a match in a string. Returns a result array
test()	Tests for a match in a string. Returns true or false

11.6. Appendix F: Anatella-specific JavaScript extensions

In the previous Appendix (Appendix E), we saw all the standard constants, functions, and objects provided by any standard JavaScript/ECMAScript compliant language.

Anatella includes a special set of functions and objects that allows you to easily manipulate row data, row meta-data, input pins & output pins. These functions and objects are:

Global Anatella variables	
nPinIn	The number of input pin of the Action.
nPinOut	The number of output pin of the Action.
isVariableNPinIn	Is the number of input pin of the Action variable?
isVariableNPinOut	Is the number of output pin of the Action variable?
anatellaDir	The directory where the “anatella.exe” file is installed. This is usually “c:\program files\TIMi\bin”.
tempDir	The directory where the temporary HD Cache files are created. This is usually “c:\Users\my_name\AppData\Local\Temp”.
currentDir	The directory where the current .anatella file is located.
currentFile	The complete filepath (dir+name) of the current .anatella file.
actionID	A number that uniquely identifies this action inside the Anatella graph

Commonly–used Anatella functions of the global object	
getCurrentRow(inputPinIndex)	Returns a “row object” that represents the ‘current’ row on the given input pin
getNextRow(inputPinIndex)	Extracts a new row from the given input pin and return it as a new “row object”
setOutputRowSize(outputPinIndex, number_of_columns)	Defines the number of columns on the given output pin
writeEOL(outputPinIndex)	Write the “End-Of-Line” marker on the given output pin
rowDeepCopy(rowObject)	Creates a deep copy of the given Row Object. See section 5.2.5 for more information on the proper usage of this function
rowGetIndexOfColumn(rowObject, string)	Returns the column index inside the given row whose name is given as second parameter.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella global-functions related to writing on the output pin	
rowWriteConstant(rowObject, outputPinIndex, string)	Replace the content of all columns of the row object with the given string and write the modified row on the given output pin. The rowObject given as first parameter is not modified.
rowWriteNull(rowObject, outputPinIndex)	Replace the content of all columns of the row object with NULL values and write the modified row on the given output pin. The rowObject given as first parameter is not modified. This function works even if the "isNull" property of the rowObject given as first parameter is true.

Anatella global-functions related to column selection on the output pin	
dropAllColumns(outputPinIndex)	All the columns written on the given output pin will be dropped. The output pin will emit a "void" table.
keepAllColumns(outputPinIndex)	All the columns written on the given output pin will be "passed" to the next Action.
dropColumn(outputPinIndex, columnIndex)	The given column index on the given output pin won't be passed to the next Action.
keepColumn(outputPinIndex, columnIndex)	The given column index on the given output pin will be passed to the next Action.

Variables of the Anatella Row class	
nColumn	The number of columns contained inside this row object.
isNull	Boolean value that is used to signal the end-of-input-table condition on a specific input pin when using the "*getNextRow()*" function.

Methods of the Anatella Row class	
write(outputPinIndex)	Write the row on the given output pin
col(columnIndex)	Returns a string that is the content of the column with the index "columnIndex" inside the "this" row object.
setColumn(columnIndex,string)	Set the content of a given column inside the "this" row object.

Anatella global-functions related to row content management	
rowSetAllColumns(rowObject, string)	The content of all the columns of the given row Object is set to the given string.

Anatella global-functions related to meta-data management	
rowGetColumnName(rowObject, columnIndex)	Return the required column name
rowSetColumnName(rowObject, columnIndex, String)	Set the column name of a specific column

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella global-functions related to meta-data management	
rowGetMetaType(rowObject, columnIndex)	Return the meta-type of the given column. This meta-type is character: • 'A': column is Alpha-numeric • '0': column is numeric • 'D': column is a date • 'B': column is a boolean • 'I': column is an image • 'S': column is a sound • 'U': unknown meta-type • 'E': the columnIndex is erroneous
rowSetMetaType(rowObject, columnIndex, string)	Set the meta-type of the given column
rowGetDateFormat(rowObject, columnIndex)	Get the date format of the given column
rowSetDateFormat(rowObject, columnIndex, string)	Set the date format of the given column
rowGetSortColumnIdx(rowObject, sortPriority)	Return the indexes of the columns on which the table is sorted. For example, if the table is sorted on two columns, then "*rowGetSortColumnIdx(r,0)*" and "*rowGetSortColumnIdx(r,1)*" are returning the required indexes. On the same example, a call to the "*rowGetSortColumnIdx(r,2)*" returns -1.
rowGetSortType(rowObject, sortPriority)	Returns the sort type of the table. This sort type is a character: a) 'A': Alphabetical order (A..Za..z) b) 'Z': Inverse Alphabetical order (z..aZ..A) c) 'a': Alphabetical order - case insensitive(Aa..Zz) d) 'z': Inverse Alphabetical order - case insensitive(Zz..Aa) e) '0': Numerical order increasing (0..9) f) '9': Numerical order decreasing (9..0) g) 'I': Date (chronological order) h) 'D': Date (inverse chronological order) i) 'E': error: undefined

Anatella global-functions related to debugging	
rowToArray(rowObject)	Return an array of strings that matches the content of the given row object.
rowToObject(rowObject)	Return a pure JavaScript Object that has as many properties as the number of column of the given row object. The properties names of the returned pure JavaScript Object are base on the column names of the given row Object. The properties are initialized based on the content of the columns of the given row Object.
printRow (rowObject)	Print the given row object inside the JavaScript debug console.
clearLog()	Clear the Log window (this does not affect the trace files)

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella global-functions related to Files	

dirList(directory,
file_filter,
sort_mode,
dir_filter)
	Return an array of objects. Each element of the array contains information about a different file or directory. The parameters *sort_mode* and *dir_filter* are optional. If omitted, Anatella will use:
	sort_mode= DirSort_ByName+DirSort_IgnoreCase
	dir_filter= DirFilter_AllEntries.
	
	For example: This will print inside the log window the complete file path to the first .anatella file located inside the current directory:
	`var d=dirList(":/","*.anatella");`
	`print(d[0].absolutePath+"/"+d[0].fileName);`
	
	The fields of the different elements of the returned array are:
	• "*absolutePath*": self-explanatory (this does not contain the filename: only the path).
	• "*fileName*": self-explanatory.
	• "*created*": the creation date (as a JavaScript date Object).
	• "modified": the last modification date (as a JavaScript date Object).
	• "isDir": self-explanatory.
	• "isFile": self-explanatory.
	• "isHidden": self-explanatory.
	• "isReadable": self-explanatory.
	• "isWritable": self-explanatory.
	• "isExecutable": self-explanatory.
	
	The *sort_mode* parameter is a combination of the following flags:

Constant	Value	Description
DirSort_ByName	0x00	Sort by name.
DirSort_ByTime	0x01	Sort by time (modification time).
DirSort_BySize	0x02	Sort by file size.
DirSort_ByExtention	0x80	Sort by file type (extension).
DirSort_DoNotSort	0x03	Do not sort.
DirSort_DirsFirst	0x04	Put the directories first, then the files.
DirSort_DirsLast	0x20	Put the files first, then the directories.
DirSort_Reversed	0x08	Reverse the sort order.
DirSort_IgnoreCase	0x10	Sort case-insensitively.
DirSort_LocaleAware	0x40	Sort items appropriately using the current locale settings.

The *dir_filter* parameter is a combination of the following flags:

Constant	Value	Description
DirFilter_Dirs	0x001	List directories that match the filters.
DirFilter_AllDirs	0x400	List all directories; i.e. don't apply the filters to directory names.
DirFilter_Files	0x002	List files.
DirFilter_Drives	0x004	List disk drives (ignored under Unix).

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

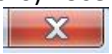
Anatella global-functions related to Files	

	DirFilter_NoSymLinks	0x008	Do not list symbolic links (ignored by operating systems that don't support symbolic links).
	DirFilter_NoDotAndDotDot	0x1000	Do not list the special entries "." and "..".
	DirFilter_NoDot	0x2000	Do not list the special entry ".".
	DirFilter_NoDotDot	0x4000	Do not list the special entry "..".
	DirFilter_AllEntries	Dirs | Files | Drives	List directories, files, drives and symlinks (this does not list broken symlinks unless you specify System).
	DirFilter_Readable	0x010	List files for which the application has read access. The Readable value needs to be combined with Dirs or Files.
	DirFilter_Writable	0x020	List files for which the application has write access. The Writable value needs to be combined with Dirs or Files.
	DirFilter_Executable	0x040	List files for which the application has execute access. The Executable value needs to be combined with Dirs or Files.
	DirFilter_Modified	0x080	Only list files that have been modified (ignored on Unix).
	DirFilter_Hidden	0x100	List hidden files (on Unix, files starting with a ".").
	DirFilter_System	0x200	List system files (on Unix, FIFOs, sockets and device files are included; on Windows, .lnk files are included)
	DirFilter_CaseSensitive	0x800	The filter should be case sensitive.

fileExist(filename)	Test if a file or a directory exists. Returns an integer number: • "0": no file or directory with the given name exists. • "1": a file with the given name exists. • "2": a directory with the given name exists.
fileMove (source, destination, alsoMoveDirs)	Move some file or directories. The parameter *alsoMoveDirs* is optional. If omitted, Anatella wil use *alsoMoveDirs*=false. If you intend to move a directory (instead of a file), you must set *alsoMoveDirs*=true. Returns true if successful.
fileCopy(source, destination alsoCopyDirs)	Copy some file or directories. The parameter *alsoCopyDirs* is optional. If omitted, Anatella wil use *alsoCopyDirs*=false. If you intend to copy a directory (instead of a file), you must set *alsoCopyDirs*=true. Returns true if successful.
fileDelete(filename, forceDelete)	Delete some file or directory. The parameter *forceDelete* is optional. If omitted, Anatella wil use *forceDelete* =false. If you intend to delete a directory or a read-only file, you must set *forceDelete*=true. Returns true if successful.
fileLoad(filename, encoding)	Returns a string with the content of the file *filename*. If the optional parameter *encoding* is omitted, the text encoding inside the file is assumed to be the local system-wide encoding, otherwise: • encoding=0: text is in utf-16 • encoding=1: text is in utf-8 If the file contains a BOM, the text-encoding specified inside the BOM is always used.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella global-functions related to Files	
fileBinaryLoad(fileName, startOffset, length)	Returns an object that contains the binary content of the file *filename*. The parameters *startOffset* and *length* are optional. If there are omitted, Anatella will use *startOffset*=0 and *length*=the size of the loaded file.
fileSave(fileName, fileContent, encoding)	Save the string *fileContent* inside the text file *filename*. The text-encoding of the saved file is given inside the optional *encoding* parameter (when the *encoding* parameter is omitted, Anatella assumes: *encoding*=0): • encoding=0: file is in utf-16 (with BOM) • encoding=1: file is in utf-8 (with BOM) • encoding=2: file is in utf-8 (without BOM) • encoding=3: file is in local system-wide 8-bit encoding (without BOM)
mkDir(path)	Create a new directory. This will create all parent directories necessary to create the directory. Returns true if successful.

Anatella global-functions : Other	
print(string)	Print the string inside the Antella log window.
exit(exitcode)	Abort immediately the graph execution and stop the whole Anatella process. If Anatella was busy writing to a file, the file can be truncated because of the abrupt "abort". The parameter *exitcode* is optional (default value is 0). Use only this function with extreme cautions: It's equivalent to a click on the  button in the top-right corner of the Anatella window.
trim(string)	Returns a string that has whitespace removed from the start and the end. Whitespace characters include the ASCII characters '\t', '\n', '\v', '\f', '\r', and ' '.
DateUTC()	Return the current DateTime in the UTC time frame
getTimeOfDay(datetime)	Returns the "time" part of the JavaScript date object "*datetime*".
getDayMonthYear(datetime)	Returns the "date" part of the JavaScript date object "*datetime*".
weekNumber(datetime)	Returns the week number (1 to 53) from the JavaScript date object "*datetime*". Returns 0 if the date is invalid. In accordance with ISO 8601, weeks start on Monday and the first Thursday of a year is always in week 1 of that year. Most years have 52 weeks, but some have 53.
nDaysInMonth(datetime) nDaysInMonth(month) nDaysInMonth(month,year)	Returns the number of days in month. (1<=month<=12)
dateAdd(datetime,y,M,d)	Add a duration (in year, month and days) to a date
dateDiffInDay(datetime1, datetime2)	Number of days between datetime1 and datetime2
dateDiffInMonth(datetime1, datetime2)	Number of months between datetime1 and datetime2
dateDiffInYear(datetime1, datetime2)	Number of years between datetime1 and datetime2
nearUniqueRandomID()	Generate a random number based on the current date, current time, current milliseconds and the current processuss ID. This number can be used as a primary key to uniquely identify the current processus and time.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella global-functions : Other	
getEnv(string)	Return the value of the specified DOS environment variable
urlEncode(string)	Encode a string so that it can be used inside a parameter inside an url **LIB:** MathParser
jsonEncode(string)	Encode a string so that it can be used inside a JSON file **LIB:** MathParser
xmlEncode(string)	Encode a string so that it can be used inside a XML file **LIB:** MathParser
sleep(number)	The Anatella process waits for "number" milliseconds.
clearLog()	Clear the log Anatella Window
isNumber(x,strict)	Test is x is a string that can be converted to a number The second paremeter "strict" is optional: The default value is True.
toNumber(x)	Convert the string x to a Number using Anatella advanced routines (that accepts NaN, NULL, Inf, -Inf and any extraneous characters after the number).
isLinux()	Returns true if Anatella is running inside Wine inside Linux. The test is actually based on the presence/abscence of the registry key: HKEY_CURRENT_USER\Software\Wine
commandLineToArray(s)	Converts the string parameters s to a Javascript array. The parsing rules are the same as in a Windows command-line prompt.

Anatella constructor (new) functions	
Row(number_of_columns)	Creates a new Row Object with the given number of columns. You can use the newly created Row Object to write to any output pin.
PredictiveModel (filename, rowObject)	Creates a new TIMi Predictive Model based on the TIMi model file (*.ModelXML) given as first parameter. The model can only be applied on row objects similar to the row object given as second parameter to this constructor (by similar, we mean the column positions are the same).
SegmentationModel(filename, rowObject)	Creates a new Stardust Segmentation Model based on the Satrdust segmentation model file (*.SModelXML) given as first parameter. The model can only be applied on row objects similar to the row object given as second parameter to this constructor (by similar, we mean the column positions are the same).

DateTool(Format string)	This creates a DateTool object that can be used to produce
	a) JavaScript Date objects from strings containing formatted dates.
	b) Strings from JavaScript date objects.
	The syntax of the format string is explained in section 5.1.3.
	
	Here is an example of usage:
	`var dateString1="5-3-2012";`
	`var dateString2="4-7-2010";`
	`var dt=DateTool("d-M-yyyy");`
	`var dateJS1=dt.parse(dateString1);`
	`var dateJS2=dt.parse(dateString2);`
	`var dateReformatted1=DateTool("yyyyMMdd").print(dateJS1);`
	`var dateReformatted2=DateTool("yy/MM/dd hh:mm:ss").print(dateJS2);`
	`print(dateReformatted1);`
	`print(dateReformatted2);`

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

<table>
<tr><th colspan="2">Anatella constructor (new) functions</th></tr>
<tr><td></td><td>This prints in the console:
20120305
10/07/04 00:00:00</td></tr>
<tr><td>ETTool(Unit,ReferenceDate)</td><td>This creates an ETTool object that can be used to manipulate “Elapsed Times”
The “Unit” parameter can have the following value:
<table><tr><th>Constant</th><th>Value</th></tr><tr><td>ETUnit_second</td><td>0x00</td></tr><tr><td>ETUnit_minute</td><td>0x01</td></tr><tr><td>ETUnit_hour</td><td>0x02</td></tr><tr><td>ETUnit_day</td><td>0x03</td></tr><tr><td>ETUnit_week</td><td>0x04</td></tr></table>
The “ReferenceDate” parameter is a date given in the “yyyyMMdd hh:mm:ss” format.</td></tr>
<tr><td>SQLite(DBFile, openMode, timeOutInMs)</td><td><u>LIB:</u> DBConnectors

This create a SQLite object that can be used to manipulate “SQLite databases”.

The “DBFile” parameter is the complete path to the SQLite database (don’t use relative paths “:/”).

The “openMode” parameter can have the following value:
<table><tr><th>Constant</th><th>Value</th></tr><tr><td>SQLite_ReadOnly</td><td>0x01</td></tr><tr><td>SQLite_ReadWrite</td><td>0x02</td></tr><tr><td>SQLite_ReadWriteCreate</td><td>0x06</td></tr></table>
The “timeOutInMs” parameter is optional: If omitted, Anatella will use “timeOutInMs”=10000. When you try to write to a locked table, the SQLite engine will wait, for at most “timeOutInMs” milliseconds, until the database/table unlocks (to be able to execute your command). After the timeout, if the database/table is still locked, the execution of your SQL command fails.</td></tr>
<tr><td>ProcessRunner(nRetry, randomWaitBeforeRun)</td><td>This creates a ProcessRunner object that is used to run external executables from Anatella.

The 2 parameters nRetry and randomWaitBeforeRun and are optional: If omitted, Anatella will use nRetry=5 and randomWaitBeforeRun=1000.

These 2 parameters are useful when launching the execution of a large list of different executables (using the “run()” method):
When an executable from the list fails, Anatella can, either: abort, ignore or re-try (i.e. “re-try”=re-execute the same executable once again). The number of re-try is defined as the nRetry parameter.
Before running any executable from the list, Anatella waits for a random number of milliseconds. The duration of the “wait” is in-between 0 milliseconds and randomWaitBeforeRun milliseconds.</td></tr>
</table>

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella constructor (new) functions	

There are 2 different ways of using a ProcessRunner object: you can, either:

- Create a list of executable to run and thereafter run the list:
  a) use the methods "*addJob()*" or "*addAnatellaJob()*" to add a executables to the list.
  (optionally: you can use the method "*listJobs()*" to view the list)
  b) use the method "*run()*" to start the execution of all the executables inside the list.
- Run directly an executable using the methods "*runImmediate()*" or "*runAnatella()*" (without going through the process of creating a list first).

When you click the STOP button, Anatella automatically aborts all the child processes that were running.
Here are some examples:

Example 1.

```
var p=new ProcessRunner();
p.runImmediate("aggregate_products.anatella","-DfileIndex=1");
p.runImmediate("aggregate_customers.anatella","-DfileIndex=1");
```

Or, alternatively:

```
var p=new ProcessRunner();
p.runImmediate( "\""+anatellaDir+"/anatella.exe\" -r aggregate_products.anatella  -DfileIndex=1");
p.runImmediate( "\""+anatellaDir+"/anatella.exe\" -r aggregate_customers.anatella -DfileIndex=1");
```

This runs the 2 Anatella-graphs '"aggregate_products.anatella" and "aggregate_customers.anatella" sequentially (i.e. one after the other). The value of Graph-Global-Parameter named "fileIndex" is set to "1". See section 4.7. for more information about the parameters that you can use when running Anatella from the command-line.

Example 2.

```
var p=new ProcessRunner();
p.addAnatellaJob("aggregate_products.anatella","-DfileIndex=1");
p.addAnatellaJob("aggregate_customers.anatella","-DfileIndex=1");
p.run(2);
```

Or, alternatively:

```
var p=new ProcessRunner();
p.addJob( "\""+anatellaDir+"/anatella.exe\" -r aggregate_products.anatella  -DfileIndex=1");
p.addJob( "\""+anatellaDir+"/anatella.exe\" -r aggregate_customers.anatella -DfileIndex=1");
p.run(2);
```

This runs the 2 same Anatella-graphs as in the previous example. However, this time, the 2 graphs are running simultaneously (i.e. in parallel). This allows us to easily use all the several CPUs available in modern computers.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella constructor (new) functions	

	Example 3.
	
	`var p=new ProcessRunner();`
	`var nFiles=100;`
	`for(i=1;i<=nFiles;i++)`
	`{`
	`  p.addAnatellaJob("sort_one_day.anatella","-DfileIndex="+i);`
	`}`
	`p.run(8);`
	
	Or, alternatively:
	
	`var p=new ProcessRunner();`
	`var nFiles=100;`
	`for(var i=1; i<=nFiles; i++)`
	`{`
	`  p.addJob( "\""+anatellaDir+"/anatella.exe\" -r sort_one_day.anatella -DdayIndex="+i);`
	`}`
	`p.run(8);`
	
	This runs 100 times the anatella-graph "sort_one_day.anatella" with different values of the Graph-Global-Parameter named "dayIndex". The objective is to sort 100 files (and each of these files contains one day of data). Each run of the "sort_one_day.anatella" graph will sort exactly one day: the day that is defined using the "dayIndex" variable.
	
	Anatella will run in parallel 8 data-transformation-graphs (because we wrote: "*p.run(8)*"). As soon as one of the 8 running Anatella-Graphs is finished, Anatella directly launch the next Anatella-Graph, so that there are always 8 graphs running at the same time. This allows us to use all the several CPUs available in modern computers at maximum efficiency.
	
	Before sorting one day (i.e. before adding one specific day to the list of jobs to run), it might be a good idea to use the function "*fileExist()*" to check if there already exists a file that contains the result of the sort (in such a situation, no need to sort the data because the resulting sorted file is already there!).

KeepBests(number of pairs)	An object that stores a limited number of (key,value) pairs: • Keys are floating point numbers (i.e. double) • Values are integers. This objects only stores the (key,value) pairs with the highest key values. All the pairs that have a "small key" are discarded.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella constructor (new) functions	

	For example:
	
	`function init(){ return 0;}`
	
	`function printKB(kb)`
	`{`
	`  var s="output: "+kb.getValue(0);`
	`  for(i=1;i<kb.n;i++)  s=s+","+kb.getValue(i);`
	`  print(s);`
	`}`
	
	`function run()`
	`{`
	`  var kb=new KeepBests(2);`
	`  kb.add(12,2);`
	`  kb.add(11,1);`
	`  printKB(kb);  // output : 2,1`
	`  kb.add(14,4);`
	`  printKB(kb);  // output : 4,2`
	`  kb.add(13,3);`
	`  printKB(kb);  // output : 4,3`
	`  return 1;`
	`}`

SendMail(smtpServer, fromAddress, encryption, portServer, connectionTimeOut)	**LIB:** EMailSMS
	
	Use this Javascript class to send emails.
	
	The parameters *encryption*, *portServer* and *connectionTimeOut* are optional: If omitted, Anatella will use these default values:
	• encryption= SendMail_SSL
	• portServer:
	  ○ if encryption is "SendMail_NoEncryption", PortServer=25
	  ○ if encryption is "SendMail_SSL", PortServer=465
	  ○ if encryption is "SendMail_StartTLS", PortServer=587
	• connectionTimeOut=5000 milliseconds
	
	The *encryption* parameter is one the following flags:
	
	Constant / Value / Description
	SendMail_NoEncryption / 0x00 / No encryption.
	SendMail_SSL / 0x01 / Mails are encrypted using SSL.
	SendMail_StartTLS / 0x02 / Mails are sent using the encrypted StartTLS protocol.
	
	Here is an example of usage:
	`var sm=new SendMail("ssl0.ovh.net", "anatella@timi.eu",SendMail_SSL, 465);`
	`sm.setLoginPassword("bob@business-insight.com",password);`
	`sm.send("bob@timi.eu","test","coucou message");`
	`sm.close();`

AnatellaGraph(file,login, password)	This create an object that represents an Anatella Graph. This object allows you to modify all the options of the Anatella Graph (i.e. all the security options, the "*Working directory for cache files*" option, the character encoding option). The "*login*" and "*password*" parameters are optional: They are only required if your Anatella graph is encrypted. For example:

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella constructor (new) functions	

	``
	var a=AnatellaGraph(":/testIn.anatella","my_login","my_password");
	if (a==null) return 1; // There was an error loading the graph:
	// maybe the (login, password) is not right?
	
	a.addUser("user1");
	a.setUserAdmin("user1",false);
	a.setUserEditGraph("user1",false);
	a.setUserExpirationDate("user1","20160101");
	a.setEncryption(true);
	
	if (a.hasChanged()) a.save(":/testOut.anatella");
	``
	The above example:
	1. Open an Anatella Graph named ":/testIn.anatella"
	2. Add a new user named "user1"
	3. The "user1" is not an admin and he can't edit the graph (he can only run it)
	4. The "user1" is an authorized user up to Januray 1st, 2016.
	5. Set the Graph as encrypted.
	6. Save the new Graph as ":/testOut.anatella"

Variables of the PredictiveModel class	
targetType	targetType is either 'T' (for binary target) or 'C' (for continuous target)
AUCTest, AUCFull AUCtopTest, AUCtopFull	Different measures of the quality of the Binary predictive model. These measures are only available for Binary predictive models.
RSquareTest, RSquareFull MAETest, MAEFull RMSETest, RMSEFull	Different measures of the quality of the Continuous predictive model. These measures are only available for Continuous predictive models.

Methods of the PredictiveModel class	
eval(rowObject)	Uses the TIM predictive model to compute a prediction based on the content of the given row. This method returns a floating point.

Variable of the SegmentationModel class	
segmentNames	This is an array-of-strings. It contains the different segment names.

Methods of the SegmentationModel class	

eval(rowObject)	This returns an integer number that is the number (the index) of the segment.
	Here is an example how to obtain the name of the segment:
	``
	var m=new SegmentationModel(":/mymodel.SModelXML",r);
	var currentSegmentAsInt=m.eval(r);
	var currentSegmentAsString=m.segmentNames[m.eval(r)];
	``

evalWidthdist(rowObject)	This returns an object that contains the following fields: idx: the number (the index) of the segment. segment: the name of the segment. dist: the distance between the current rowObject and the segment center.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the DateTool class	
parse(dateAsString)	Uses the DateTool object to return a JavaScript Date object constructed from the string given as parameter.
print(myDate)	Uses the DateTool object to return a string from the JavaScript date object given as parameter.

Methods of the ETTool class	
add(dateAsET,y,M,d)	add a duration (in year, month and days) to an Elapsed Time
diffInMonth(ET1,ET2)	number of months between ET1 and ET2
diffInYear(ET1,ET2)	number of years between ET1 and ET2
toDate(ET)	returns a Javascript date object representing the given ET
fromDate(dateAsString)	returns an Elapsed Time computed from a Javascript date object
toString(ET,dateFormat)	Returns a dateAsString that represents the given ET
fromString(dateAsString,dateFormat)	returns an Elapsed Time computed from the given string

Methods of the ProcessRunner class	

runImmediate(
processAndParameters,
currentDir,
randomWaitBeforeRun)
 	
runImmediate(
process,
arrayOfParameters,
currentDir,
randomWaitBeforeRun)
	Run an external executable.
	
	Here is an example that downloads the homepage from Google using curl:
	
	`var p=new ProcessRunner();`
	`var res=p.runImmediate("\""+anatellaDir+"/../curl/curl\" -k -o index.html https:\\www.google.com");`
	`if (res==0) print("download successful"); else throw "Error download";`
	
	Or, alternatively (this is better):
	
	`var p=new ProcessRunner();`
	`var res=p.runImmediate("\""+anatellaDir+"/../curl/curl\"",`
	`["-k","-o","index.html","https:\\www.google.com"]);`
	`if (res==0) print("download successful"); else throw "Error download";`
	
	The second syntax (based on an array of parameters) is more complex to use but it allows you to pass any command-line switches to the launched process (i.e. even the command-line switches that includes forbidden characters such as the “quote” or the “ampersand” character are allowed using this syntax).
	
	The last two parameters (*currentDir* and *randomWaitBeforeRun)* are optional.
	
	The *process* parameter contains the command-line used to run the process: i.e. The full path to the executable, optionnaly followed by command-line switches (such as “–DfileIndex=3 –DObsDate=20120629”).
	
	The *currentDir* parameter defines the current Directory used to run the process. The default value of the *currentDir* parameter is “:/”.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the ProcessRunner class	
	The *randomWaitBeforeRun* parameter defines the maximum number of milliseconds that Anatella must wait before running the process. The default value of the *randomWaitBeforeRun* parameter is the value given at the creation of the ProcessRunner Object. When a program terminates, it always returns back to MSWindows a number. This number is named the "error level" of the program. By convention, an "error level" that is non-zero indicates that there was an error during the execution of the program. Anatella extends slightly the convention: see section 4.7. for more information about this subject. The "*runImmediate()*" method returns the "error level" of the program it executed.
runImmediateExtended(processAndParameters, currentDir, textEncoding, dataToWrite) runImmediateExtended (process, arrayOfParameters, currentDir, textEncoding, dataToWrite)	This is basically the same method as the "*runImmediate*()" method described here above. The difference between "*runImmediateExtended*()" and "*runImmediate* ()" exists only in the return value: • The "*runImmediate()*" method simply returns the "error level" of the program it executed. • The "*runImmediateExtended*()" method returns an object with two fields: o "*errorLevel*": the "error level" of the executed program. o "*output*": the content of the text displayed inside the console during the program execution (all stdout+stderr). By default, the text encoding is the global-OS-text encoding. The last parameter "*textEncoding*" allows you to change the encoding. Only the first parameter is mandatory. The *currentDir* parameter is optional: It defines the current directory used to run the process. The default value of the *currentDir* parameter is ":/". The *currentDir* parameter defines the current Directory used to run the process. The default value of the *currentDir* parameter is ":/". The *dataToWrite* parameter defines the data that will be passed to the stdin of the launched process. The *dataToWrite* parameter can be a string or a binary object obtained using the "*fileBinaryLoad*()" function. The *textEncoding* parameter defines the encoding of the characters obtained from running the process (when the *encoding* parameter is omitted, Anatella assumes: *encoding*=0): • encoding=0: The characters are in the local system-wide encoding (This is usually ISO-8859-1) • encoding=1: characters are in UTF8 • encoding=2: characters are in UTF16 • encoding=3: characters are in Latin1 • encoding=3: characters are in IBM850 encoding
runDetached(processAndParameters) runDetached(process,	Starts the new process with the given arguments, and detaches from it (i.e. Anatella won't wait until the termination of the new process). If Anatella exits, the detached process will continue to live. Return true on success (i.e. a new process was launched) and false otherwise.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the ProcessRunner class	
arrayOfParameters, currentDir)	The *currentDir* parameter is optional: It defines the current directory used to run the process. The default value of the *currentDir* parameter is ":/".

runAnatella(
anatellaGraphFile,
AllParametersAsAString,
currentDir,
randomWaitBeforeRun)
 	
runAnatella (
anatellaGraphFile,
arrayOfParameters,
currentDir,
randomWaitBeforeRun)
	Runs a .anatella data-transformation-graph file.
	
	This is a specialized version of the "runImmediate()" method that allows to run .anatella files with a simpler syntax.
	
	Here is a small example:
	
	`var p=new ProcessRunner();`
	`p.runAnatella("aggregate_products.anatella","-DfileIndex=1 -Dobservation=2000");`
	
	Or, alternatively (this is better):
	
	`var p=new ProcessRunner();`
	`p.runAnatella("aggregate_products.anatella",["-DfileIndex=1","-Dobservation=2000"]);`
	
	The second syntax (based on an array of parameters) is more complex to use but it allows you to pass any command-line switches to the launched process (i.e. even the command-line switches that includes forbidden characters such as the "quote" character are allowed using this syntax).

addJob (processAndParameters, currentDir, nRetry) addJob (process, arrayOfParameters, currentDir, nRetry)	Add an executable to the list of executable that will be launched when calling the "*run()*" method of the ProcessRunner Object. The second syntax (based on an array of parameters) is more complex to use but it allows you to pass any command-line switches to the launched process (i.e. even the command-line switches that includes forbidden characters such as the "quote" character are allowed using this syntax). The 2 last parameters (*currentDir* and *nRetry*) are optional. The *currentDir* parameter has the same purpose and meaning as for the "*runImmediate()*" method, hereabove. When the executable defined by the current *process* parameter fails, Anatella can, either: abort, ignore or re-try (i.e. "re-try"=re-execute the same executable once again). The number of re-try is defined as the *nRetry* parameter. The default value of the *nRetry* parameter is the value given at the creation of the ProcessRunner Object (that is, by default, 5).
addAnatellaJob(anatellaGraphFile, AllParametersAsAString, currentDir, nRetry) addAnatella Job(anatellaGraphFile,	Add a .anatella file to the list of processes that will be launched when calling the "*run()*" method of the ProcessRunner Object. This is a specialized version of the "addJob()" method that allows to queue .anatella files for execution with a simpler syntax.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

<table>
<tr><th colspan="2">Methods of the ProcessRunner class</th></tr>
<tr><td>arrayOfParameters,
currentDir,
nRetry)</td><td></td></tr>
<tr><td>listJobs()</td><td>Returns an array of objects. Each object has the following fields:
process : Self-explanatory.
defaultDir: Self-explanatory.
args: Self-explanatory.</td></tr>
<tr><td>run(ncpu,
abortCondition,
randomWaitbeforeRun)</td><td>This runs all the executables that are inside the list created using the “addJob()” method.

All parameters are optional. Usually, only the first parameter is used.

Anatella will run in parallel ncpu executable (from the current list of executable to run). As soon as one of the ncpu running process is finished, Anatella directly launch the next process, so that there are always ncpu processes running at the same time. This allows us to use all the several CPUs available in modern computers at maximum efficiency. The default value for the ncpu parameter is “1” (when ncpu=1, Anatella runs sequentially all the processes in the list in the order they were placed inside the list).

The abortCondition parameter can have the following values:
<table><tr><th>Constant</th><th>Value</th><th>Description</th></tr><tr><td>PR_NeverAbort</td><td>0</td><td>Never abort (No retry)</td></tr><tr><td>PR_NeverAbortButRetry</td><td>1</td><td>Never abort but does retry while the run FAILED</td></tr><tr><td>PR_AbortIfFail</td><td>2</td><td>Abort if the run still FAILED after some retry</td></tr><tr><td>PR_AbortIfWarning</td><td>3</td><td>Abort if the run had still a WARNING after some retry</td></tr></table>
To detect a FAILURE or a WARNING during a graph execution, Anatella uses the “error level” of the process: See section 4.7. to know more about “error levels”. The default value for the abortCondition parameter is “PR_NeverAbortButRetry”.

Typical usage of the abortCondition=“PR_AbortIfFail” option is executing a FTP file transfer (e.g. using cUrl). Such type of task can easily fail. Failure is detected and Anatella re-attempt a new FTP file transfer. The maximum number of attempt is specified using the parameter “nRetry” of the job (defined using the “addJob()” method of the ProcessRunner object).

The randomWaitbeforeRun parameter has the same purpose and meaning as for the “runImmediate()” method, hereabove.

The “run()” method returns the “error level” of the program with the largest “error level” (in absolute value) amongst all the programs it executed. The return value is zero if no errors or warnings were detected.</td></tr>
</table>

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the ProcessRunner class	
setVerbose(b)	Self-explanatory. The parameter "b" is true/false. You can reduce verbosity writing "*p.setVerbose(false);*"

Variables of the KeepBests class	
N	The number of (key,value) pairs to store inside the KeepBests Object.

Methods of the KeepBests class	
add(key,value)	Add a new (key,value) pair inside the KeepBests Object. If the object already contains the maximum number of pairs, the pair with the lowest key is discarded.
getKey(index)	Gets the key stored inside the (key,value) pair of given index. The index zero returns the greatest key.
getValue(index)	Gets the value stored inside the (key,value) pair of given index. The index zero returns the values associated with the greatest key.
reset()	Empties out the object.

Methods of the SendMail class (**LIB:** EMailSMS)	
setLoginPassword (login,password)	Self-explanatory
isConnected()	Returns true if this instance of the SendMail Object is now connected to the SMTP server. Once the initial connection has been established, all further emails are sent at very high speed.
send(toAddress,subject, messageBody,ccAddress, bccAddress,isHTML)	Sends one email (and establish a connection to the SMTP server first, if required). The parameters *ccAddress*, *bccAddress* and *isHTML* are optional. If omitted, these default values are used: • *ccAddress=""* • *bccAddress=""* • *isHTML*=false All *address* parameters (i.e. toAdresss, ccAddress and bccAddress) have the following properties: • They can contains several e-mails, separated by comma's. • Each e-mail can optionally have a "name" that precedes it: When using "names" the actual email must follow and be enclosed inside < and >. For example: John Smith <JSmith@gmail.com> The *messageBody* parameter usually contains simple text (this is the default behaviour). It can also contains HTML code (when the parameter *isHTML*=true).
close()	Close the connection to the SMTP server and free all related resources.

Methods of the SQLite class (**LIB:** DBConnectors)	
isOpen()	Self-explanatory.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the SQLite class (**LIB:** DBConnectors)	
close()	When you don't need the "SQLite" object anymore, you must always call the "close()" method to remove any lock on the SQLite dabase file, as-soon-as-possible, to allow other processes to access the database.
lastError()	Returns the last SQLite Error. Please, refer to this page: https://www.sqlite.org/c3ref/c_abort.html … to have more information about the error.

execute(stmt)	Execute the given SQL statement using the SQLite engine.
	
	The returned value depends on the nature of the statement:
	• If the statement returns no rows:
	  ○ If the execution succeeded: The return value is 101.
	  ○ If the execution failed: The return value is null. Use "lastError()" to get a more precise description of the error.
	• If the statement returns one or more rows, only the first row is returned inside the Javascript environment.
	  ○ If the returned row contains many columns, you'll receive an array with the content of the columns.
	  ○ If the returned row contains only one column, you'll receive a simple variable with the content of the column.
	
	Here is an example of usage:
	`if (fileExist(":/Profiles.sqlite"))`
	`{`
	`  var sq=new SQLite(currentDir+"Profiles.sqlite",SQLite_ReadOnly);`
	`  var r=sq.execute("SELECT 1 FROM sqlite_master WHERE type='table' AND name='Profiles'");`
	`  if (r==1)`
	`  {`
	`  // we arrive here if the table exists...`
	`  r=sq.execute("SELECT 1 FROM Profiles where ID=42");`
	`  if (r==1)`
	`  {`
	`  // we arrive here if the given ID exist`
	
	The above Javascript code test if the table "Profiles" contains the ID=42. Before checking for the ID, we check that:
	• the database file exists (i.e. the file ":/Profiles.sqlite" must exist).
	• the table "Profiles" exists.

Methods of the AnatellaGraph class	
isOk()	Self-explanatory.
hasChanged()	Self-explanatory.
save(filename)	The parameter "filename" is optional. If omitted, Anatella uses the filename used to open the graph.
saveForVersioning(filename, resetUsers)	Both parameters are optional. If the "filename" parameter is omitted, Anatella uses the filename used to open the graph. If the "resetUsers" parameter is omitted, Anatella uses "resetUsers"=True
setEncryption(b)	The parameter "b" is true/false.
setEncoding(s)	The parameter "s" can be "utf8" or "utf16".
setTempDir(dir)	Self-explanatory.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Methods of the AnatellaGraph class	
setTempDirLoc(i)	Set the Temporary Directory location: i=0 : global OS temp dir i=1 : Anatella-specific temp dir i=2 : Graph-specific temp dir
addUser(login)	Self-explanatory.
removeUser(login)	Self-explanatory.
setUserPassword(login,password)	Self-explanatory.
hasUserPassword(login)	Self-explanatory. Returns true/false.
setUserAdmin(b)	Self-explanatory. The parameter “b” is true/false.
setUserExpirationDate(date)	Self-explanatory. The parameter “date” is a string with the format “yyyyMMdd”.
setUserEditGP(b)	Self-explanatory. The parameter “b” is true/false.
setUserEditGraph(b)	Self-explanatory. The parameter “b” is true/false.
setUserReadDB(b)	Self-explanatory. The parameter “b” is true/false.
setUserWriteDB(b)	Self-explanatory. The parameter “b” is true/false.
setODBCLogin(connection name,login)	Self-explanatory.
setODBCPassword(connection name,login)	Self-explanatory.
setGP(global parameter name, value)	Self-explanatory. Create a new Global Parameter if none is found with the given name.
getUsers()	Returns an array describing the users of the Anatella graph. Each element of the array is an object with the fields “login”,“isAdmin”, “canEditGP”, “canEditGraph”, “canReadDB”, “canWriteDB”, “expirationDate”
getODBC()	Returns an array describing the ODBC connection used in the Anatella graph. Each element of the array is an object with the fields “name”, “link”, “login”, “password”
getGP()	Returns an array describing the Global Parameters of the Anatella graph. Each element of the array is an object with the fields “name”, “value”, “comment”
getFiles()	Returns an object with 2 fields: “input” and “output”. These two fields (“input” and “output”) are arrays describing the input and ouput files used in the Anatella graph. Each element of these array is an object with the fields “action” and “filename”.

Anatella global-functions related to Mathematical Optimization	
simpleGlobalOptimizer (nameObjectiveFunction, minX, maxX, nIterationMax)	Find x*, the exact value of x (between minX and maxX) for which the ObjectiveFunction is minimum (it returns a minimum value). The prototype of the ObjectiveFunction is something like: function f(x) { return <a_function_of_x> ; }

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu

Anatella global-functions related to Mathematical Optimization	
	The parameter "nameObjectiveFunction" is a string that contains the name of the Javascript objective function to minimize. The algorithm searches for a **global** minimum in-between minX and maxX. It terminates when the number of iterations reaches "nIterationMax". Return an object with 2 fields: xOpt and yOpt.
simpleLocalOptimizer (nameObjectiveFunction, minX, maxX, startX, startStepX, finalStepX)	Find x*, the exact value of x (between minX and maxX) for which the ObjectiveFunction is minimum (it returns a minimum value). The prototype of the ObjectiveFunction is something like: function f(x) { return <a_function_of_x> ; } The parameter "nameObjectiveFunction" is a string that contains the name of the objective function to minimize. The algorithm searches for a **local** minimum in-between minX and maxX. It starts from the position "startX" (ideally, "startX" should be near x*, otherwise there is a larger risk of finding a local minimum instead of a global minimum) and does "steps" of size "startStepX". When approaching x*, the step sizes are decreased. The algorithm stops when the step size is smaller than "finalStepX". Return an object with 4 fields: xOpt, yOpt, nIter, info.

©2010 TIMi SPRL. E-mail: sales@timi.eu - Visit us at www.timi.eu