

Datalator 3 Visual Development Guide.

1. Preparation.

The easiest way to start developing of a new application is to download and unzip the Datalator3, Lite Edition. It comes with a server and a client in one package, easy to begin with, provides the same functionality and those "lite" applications are fully compatible with other Datalator3 Editions.

Just to remind you, the home directory of the Lite Edition after installation looks like that:

Name	Date modified	Type	Size
AdminApplication	30/08/2011 4:56 PM	File Folder	
DatalatorApplication	30/08/2011 4:56 PM	File Folder	
lib	30/08/2011 4:56 PM	File Folder	
Datalator3_Lite.jar	30/08/2011 2:54 PM	Executable Jar File	5 KB
LICENSE.TXT	14/06/2011 2:16 PM	Text Document	8 KB

First "empty" DatalatorApplication is already created (see Admin Guide to learn how) and all you need to do is to start the Lite version by double clicking on the Datalator3_Lite.jar or typing in the Command Prompt window: **java -jar Datalator3_Lite.jar**. Click a "Connect" button now and start the development (see below).

The home directory of a Personal or Professional Edition look just a bit different, like this:

Name	Date modified	Type	Size
AdminApplication	30/08/2011 4:42 PM	File Folder	
lib	30/08/2011 4:42 PM	File Folder	
DatalatorClient3.jar	30/08/2011 2:47 PM	Executable Jar File	813 KB
DatalatorServer3.jar	30/08/2011 4:42 PM	Executable Jar File	82 KB
Download Time	30/08/2011 4:37 PM	File	1 KB
LICENSE.TXT	16/06/2011 10:16 ...	Text Document	8 KB
LinuxLaunch.sh	16/06/2011 12:06 ...	SH File	1 KB
Server ID	30/08/2011 4:37 PM	File	1 KB
WindowsLaunch.bat	16/06/2011 12:03 ...	Windows Batch File	1 KB

There is no ready to use an "empty" application for development. You have to start the server first by starting **WindowsLaunch.bat** or **LinuxLaunch.sh** in the Command Prompt, start the client Datalatorclient3.jar like: **java -jar DatalatorClient3.jar**, connect as admin/admin and create an empty application (see Admin Guide to learn how). Then start and connect another instance of the DatalatorClient as developer to that empty application to proceed with development.

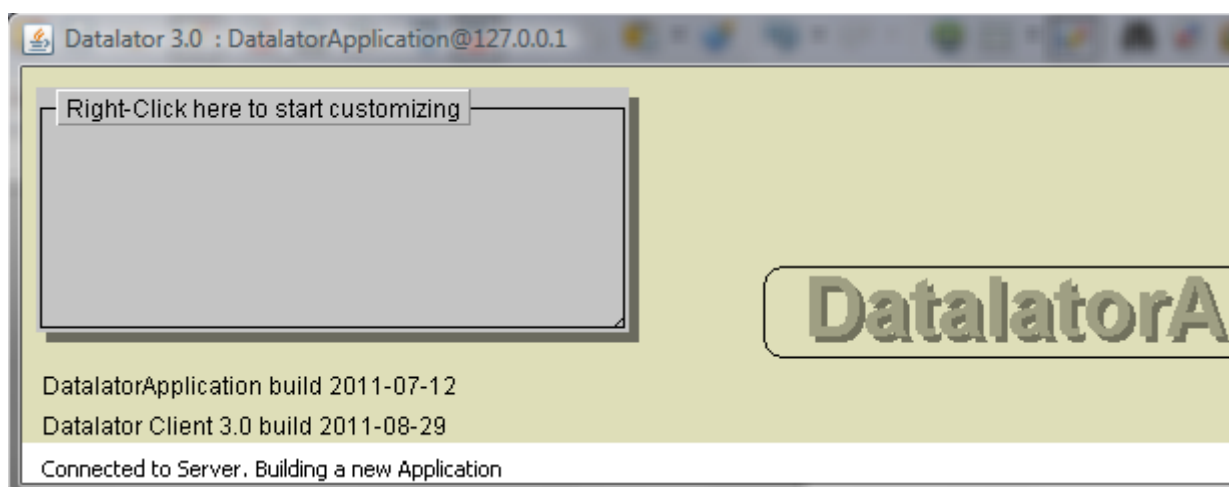
At this point development of a new application will be exactly the same for all Datalator Edition.

2. Starting Development.

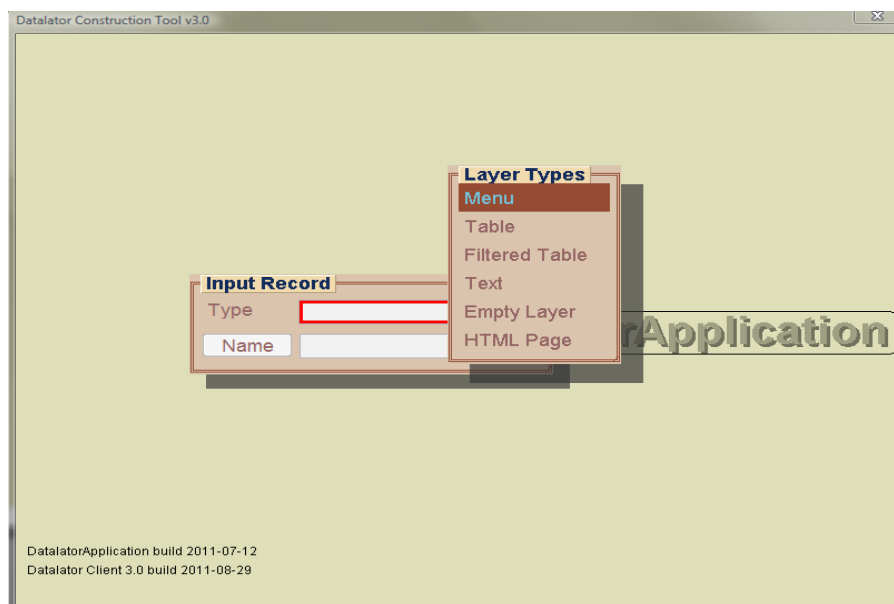
When you connect a client with a dialog like that to an "empty" application:

A connection dialog box with a light green background. It contains four input fields: 'Server Name' with '127.0.0.1', 'Application Name' with 'DatalatorApplication', 'Server Port' with '8081', and 'Service Port' with '8082'. Below these are 'User Name' with 'owner' and 'Password' with six dots. At the bottom left is a 'Cancel' button and at the bottom right is a 'Connect' button.

you will first see a "dummy" layer (or component) of that application like this:



The frame of the connected client might be bigger and you might name an application differently, but that's all the differences. Follow the prompt on the caption of the gray "dummy" and right-click on it. That it a standard way to start customization of a new or existent layer. Another frame will pop up, that is the built-in development tool – and it looks like this:



Every time you customize a new component-layer you see that tool with shown dialog. Lets speak about components now.

3. Components.

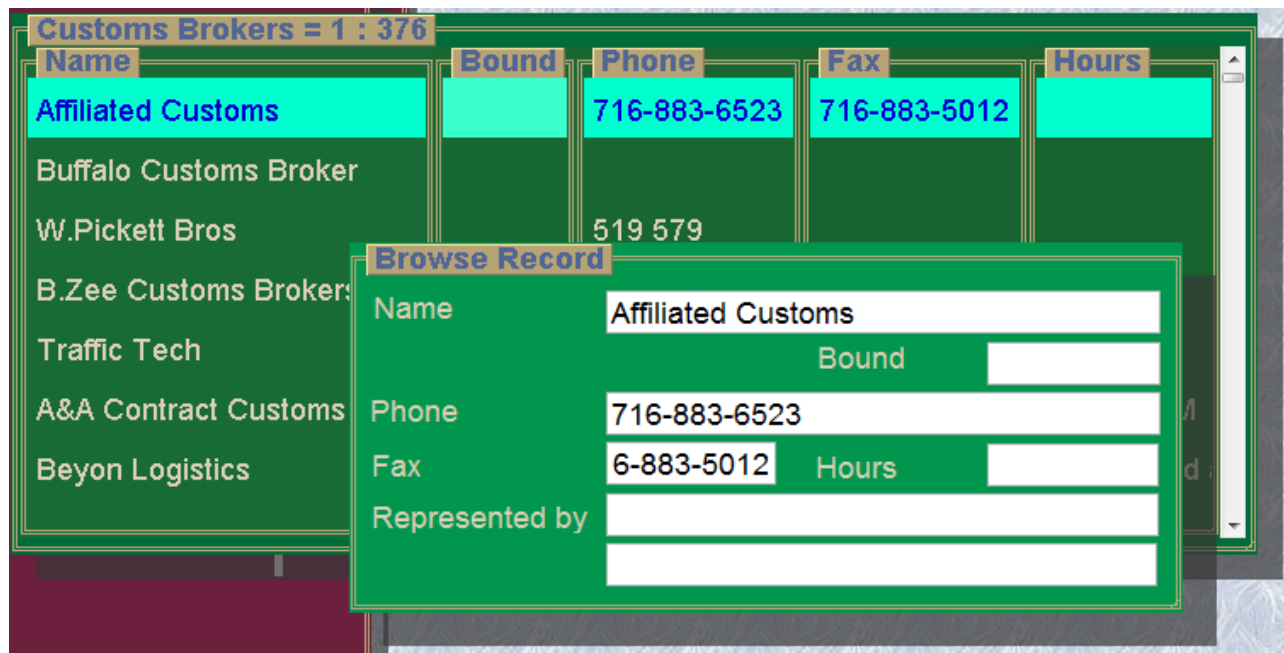
Present Datalator version creates an application from 6 types of components. Those are:

Menu



A menu is a simple list of lines, every line is meant to activate another layer of any type or to initiate an action, like shut down the program. Only a developer can change the menu content. Developer can change the menu records' order without breaking already defined links.

Table



A table is an unlimited set of records, each consisting of number of fields. Records maintain given order, might be sorted, searched, added, changed, deleted. Table supports concurrent users access, meaning only one user might own a lock to edit a record – all others will be able to browse it. A locked record cannot be deleted. A table gets instant notification from the server if it's records have been changed by other users and reloads the content on the background. Design could be changed

by developer online when other users working, content could be exported/imported in CSV format (MS Excel). A record of one table could be linked to another component or action, like a menu. Convenient to support hierarchical data structures – supports "one-to-many" relations. A table is visualized in a "rows and columns" fashion and equipped with a form, which servers as an access tool to the fields of a particular row and as a record search tool.

A table might be supplemented with a number of "reports" – visible and printable tables with a subsets of original table columns and simple summarizing abilities.

Filtered Table

Filter

Dispatcher = Truck = 108
 Market = Trailer =
 Driver = Date >< 2011-07-01 2011-07-31

Trip	Driver	Truck	Date	Price Out	Price In	Stops	Price	Expences	Mileage
VZ200	Vlad2	108	2011-07-05	\$575.00	\$1,800.00	5	\$2,375.00	-\$0.00	
VZ208	Vlad2	108	2011-07-29	\$500.00	\$2,250.00	7	\$2,750.00	-\$0.00	
VZ207	Vlad2	108	2011-07-27	\$500.00	\$1,792.60	7	\$2,292.60	-\$0.00	
VZ206	Vlad2	108	2011-07-22	\$450.00	\$2,100.00	7	\$2,550.00	-\$0.00	

Browse Record

Trip	VZ208	Skids Out	26	Skids In	23	Skids	49	Mileage	
Driver	Vlad2	Spots Out	26	Spots In	19	Stops	7	Fuel used	
Truck	108	Weight Out	42640	Weight In	29802	Market	Cleveland, OH	Fuel paid	
Trailer	1109	Stops Out	1	Stops In	6	Dispatcher	Ella	Tolls paid	
Date	2011-07-29	Price Out	500.0	Price In	2250.0	Price	2750.0	Repairs	
		Border Out		Border In		Expences	-0.0	Other expences	

Average stops = 6.22 Average = \$2,696.68 Gross = \$24,270.10

A filtered table is a table equipped with a query maker – a filter component shown above the table. Supports "many-to-many" relations, convenient to search for subsets of records related to other tables. Allows in "click-and-point" fashion construct and execute complex searches like "what orders where taken by a specific agent for a specific class of products during specific time period?".

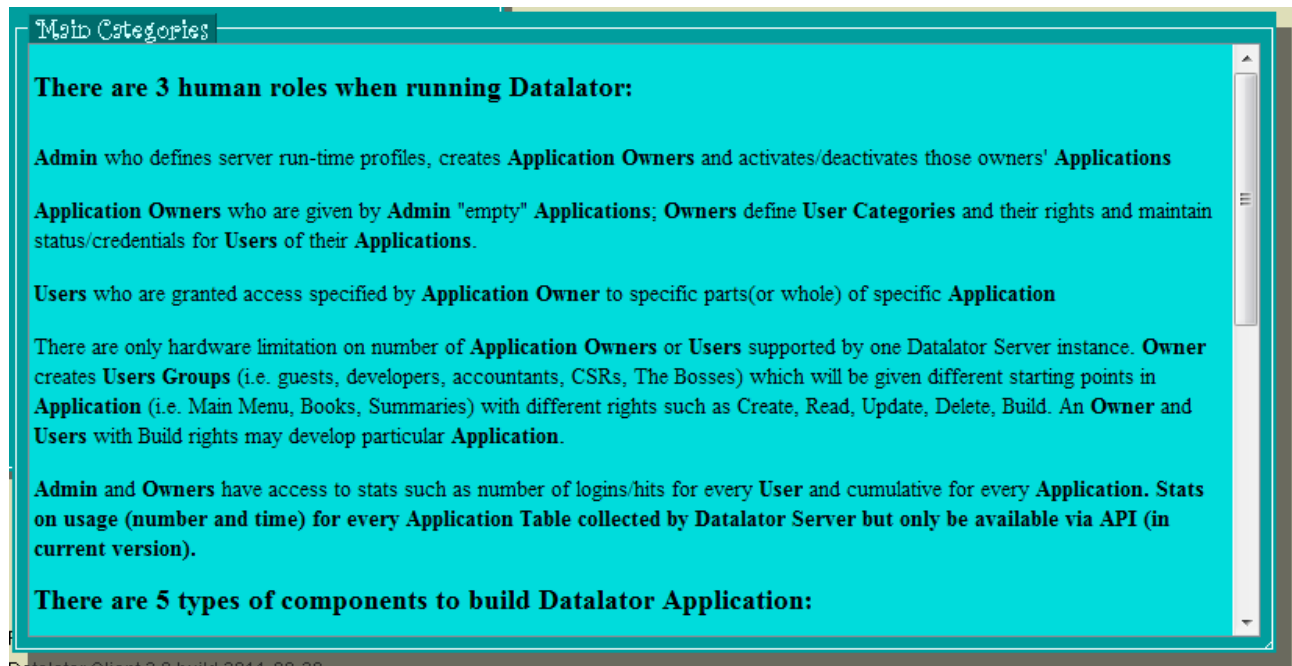
Text Layer

Present Log's Content

2011/09/07 09:31:45:417	< -- Transaction Committed (Dispatch 3) Read Sprav Cont
2011/09/07 09:31:54:950 76.70.13.13	(ella)Read Sprav Content
2011/09/07 09:31:54:950	-- > Transaction Started (Dispatch 3) SD: readContent
2011/09/07 09:31:54:951 76.70.13.13	SELECT * FROM "i117" ORDER BY recordOrder
2011/09/07 09:31:57:936 76.70.13.13	< -- Transaction Committed (Dispatch 3) Read Sprav Cont
2011/09/07 09:31:57:937 76.70.13.13	(ella)Read Sprav Content
2011/09/07 09:31:57:937 76.70.13.13	-- > Transaction Started (Dispatch 3) SD: readContent
2011/09/07 09:31:57:938 76.70.13.13	SELECT * FROM "i174" ORDER BY recordOrder
2011/09/07 09:32:01:065 76.70.13.13	< -- Transaction Committed (Dispatch 3) Read Sprav Cont
2011/09/07 09:32:01:065 76.70.13.13	(ella)Read Sprav Content
2011/09/07 09:32:01:065 76.70.13.13	-- > Transaction Started (Dispatch 3) SD: readContent
2011/09/07 09:32:01:066 76.70.13.13	SELECT * FROM "i195" ORDER BY recordOrder
2011/09/07 09:32:05:169 76.70.13.13	< -- Transaction Committed (Dispatch 3) Read Sprav Cont
2011/09/07 09:32:05:281 76.70.13.13	(ella)Add Sprav Record
2011/09/07 09:32:05:281 76.70.13.13	-- > Transaction Started (Dispatch 3) SD: addSpravRec
2011/09/07 09:32:05:281 76.70.13.13	SELECT * FROM "i472" WHERE recordId = '251'
2011/09/07 09:32:05:281 76.70.13.13	SELECT * FROM "i472" WHERE recordId = '266'

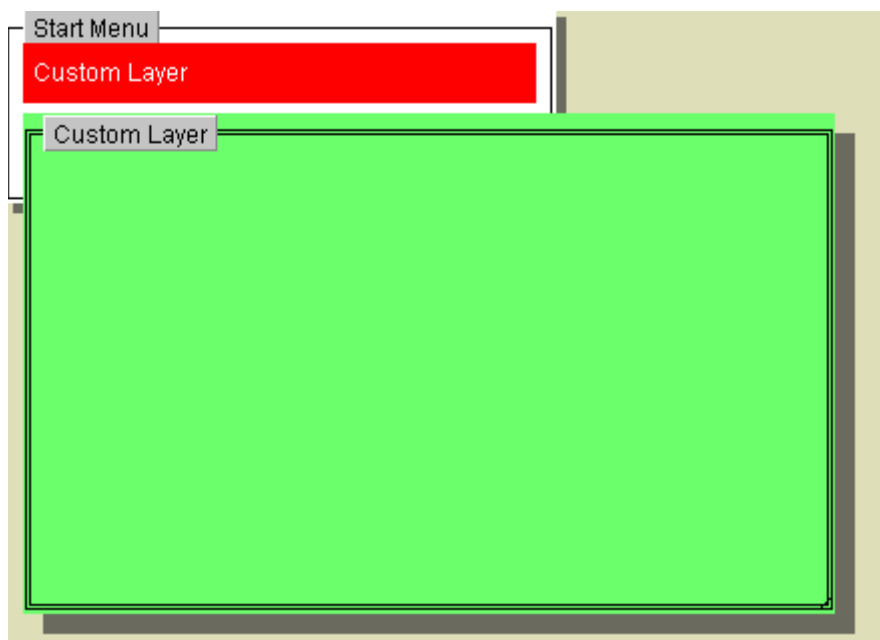
Text layer is a simple container for text information. Every line of a menu and every record of any table could be associated with a different editable text with specified design. Convenient to use as "sticky notes" for records in the database.

HTML Page

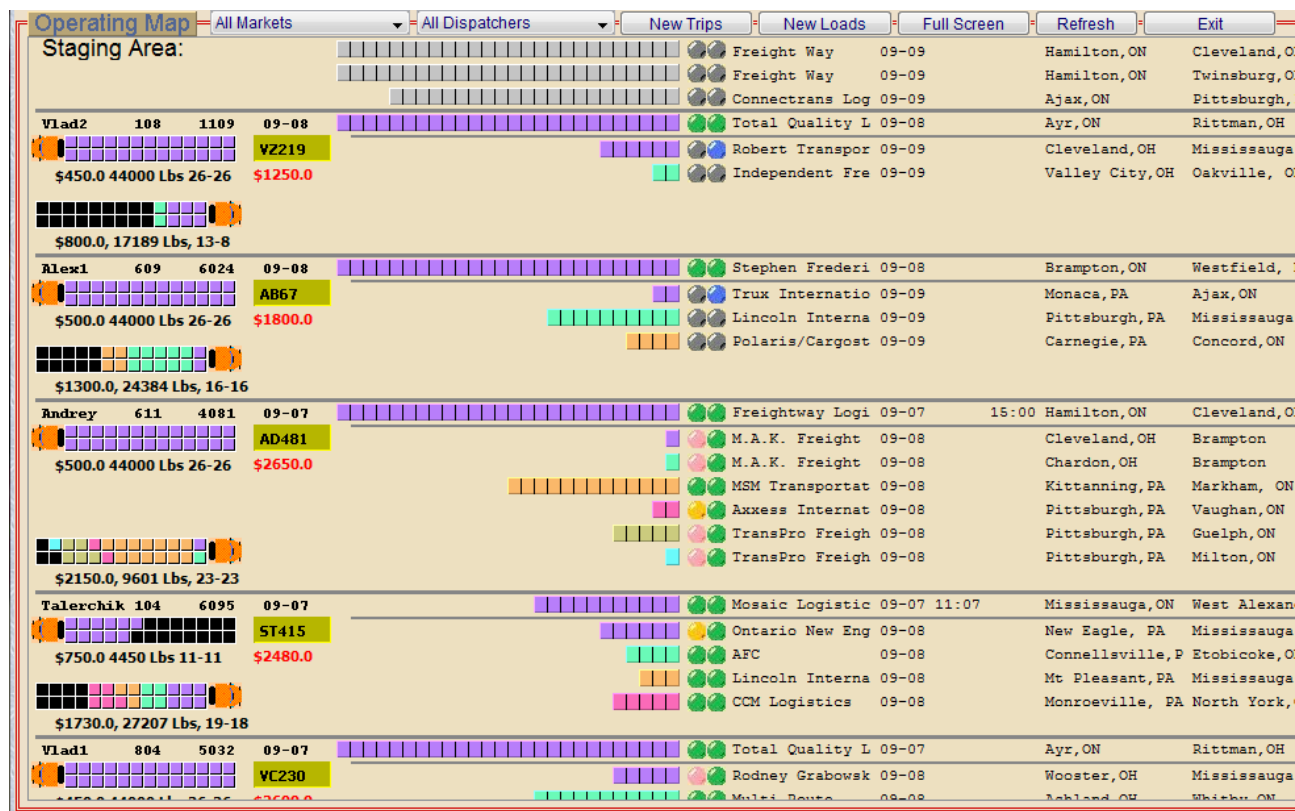


HTML component is a derivative to Text component, except it could be switched between "text" and "html" modes, which allows to edit content as html source and show it as an HTML page. That is not a full featured Web Browser, but is good to beautify the text and complement it is with images.

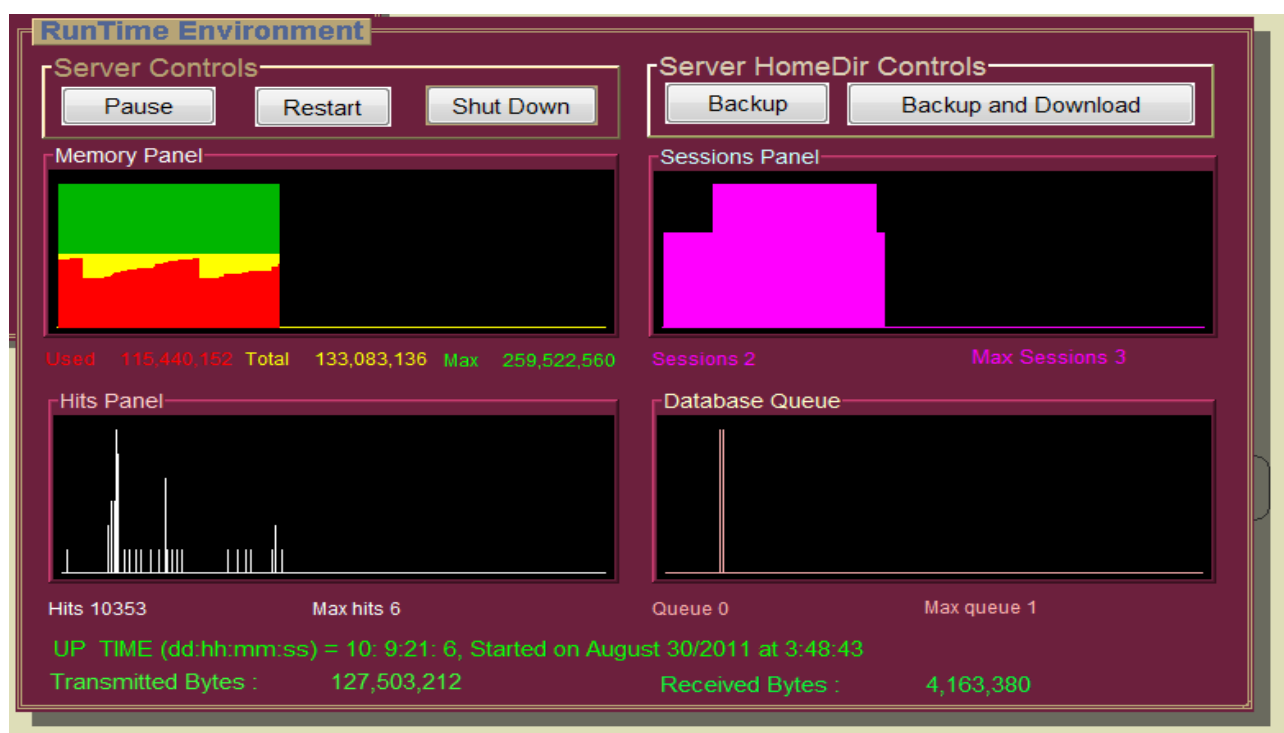
Empty Layer



Empty Layer is, well, just an empty panel. Named sometimes Custom Layer, because it is good for building interfaces which cannot be expressed in terms of columns and rows. Without programming it is not very useful, maybe only for static "help" or "about" screens. But with some programming it could be transformed into powerful dashboards or game panels, communicating real-time to a server and/or another users. Like that:

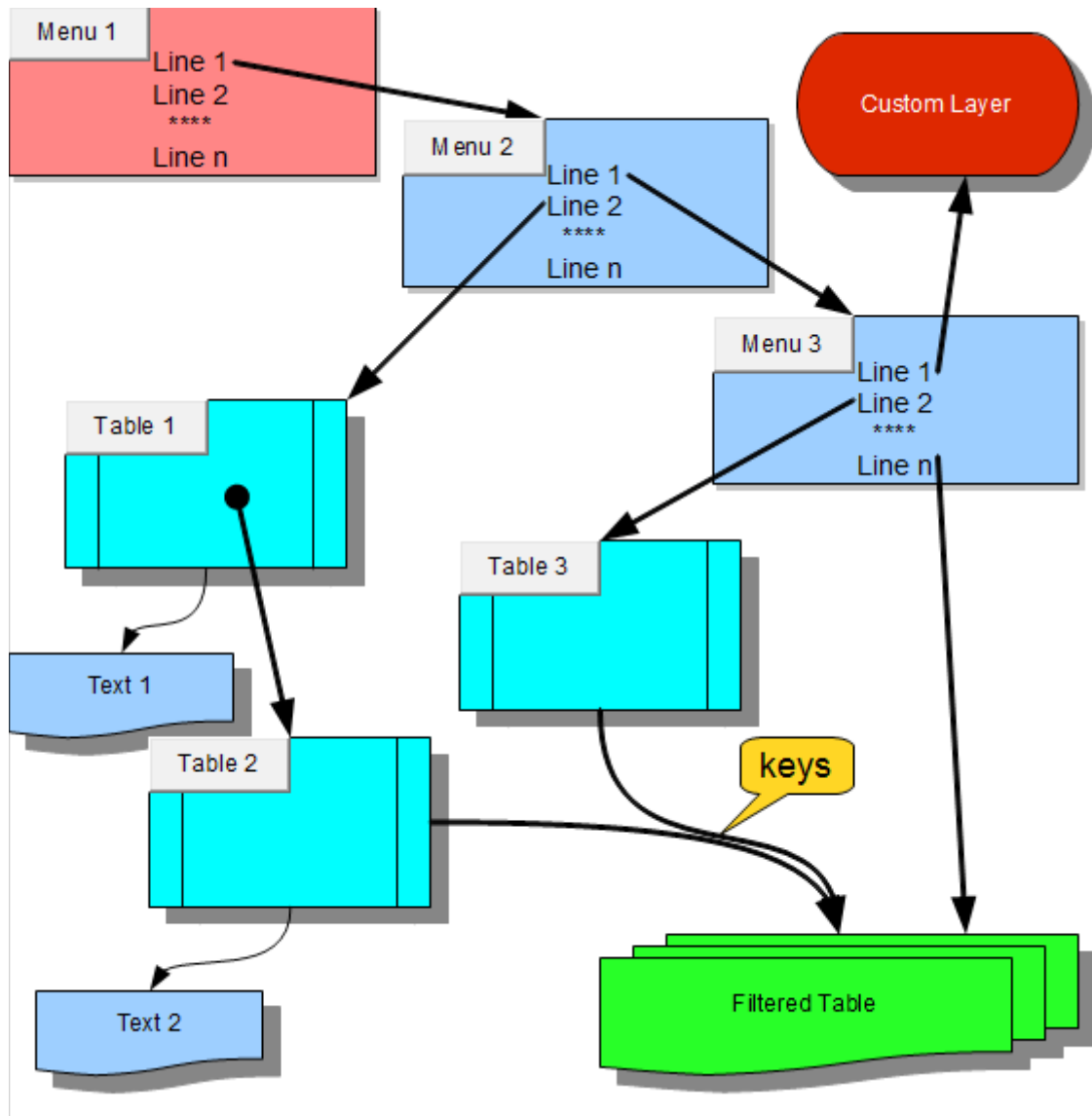


or like this:



4. Assembling Components.

To build an Application you have to connect some components into a schema like that:



Note: An application's entry point is always a Menu Layer.

Note: Every line of a Menu Layer represents a pointer to another component.

Note: All records of a Table Layer point to the same another component (usually another table).

Note: Every line of a Table Layer is a "key" to a subset of another table records (that subset is defined when inserting content into "another" table).

Note: A filtered table can be activated by one path (straight arrow from Menu 3) and queued by a few "key" tables (Table2 and Table3 in our example).

Note: Menu, Table and Filtered Table can activate other layers.

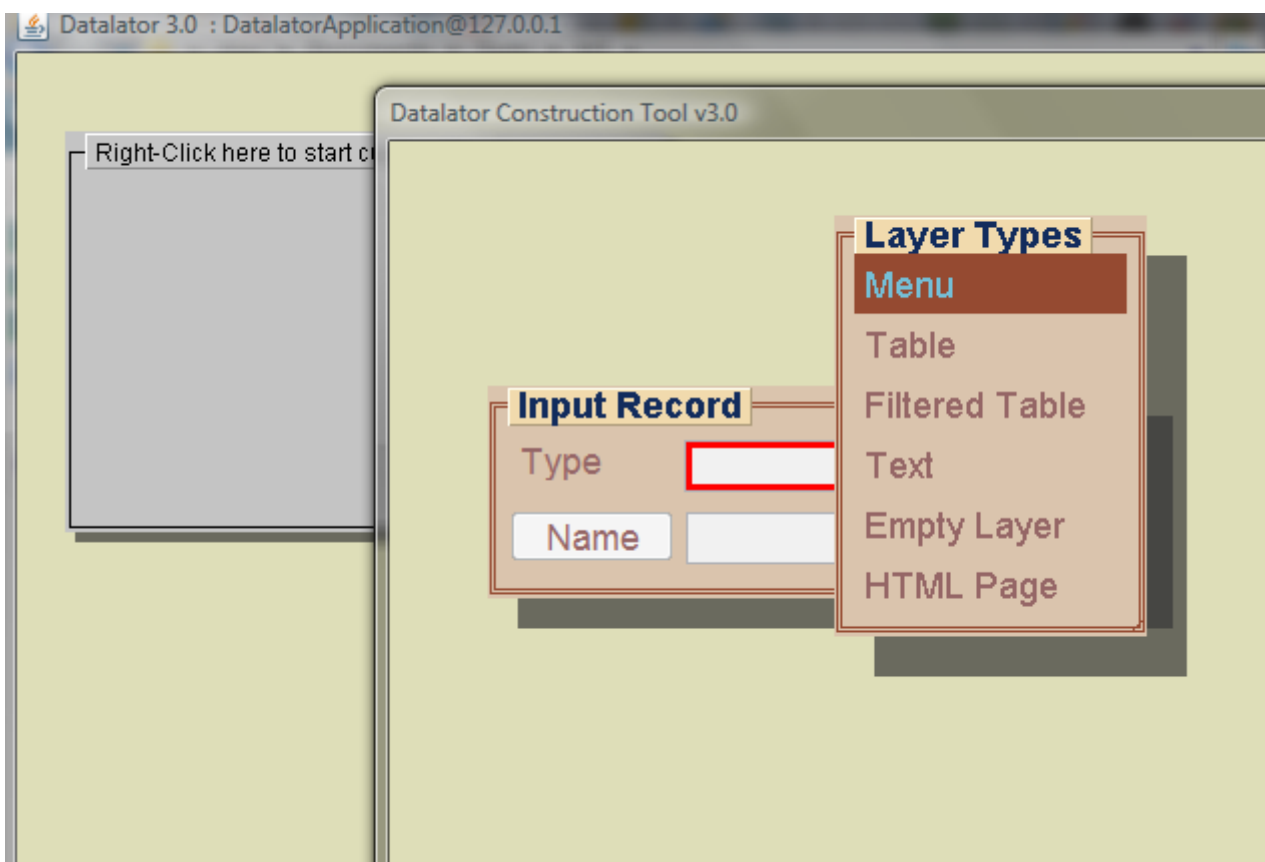
Note: Text, HTML and Custom Layers can not activate another layers – they are terminal.

Note: More components could be added/removed when the Application is online and other users working with data. A fields set of a table cannot be changed if a table contains(contained) data already. Content has to be exported, the table changed and content is to be imported back.

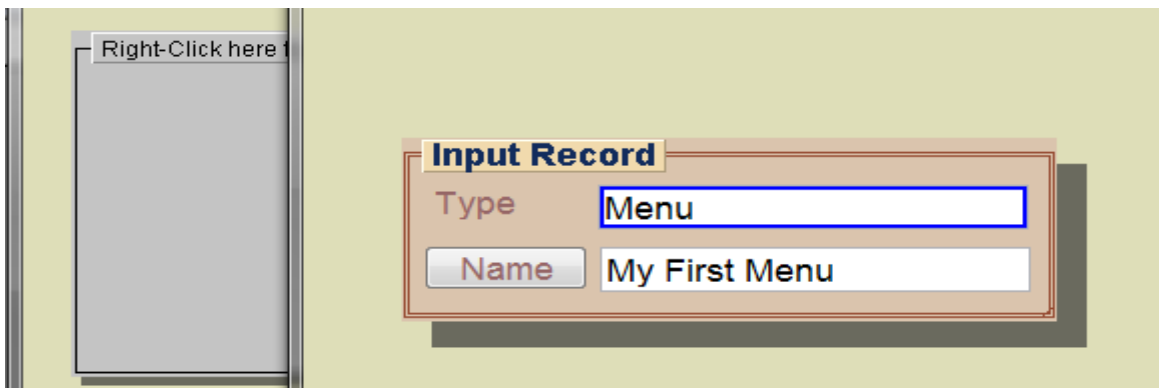
Thus building an application is the components(Layers) creation, customization and connection. That is a point-and-click process(or plug-and-play if you will) and is almost foolproof. There is no formal definition of application schema, nothing to type except components' and fields' names, no programming language to learn and to make mistakes with, when visually building applications. You may build a pretty complex one, with unlimited number of layers, multi-user by default with real-time collaboration, transactional and always consistent (google "acid" database to see more). To see how it's done let us build some killer application and we'll start with a Menu.

5. Building A Menu.

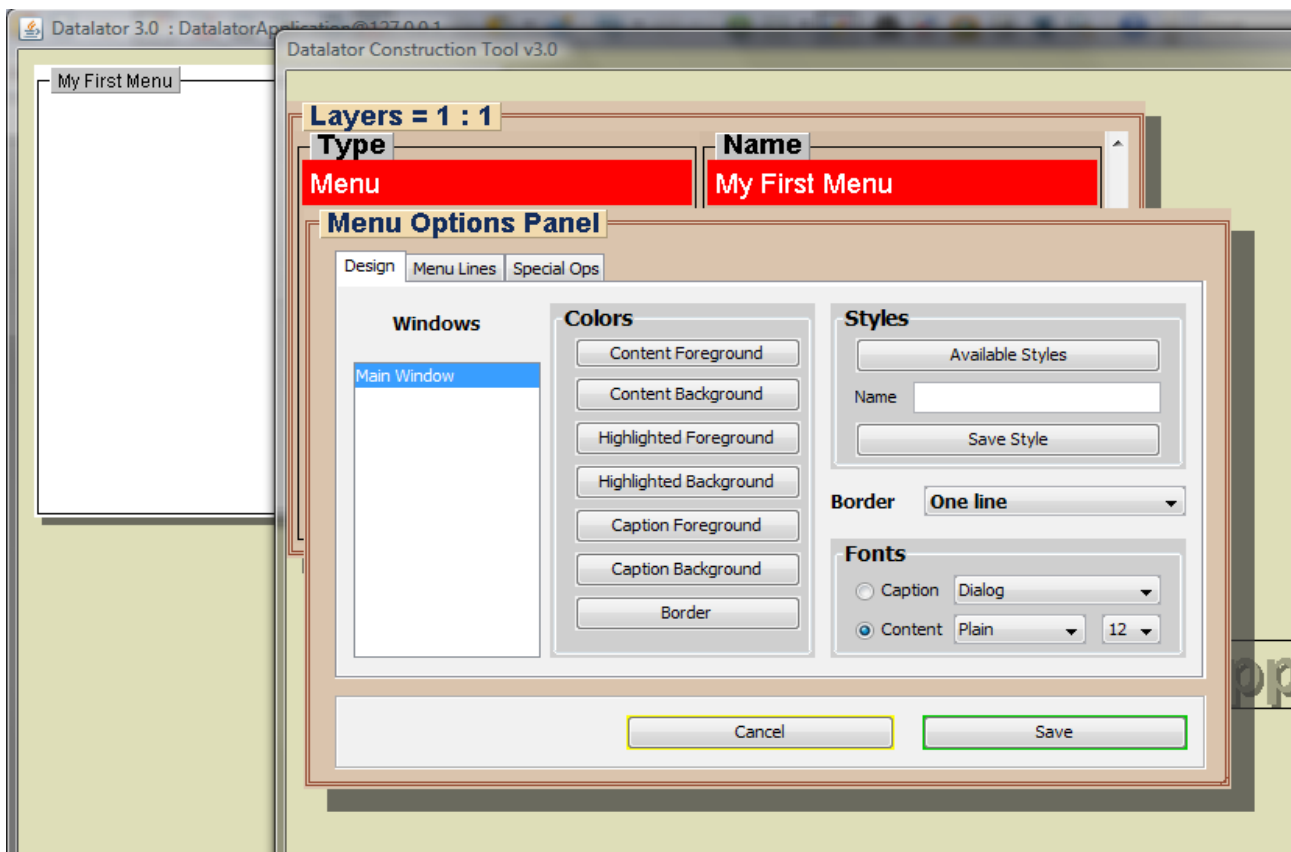
When an application developer/owner/administrator first connects to an "empty" application he or she will see the first "dummy" component; that component is to be given a type, a name and customized. Right-click on it's caption and you'll get a Construction Tool, asking for a type and a name:



It must start with a Type "Menu" and any name we like:

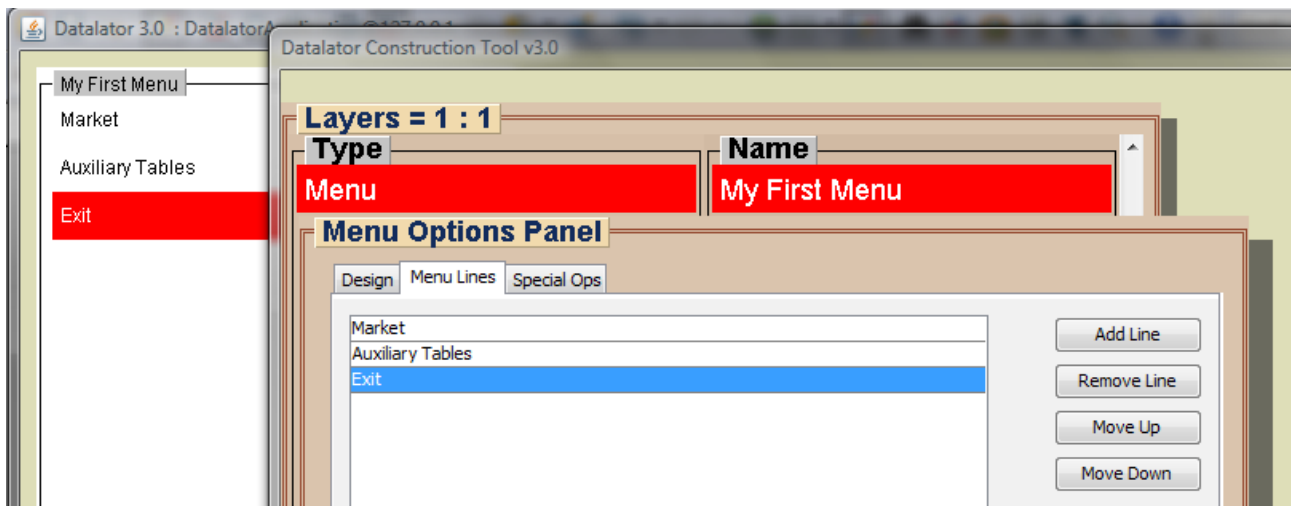


The Construction Tool is a built-in Datalator application by itself, running in another frame. And we can see a Menu Layer with the caption "Layer Types" and the Input Form, with a motto "Input Record". Every Input Form contains number of text fields and an "activator" button, which is responsible for input(or replacement) of a record. Click on that button or press "Enter" button, when cursor is in the last text field and you can see your first menu created and the Construction Tool visualizes Menu Options (or customization) Panel:

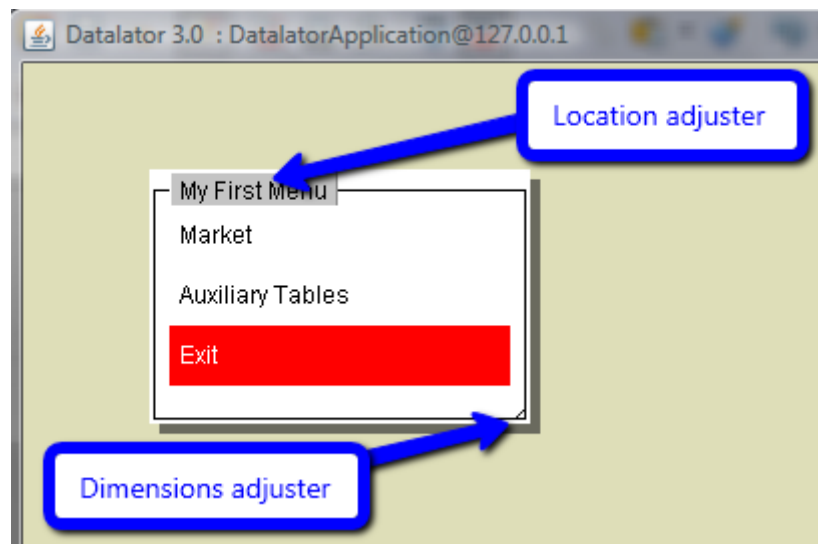


The Tool supports a list of the application's layers automatically(background layer named "Layers" with number of records 1 and "active" record 1 ("= 1 : 1" – notation). That is a convenience information, we are to use the "customization" layer named "Menu Options Panel" with 3 tabs: "Design", "Menu Lines" and "Special Operations".

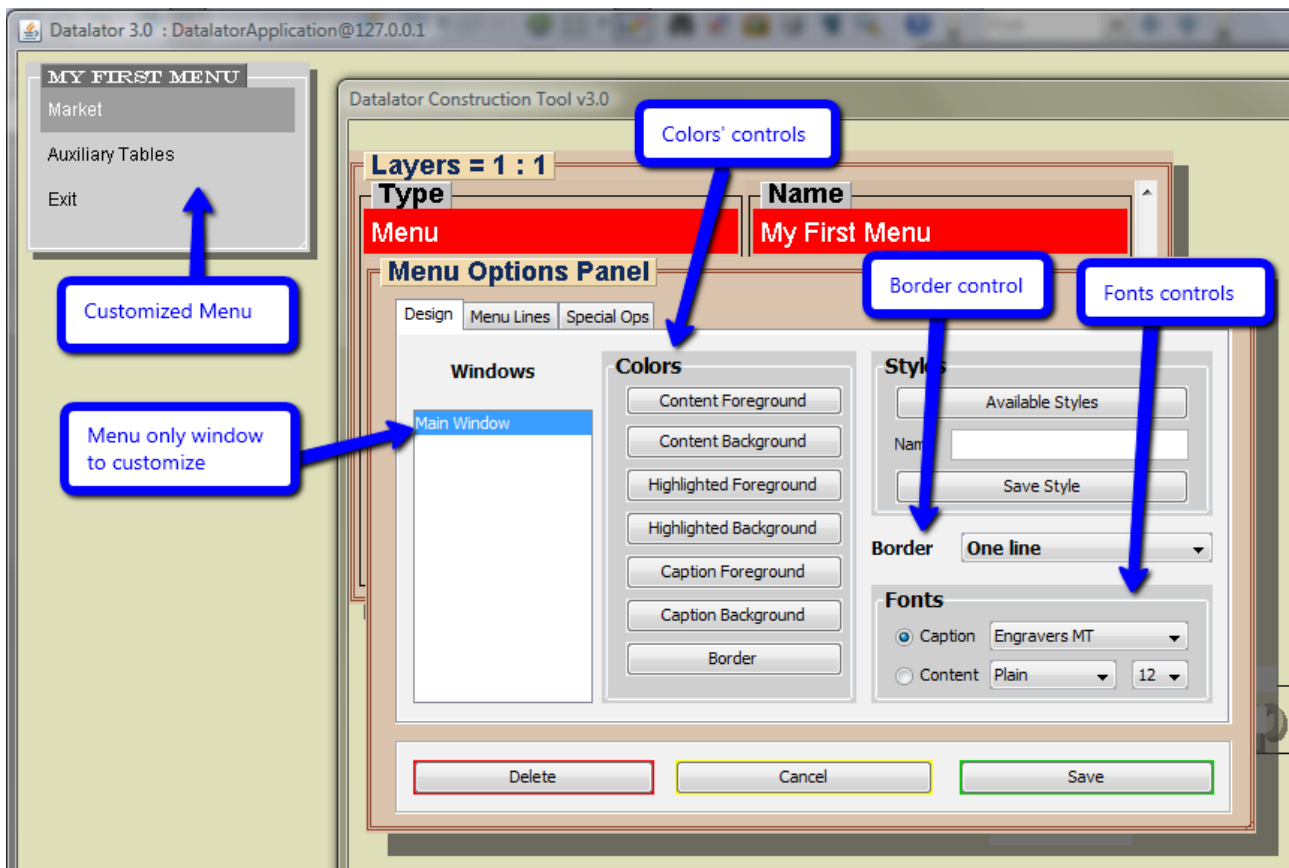
It make sense to start with "Menu Lines" tab and insert a few lines into that Menu:



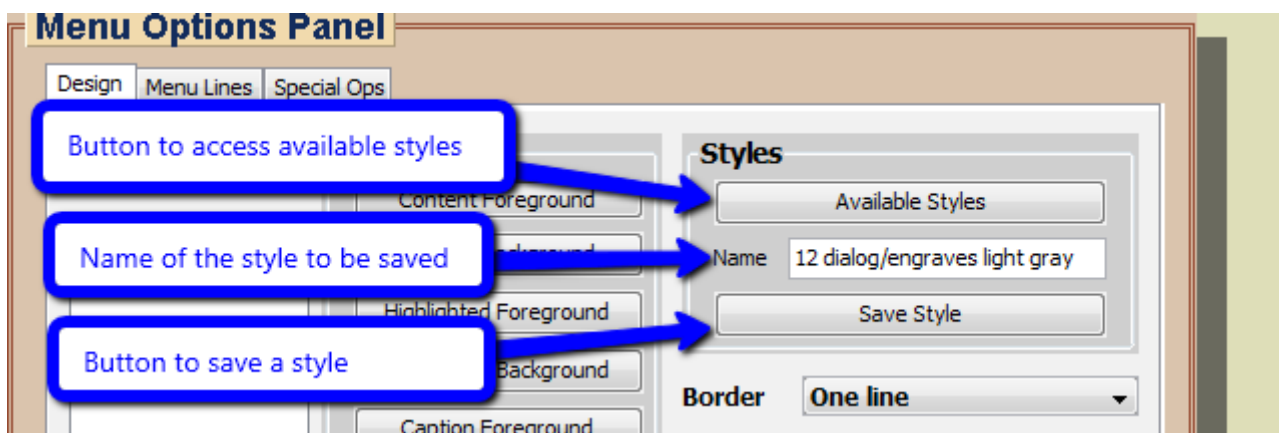
We obviously building some kind of market oriented application – "Buy and Sell" type. After defining the menu content we need to adjust the design and geometry(location and dimensions) of this menu. The location of a layer is set by dragging it by it's caption and to define the size you drag it by right-lower corner:



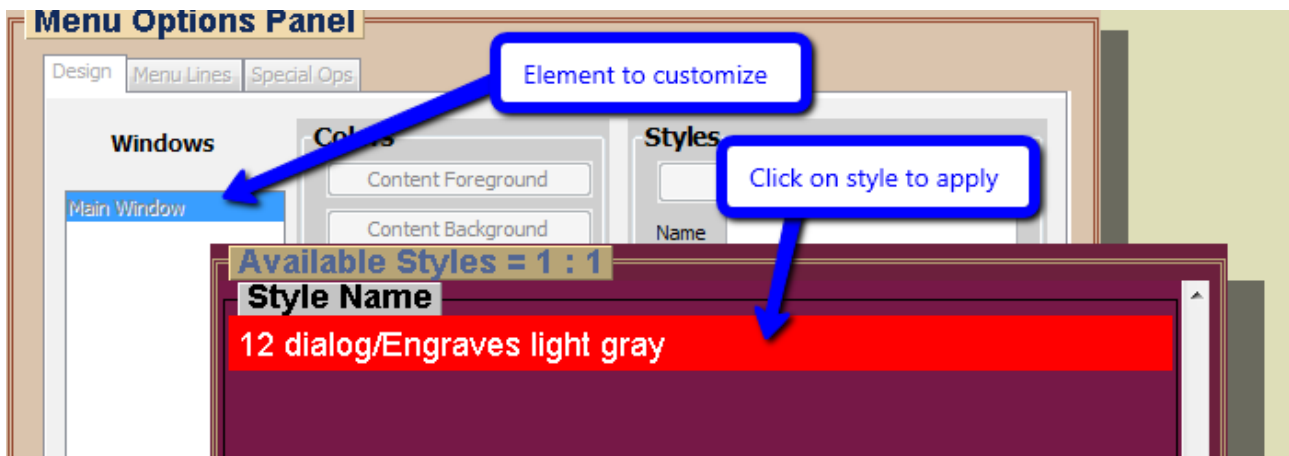
Now we can use colors, border and fonts controls to customize the appearance of the menu:



And when we've got a desirable design we can save it as a style to use it later when customizing other layers. Just type in the name of the style you want to save in the "Style name" text field and click on the "Save Style" button:



We can see our freshly developed style has been saved in the "Available Styles" table:

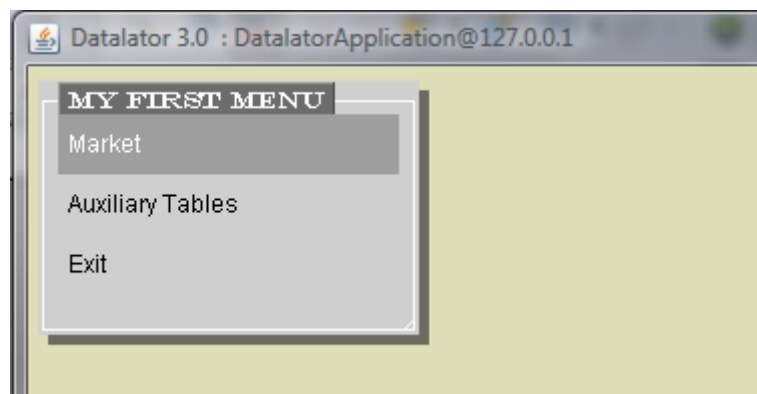


To use your styles later, click on the desirable element to customize in the "Windows" list, click on the "Available Styles" button and then on the desirable style from the list; chosen style will apply to the element, use it if you like it or try another one.

Finally, when you are completely satisfied with your Menu Layer, click on "Save" button – it will save and deploy the layer: it is online and could be used now.

Note: to change the design of any layer at any time right-click on the layer's caption and the Construction Tool will pop up to assist you with remastering of the component.

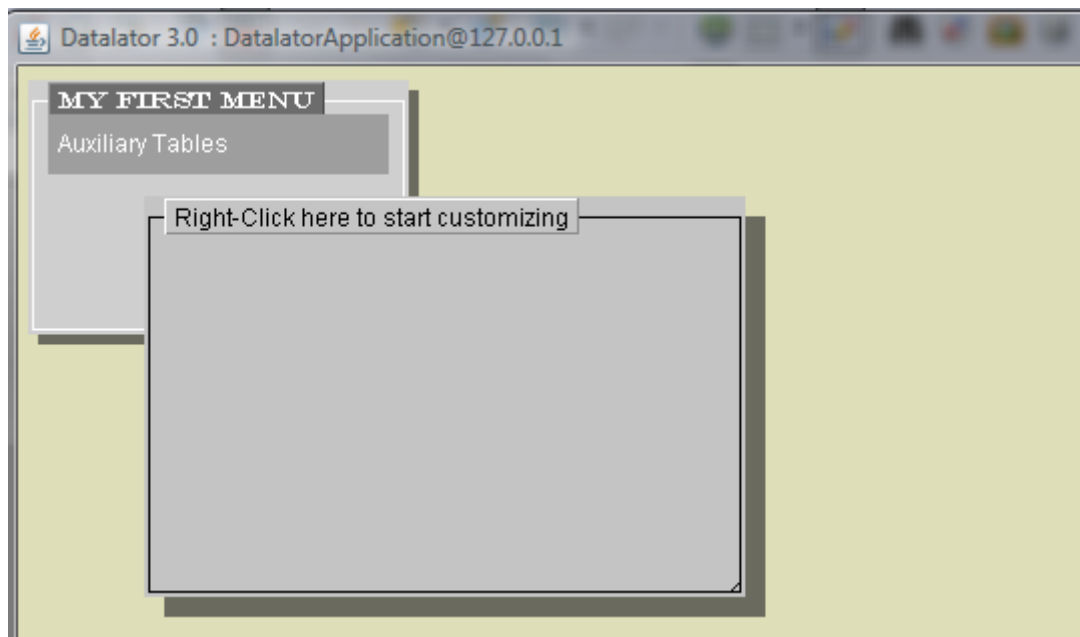
So far we've got our Main Menu online:



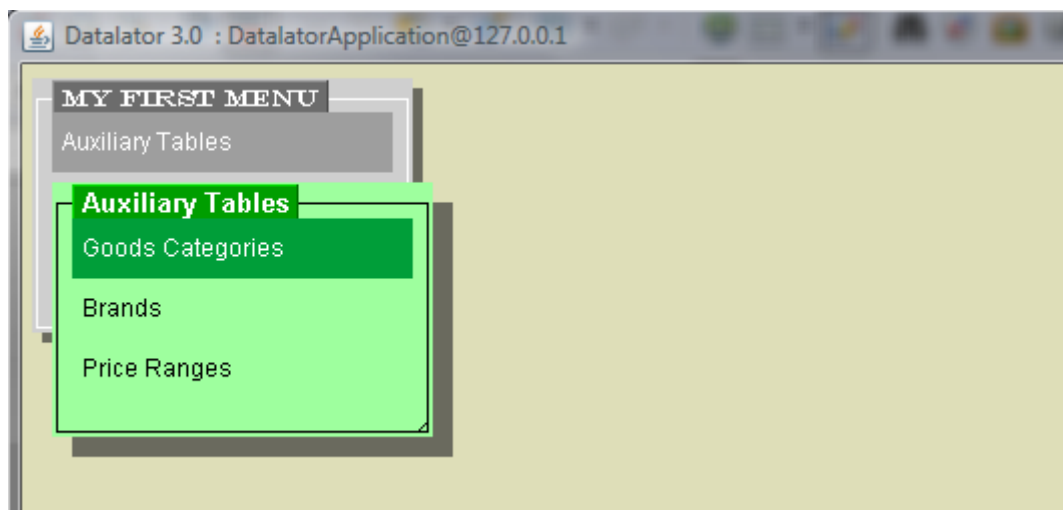
6. Building A Table.

Lets say we are building an application which implements a classified-like list of the advertized goods, indexed by goods category, brand, price range and the date the particular item got on the market. We have to build containers to keep and support those indexes; category, brand and price range will be designed as Table Layers to allow to input new records after the application is deployed. As the Calendar component ios already built-in, thus we do not have to build it as a layer.

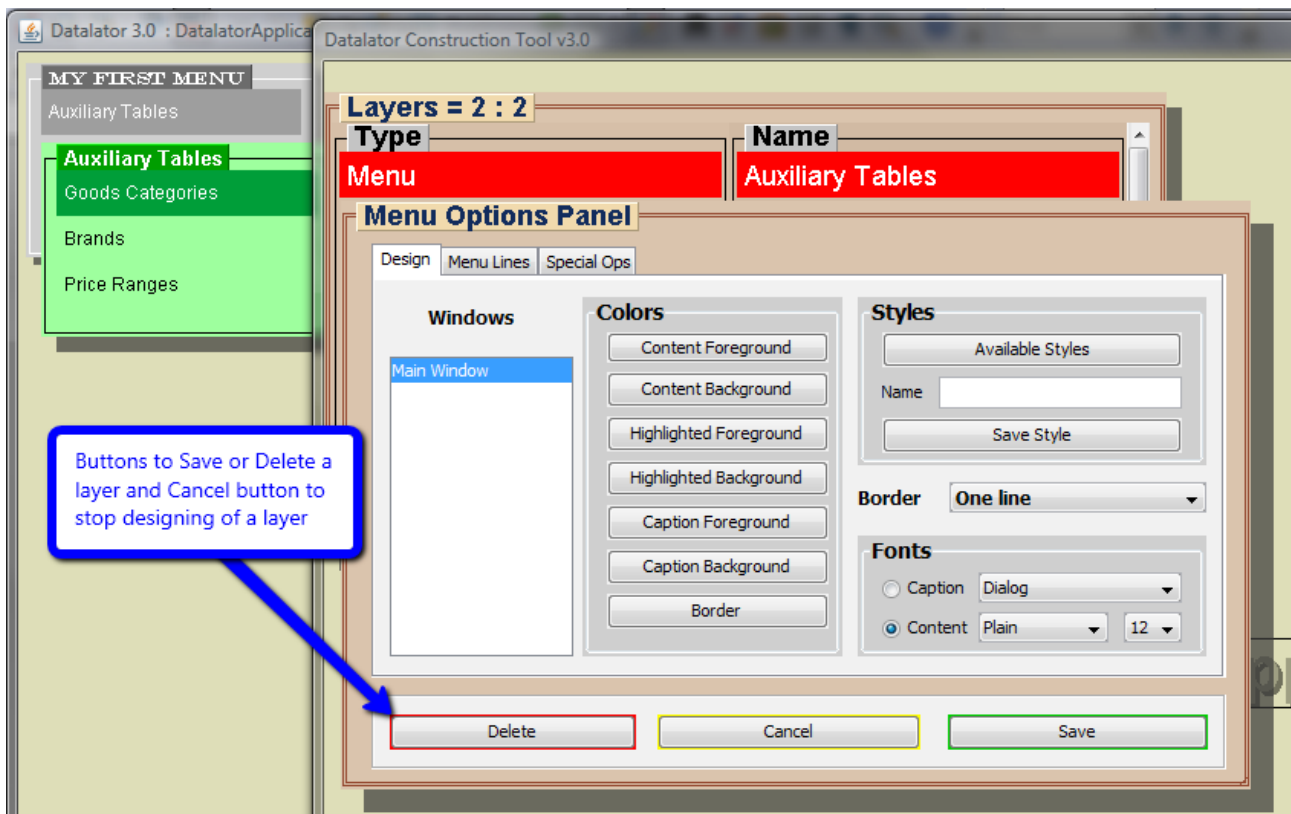
Before building the goal layer "Classified" it is advisable (but not mandatory) to start with auxiliary tables. To give access to those tables we'd have to create another Menu, named for example "Auxiliary Tables". To do this, double left-click on the "Auxiliary Tables" line (or use "down" and then "enter" keys) to "enter" that line. As nothing was connected to the line, we'll get another "dummy" layer to customize it into "Auxiliary Tables" Menu:



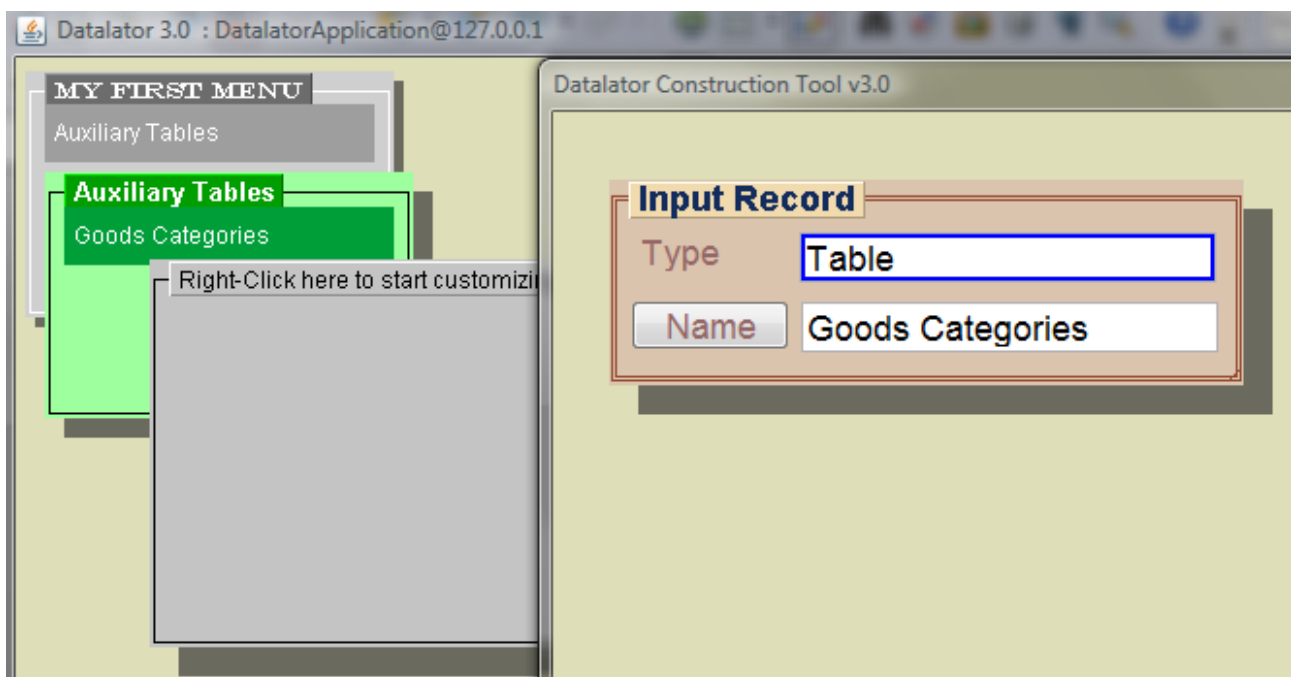
By repeating the process described in "5. Building a Menu" we can design and save that "Auxiliary Tables" Menu:



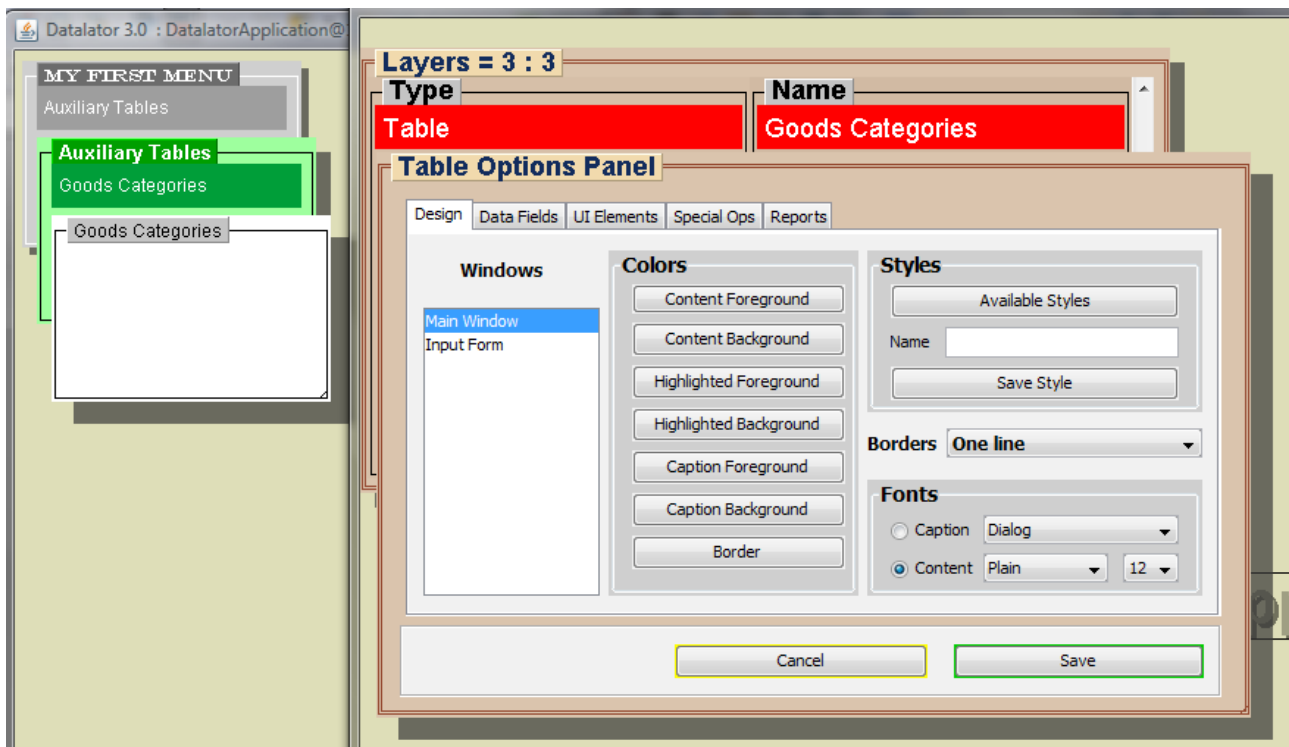
This way we've build, deployed and connected the "Aux Tables" submenu to the second line of "My First Menu". Any time we "enter" that line our green submenu will pop up and every time we "enter" a menu line with no defined yet link we get a "dummy" to be customized and attached. As a component could be created, it might be deleted as well, and that removes the link to that component, effectively replacing the component with the "dummy" layer, which allows to start the component development all over again, if needed:



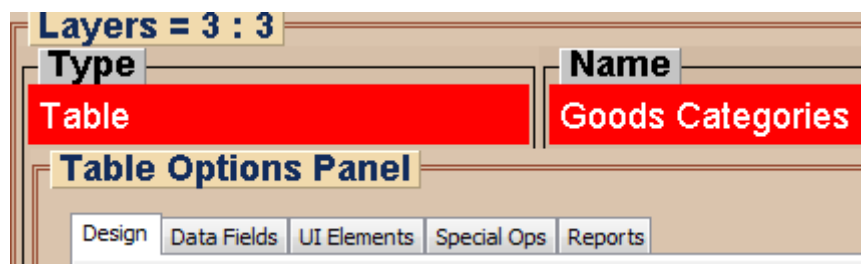
We are ready now to create our first "Goods Categories" table. Lets "enter" the first line of the "Aux Tables" Menu and set the type and the name of the new layer as shown:



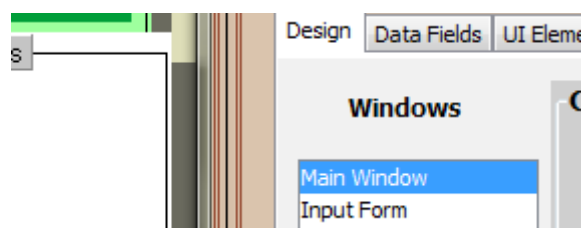
Click on the "Name" button and get the Construction Tool ready to assist in building a Table Layer:



We've got more tabs in case of a Table:



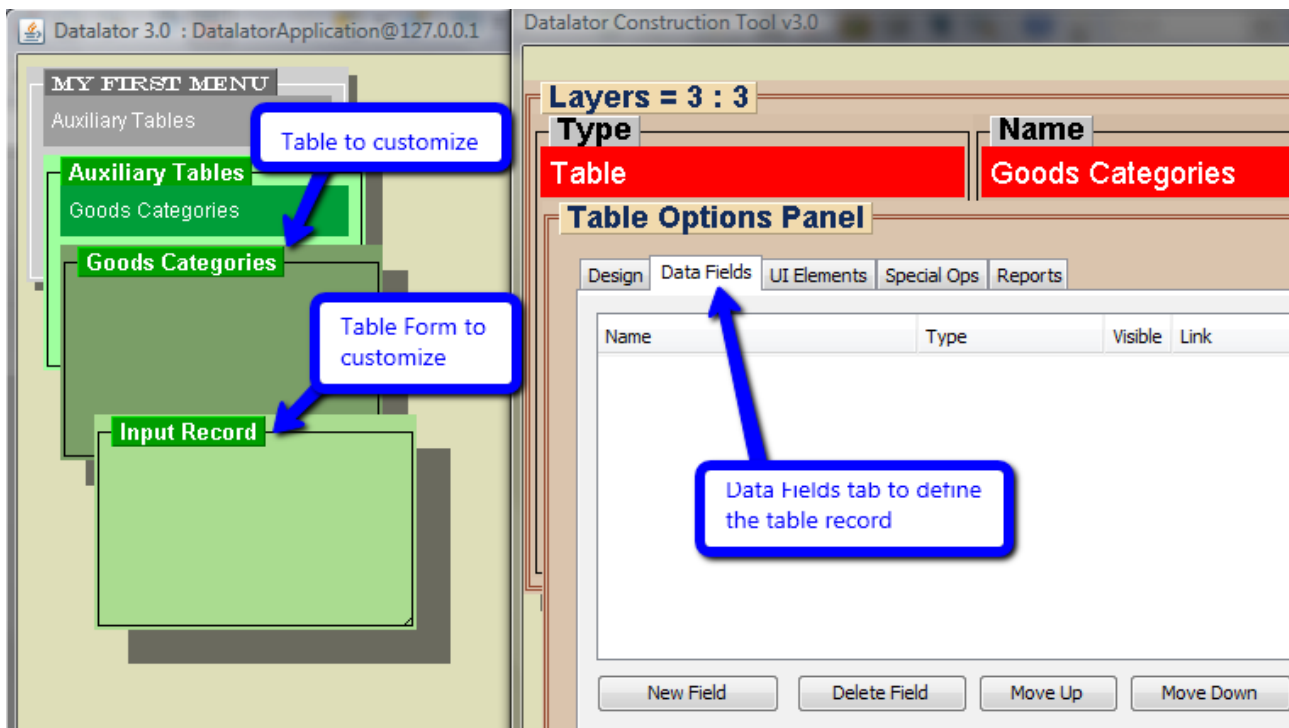
and more windows(there will be even more later) to work with:



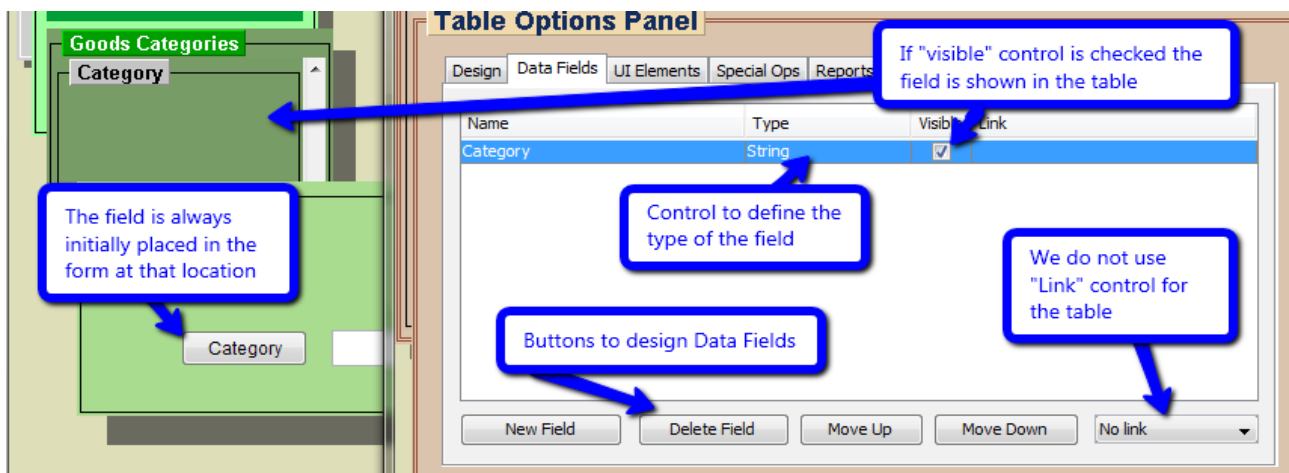
To customize a table we have to define the table fields to keep the data, adjust the table appearance, place additional controls if we going to program them("UI Elements" tab); we can export to or import from the design and/or content of a Table as a text file("Special Ops" tab) and we can create printable reports with "Reports" tab.

Note: it is advisable to start with setting up the background color of the Main Window or to use one of available styles – it will define the color schema of many elements of the Table layer.

After choosing the background we have to define data fields of the Table record, by clicking on the "Data Fields" tab. When "Data Fields" tab is chosen, the Table Input Form is shown, so we could define, design and place our fields conveniently:



We are going to design a table with only field(column) of "string" type to keep the names of categories of goods. And we do it like this, commencing by clicking on "New Field" button:



When a field is defined it is

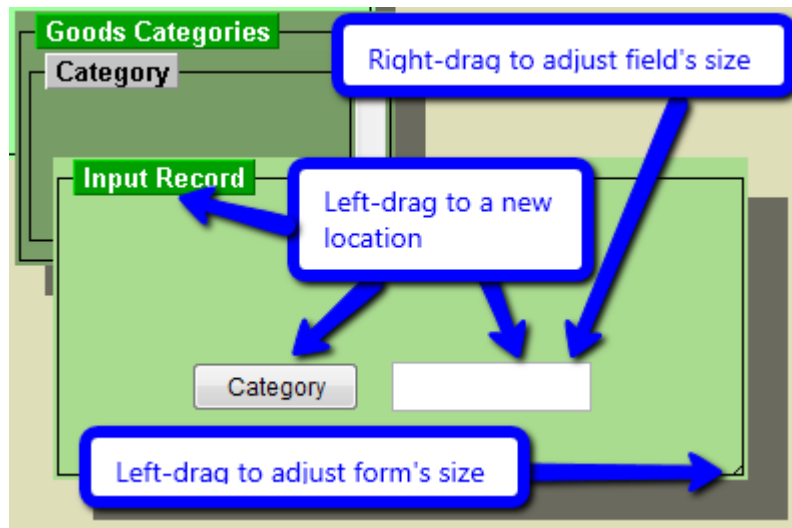
- presented as a new window("Category") in the main table window to form a table
- presented in the Input Form of the Table to provide a way to input and/or change data

Note: A new field is always placed 100 pixels below the top and 100 pixels right from the left edge of the Input Form; it makes sense to make the Input Form bigger by dragging it's right bottom corner.

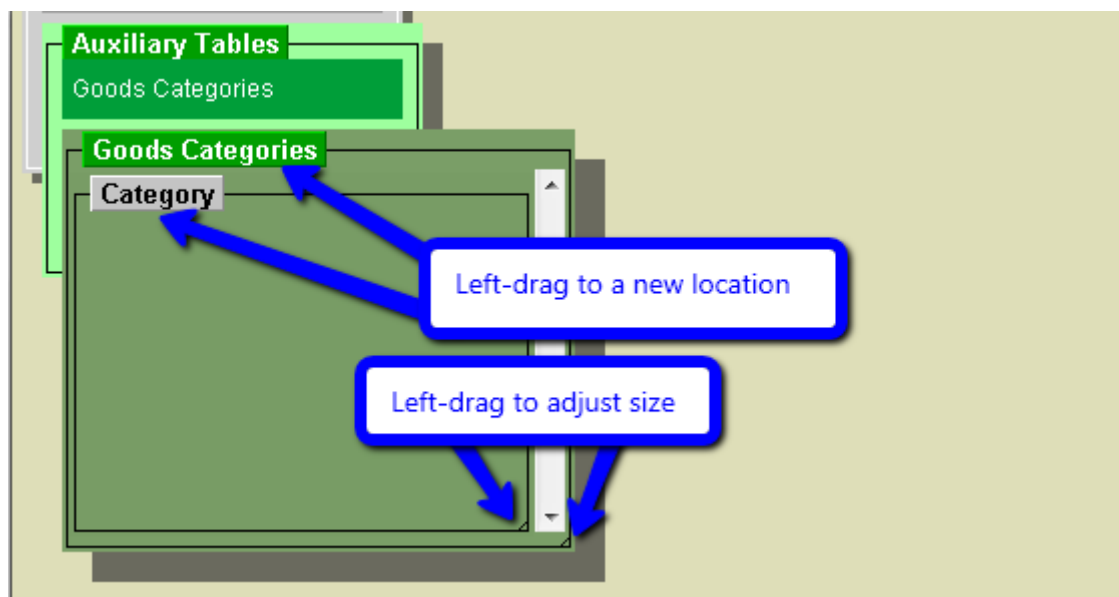
Note: A field in the Input Form could be left-dragged to the new location by the field's label or button.

Note: A text field of the field in the form could be left-dragged to the new location and it's size could be adjusted by right-dragging.

Next picture shows how to control the geometry of the form:



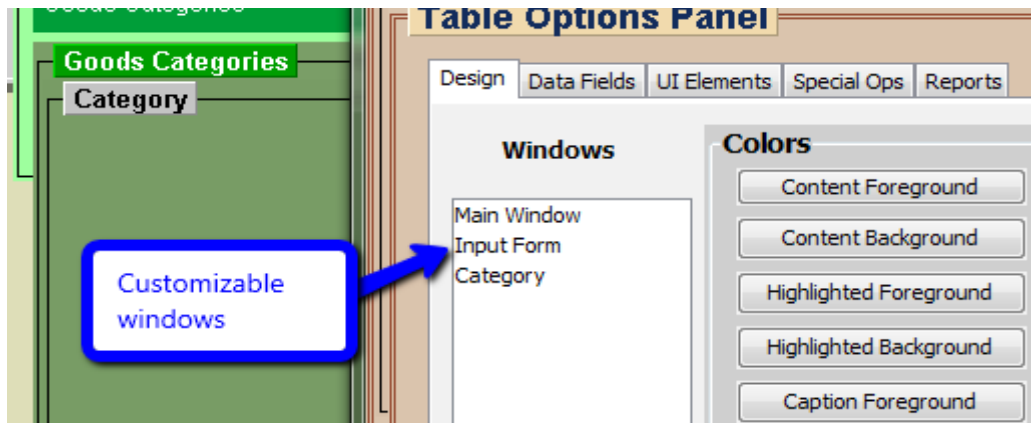
That is how to control the geometry of the table:



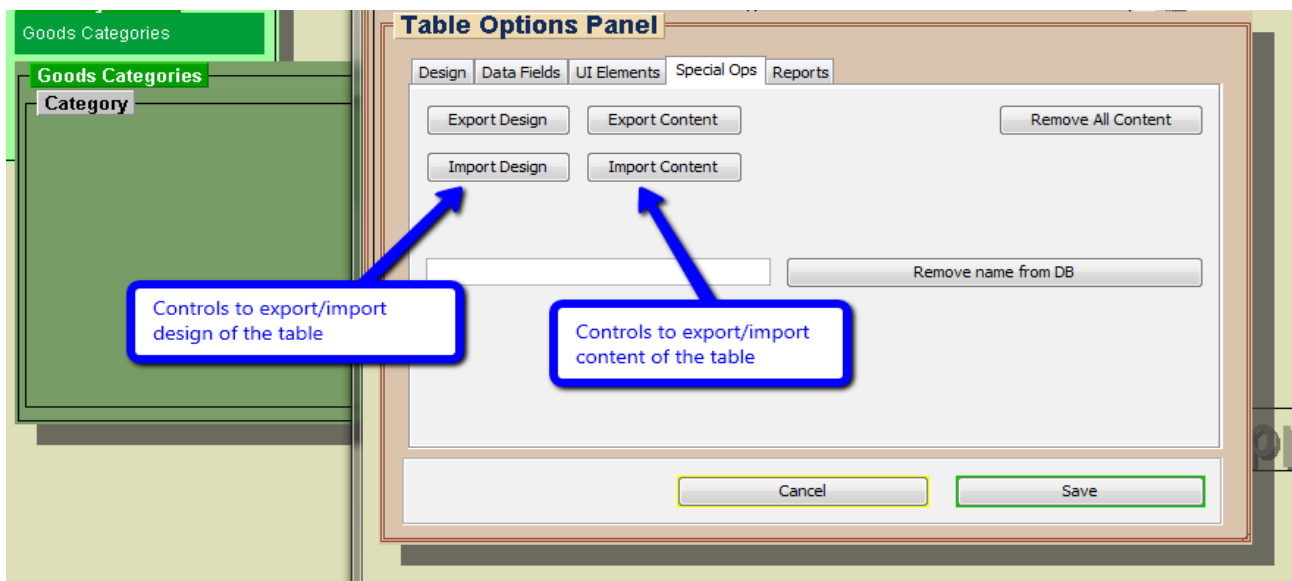
Note: Field Windows (like "Category" shown) **will** adjust their size if the main Table Window is resized by **LEFT** dragging it's right-lower corner and **will not** adjust if the main Table Window was **RIGHT**-resized. The later allows to create room to place some additional interface elements.

The Construction Tool will place a scroll bar at the right edge of the table and will use all the surface of the main window to place fields' windows. Later we learn how to adjust sizes of multiple field windows to create nice-looking tables.

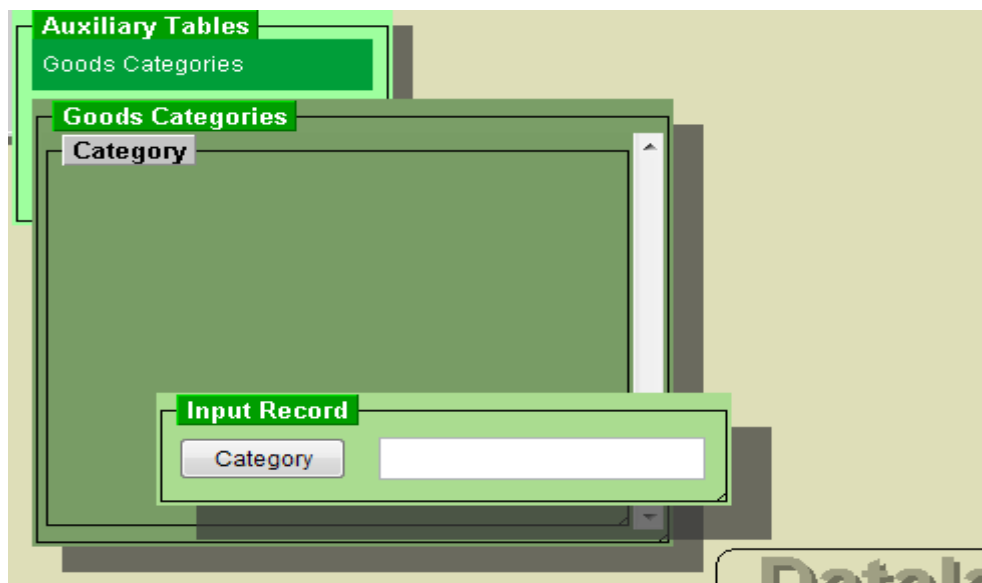
Note: All visible field windows and all additionally created UI elements are present now in the Windows list of the "Design" tab of the Construction Tool. They are all customizable individually.



Before saving the Table we should use the "Special Ops" tab of the Construction Tool to save the design of "Goods Categories" table in case the server is down or the session is expired or we just want to use the layer in another application later. We do it like this:



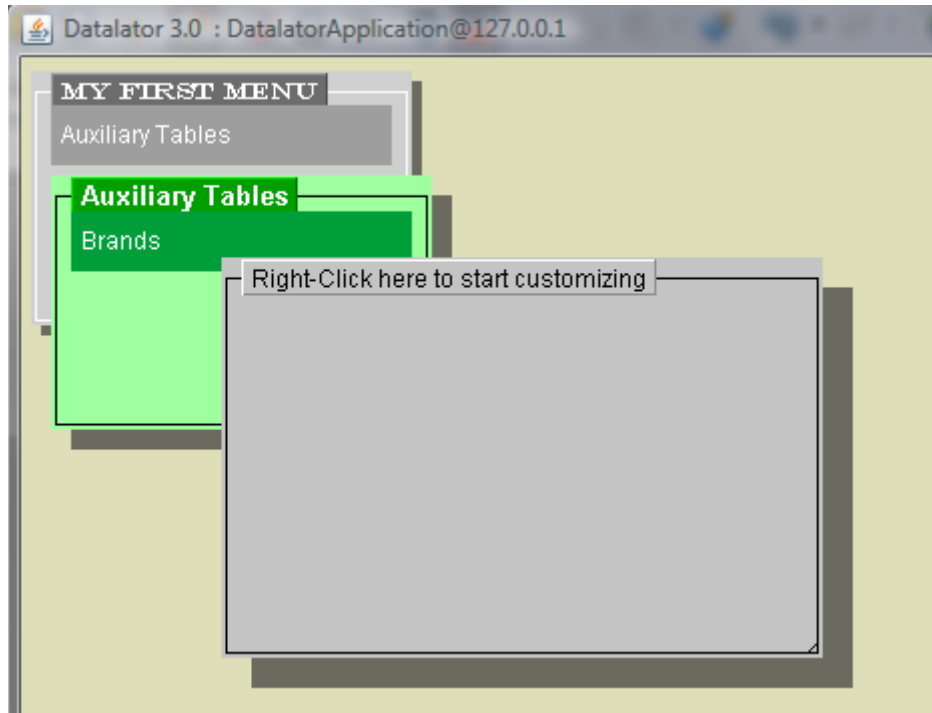
The final design of the simple "Goods Categories" Table might look like that:



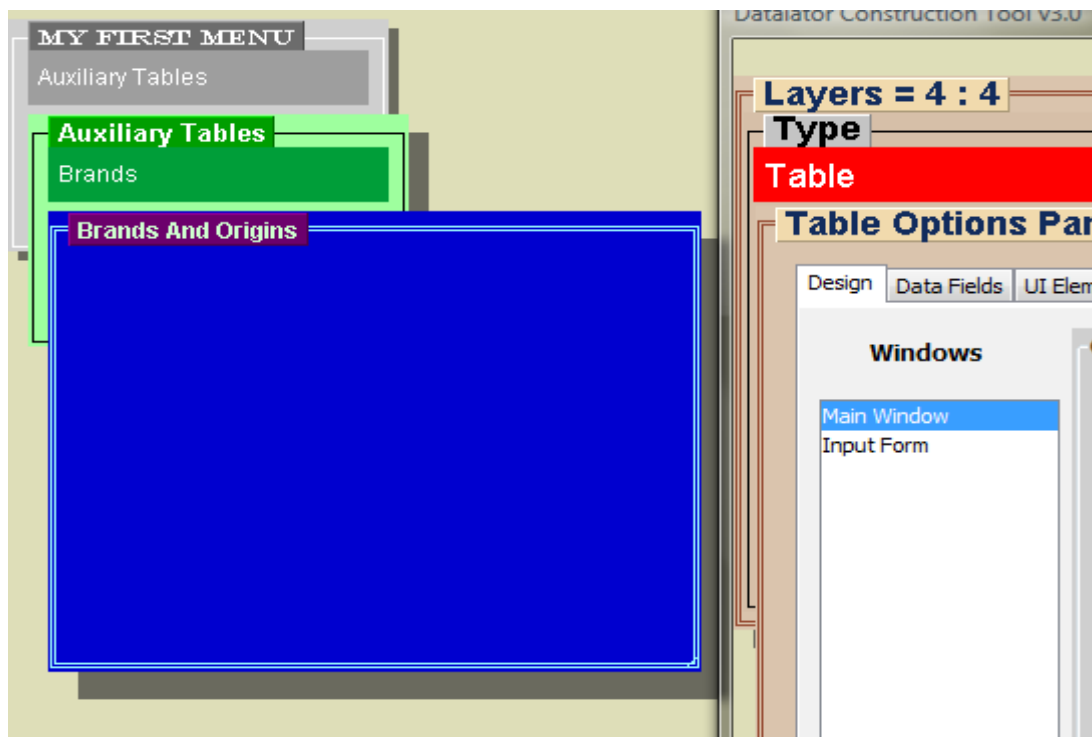
7. Building a More Complex Table.

It is good time now to build another Table – "Brands", which will consists of the "Brand Name" and "Country of Origin" fields just for the same of checking on some advanced topics. Thus repeat as in (6. Building a Table):

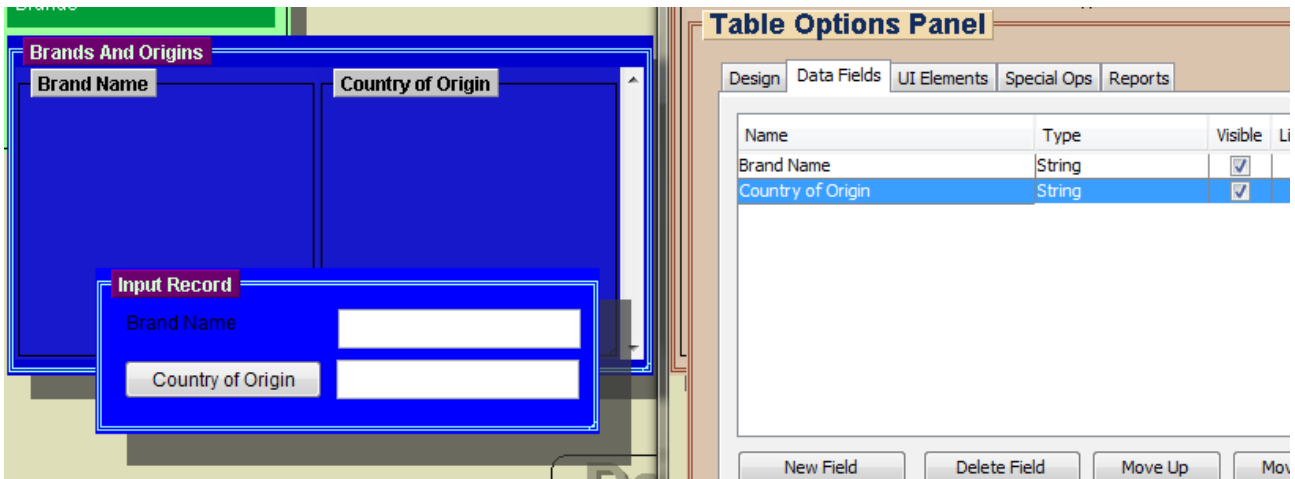
a) "enter" the line "Brands" of the menu "Auxiliary Tables":



b) right-click on the "dummy" layer caption, get the Construction Tool, set the type "Table", name it "Brands And Origins", expand the size and set the background of the main window:



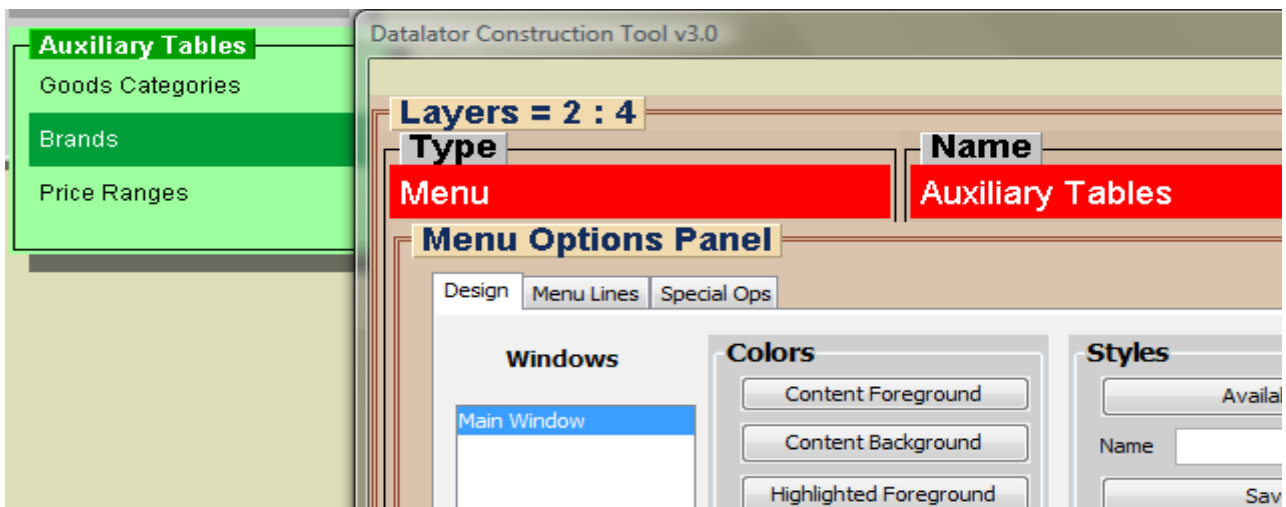
c) click on the "Data Fields" tab of the Construction Tool, input two fields and adjust geometry:



If we save the table now, our users will have to type in the country of origin every time a new brand is required; country names might be misspelled or inconsistent. To prevent it we will create one more auxiliary table, containing names of the countries which will assist in defining the value of the "Country of Origin" field.

d) export the "Brands And Origins" table design("Special Ops" tab) and save the table

e) "escape" to "Auxiliary Tables" menu and right-click on its caption:

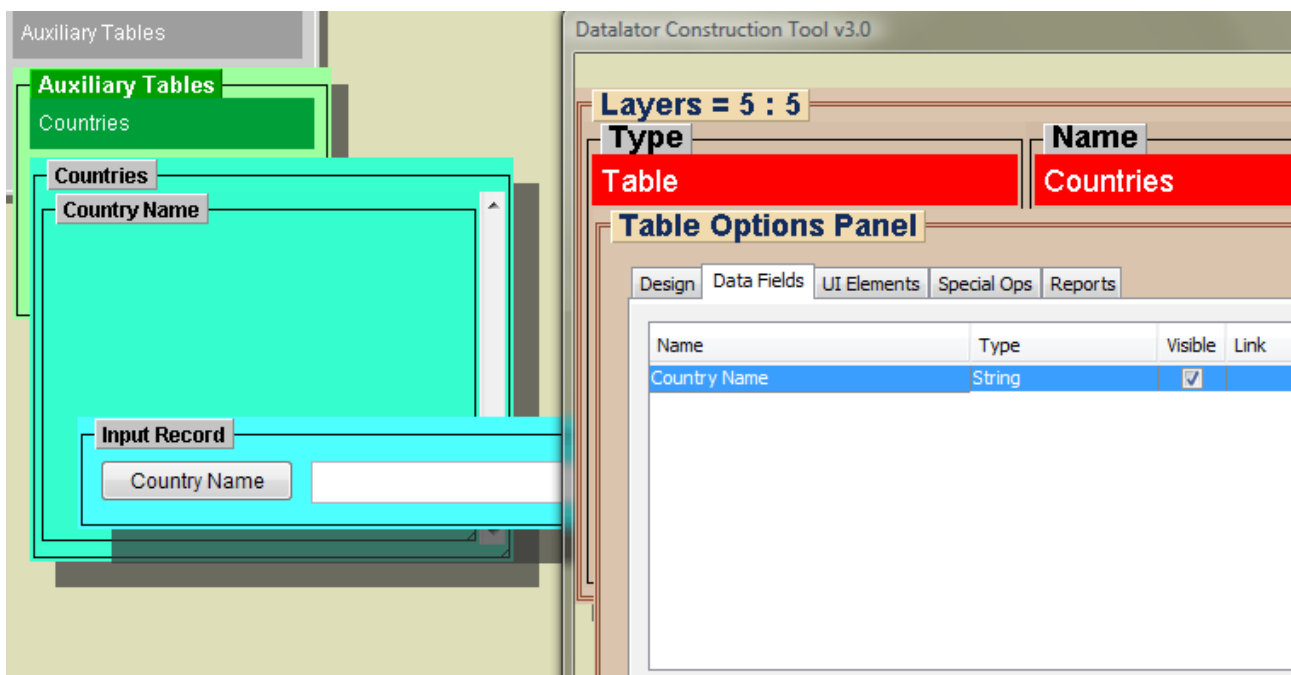


As we see, the Construction Tool now assisting us to redesign an existent layer.

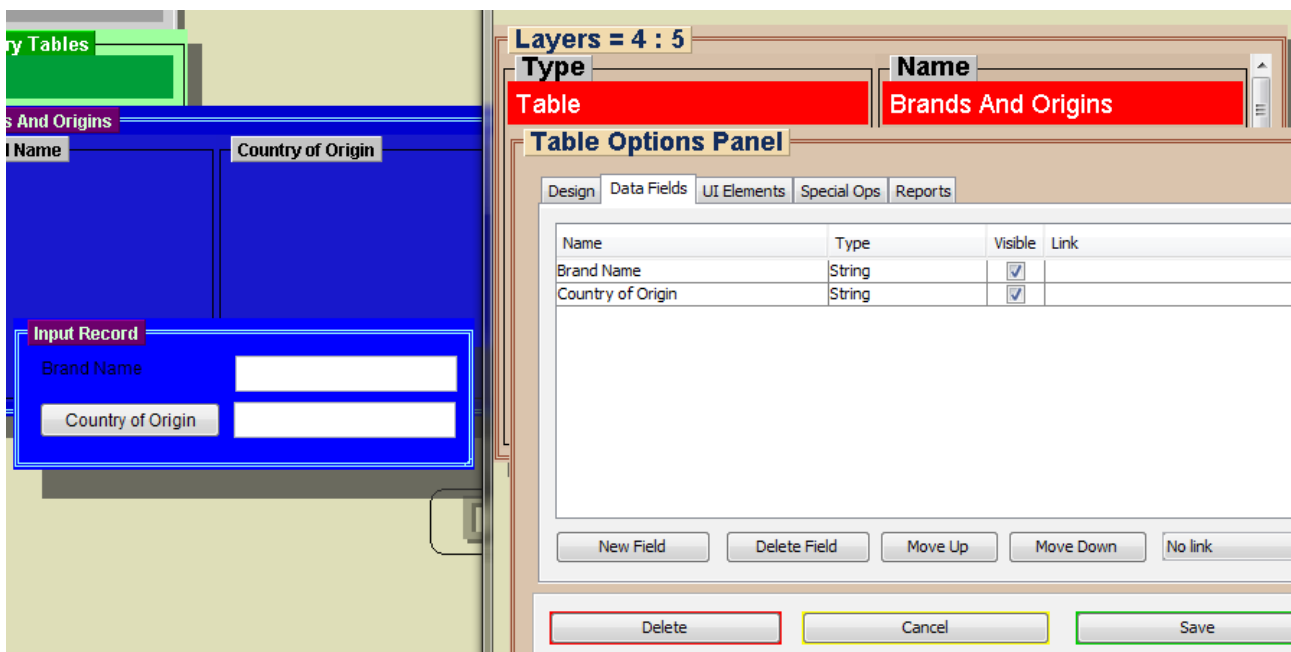
f) click on the "Menu Lines" Tool's tab, add a new line "Countries", extend the height of the menu window, export the new design and save the menu:



g) "enter" the "Countries" line and build the new table "Countries" with the only field "Country Name" of string type, like that:

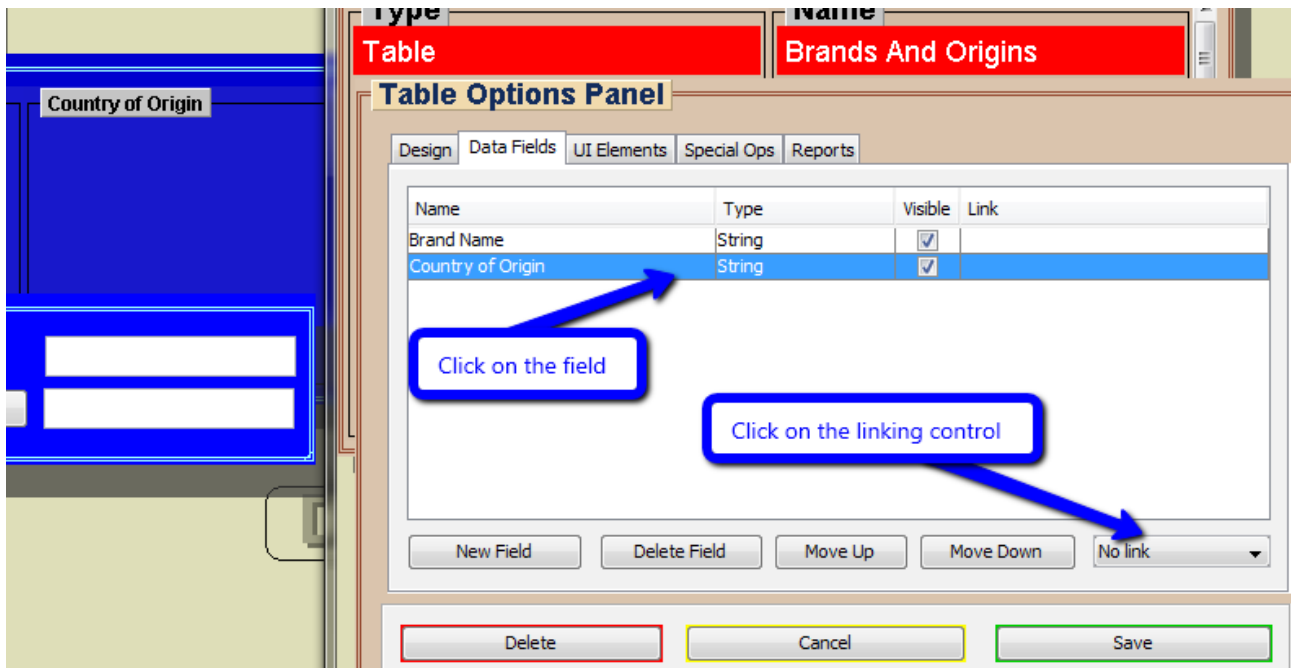


h) export the design, save the "Countries" and navigate back to the "Brands" table; start the Construction Tool, go to "Data Fields" tab:

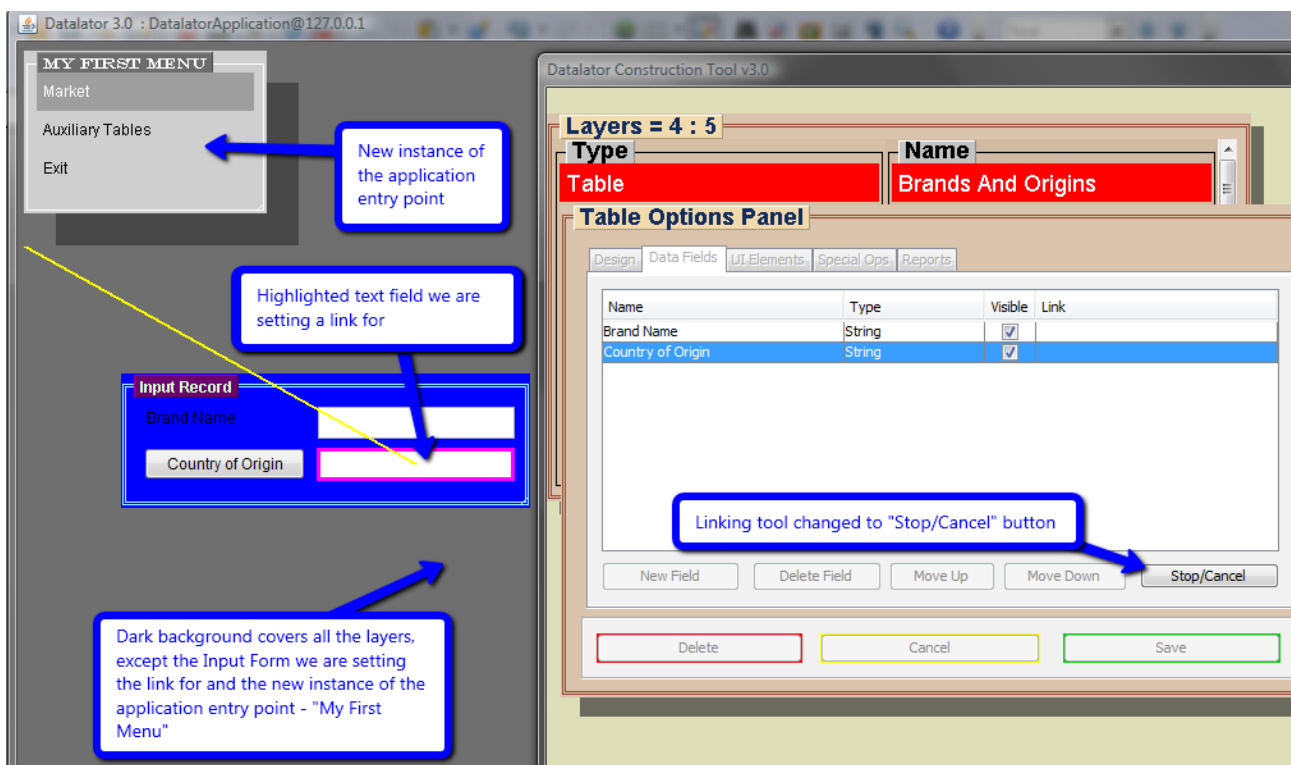


The "Link" column of our Data Fields table is empty. By "linking" the "Country of Origin" field to the "Countries" table's field "Country Name" we make the "Country" table to assist to input the field "Country of Origin" of the table "Brands".

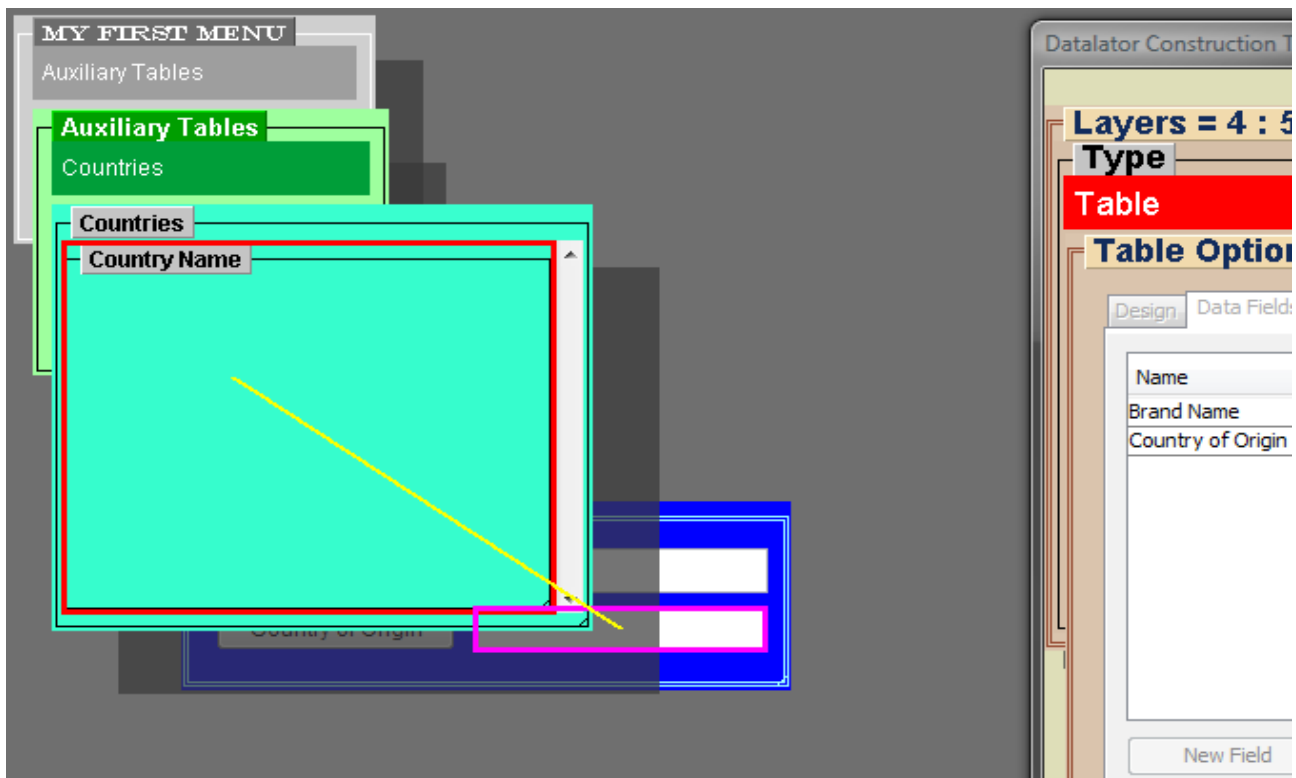
i) click on the "Country of Origin" field in the Construction Tool and click on the linking control:



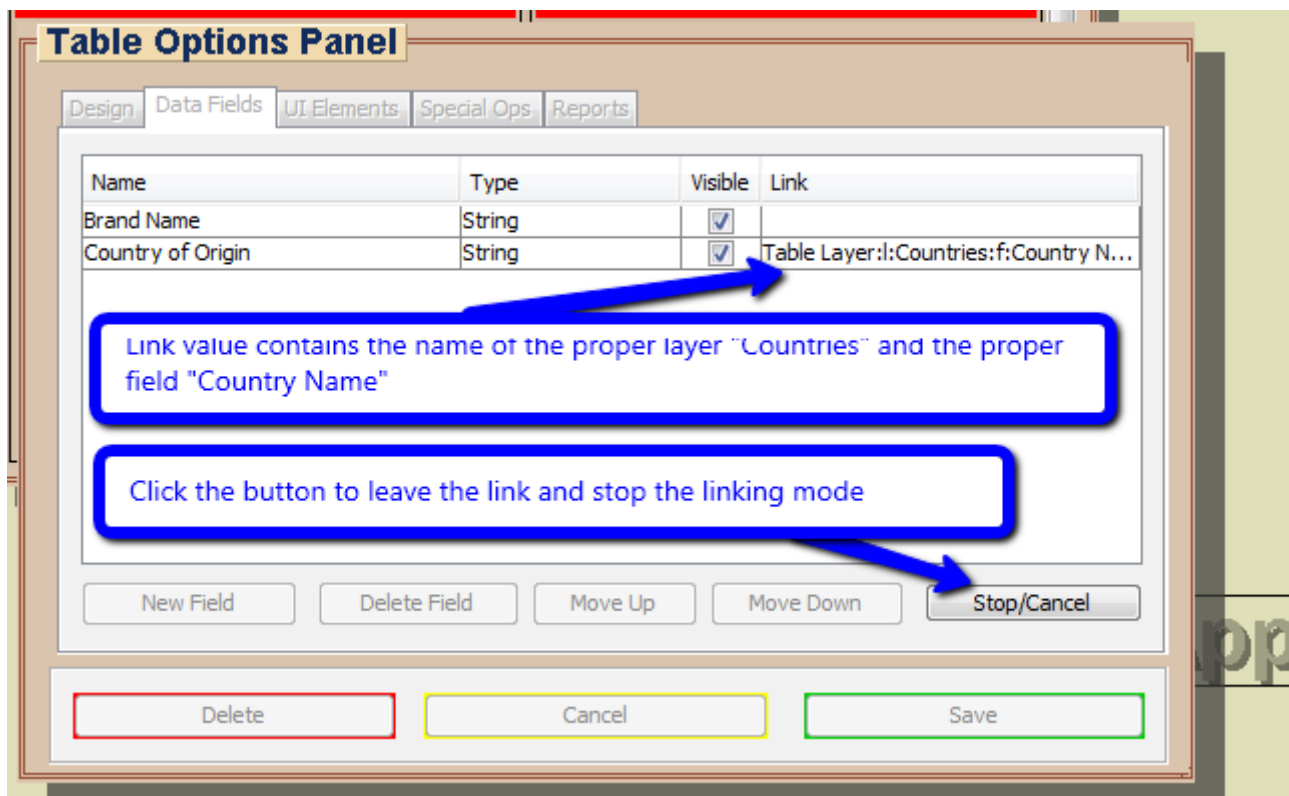
j) choose the "Set The Link" line in the linking control, that starts the linking process like that:



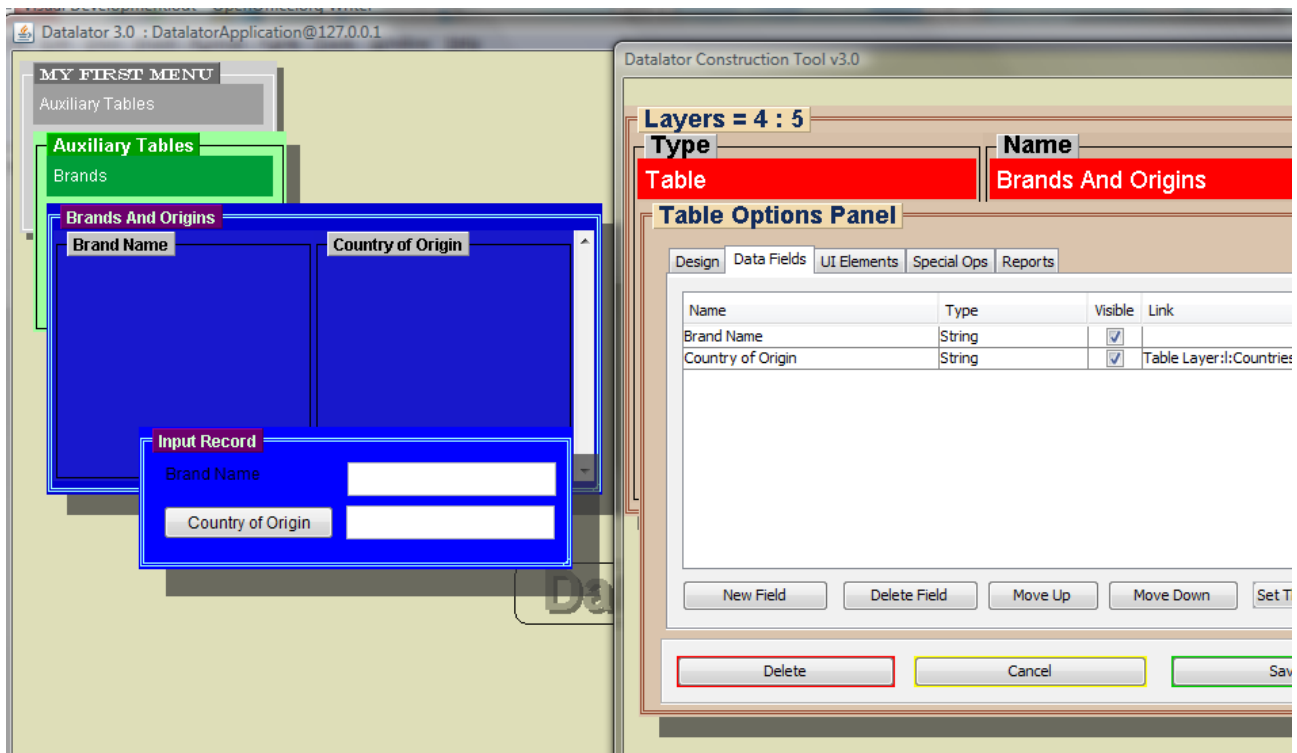
k) navigate to the table "Countries" from the new instance of the "My First Menu". The Tool highlights the input field to be connected (pink color), highlights the possible connecting target (red) and hints the connection with the yellow link



l) click anywhere on the highlighted "Country Name" window and make sure your click was accepted – the "Link" column in the Construction Tool has to be defined like this :

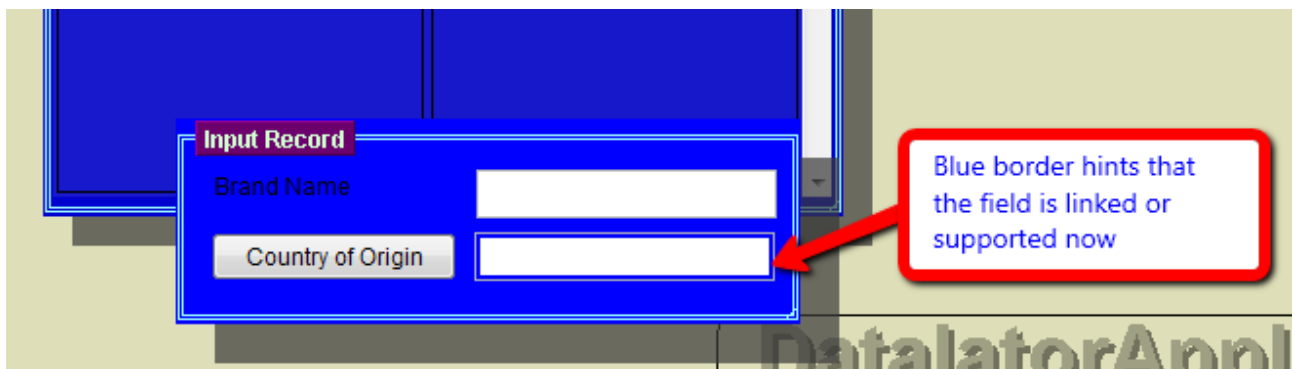


m) click the "Stop/Cancel" button of the Construction Tool; the link is defined, extra layers are gone and things are back to normal:

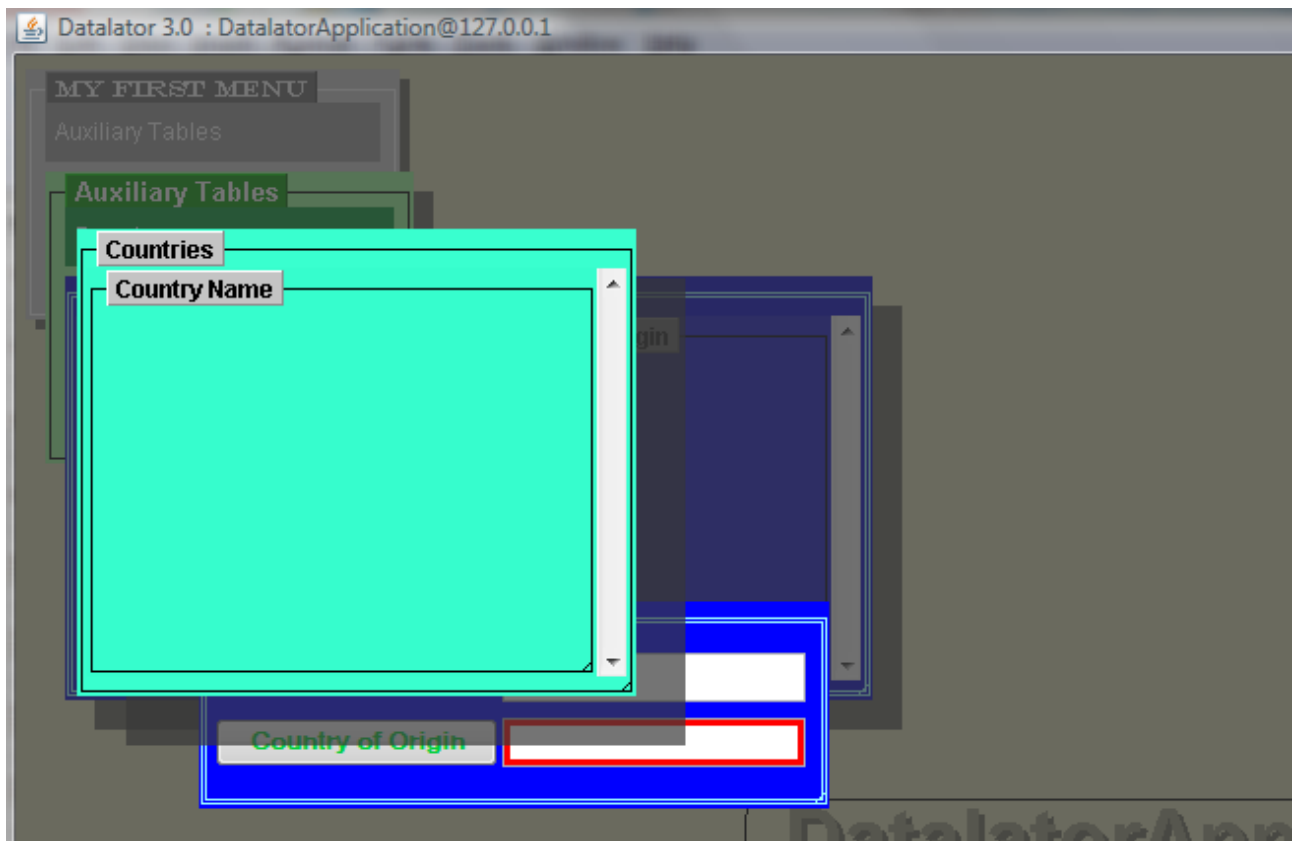


Note: Design export process leaves all layer's links blank, because the layer might be used in a different application and the links might be rendered broken.

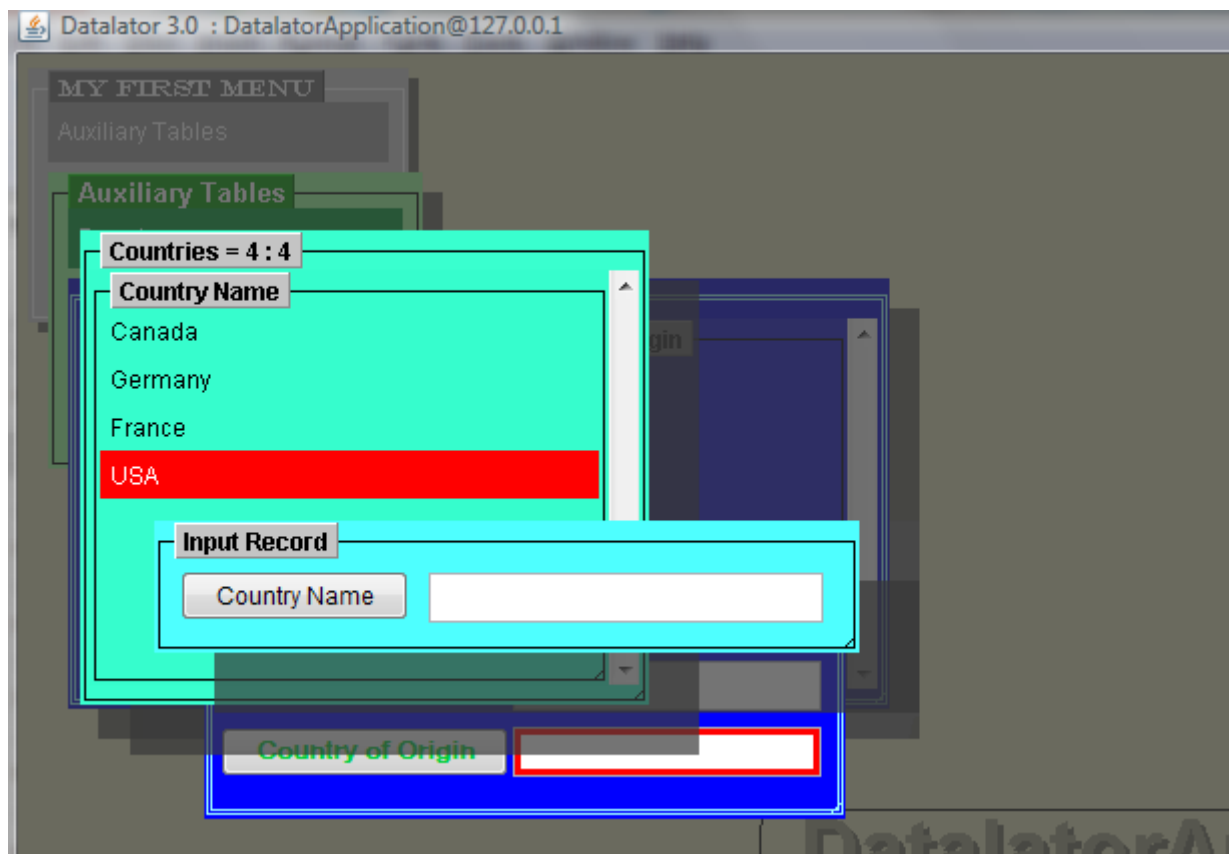
Just save the layer and open the Input Form again:



We see the extra border marks the linked text field, think of it as of a link on a web page. Click on that field or CTRL/Space it and the supporting Table "Countries" will pop up, marking the text field with a brighter(now red) border and dimming the background, like this:



There are no records in the table "Countries" yet, so let us input some. Right-click on the "Country Name" window or use an F5 button to open an Input Form and use it to type in a few countries:



Close the Input Form, click on a "USA" record and you can see the "Country of Origin" text field value is set to USA:

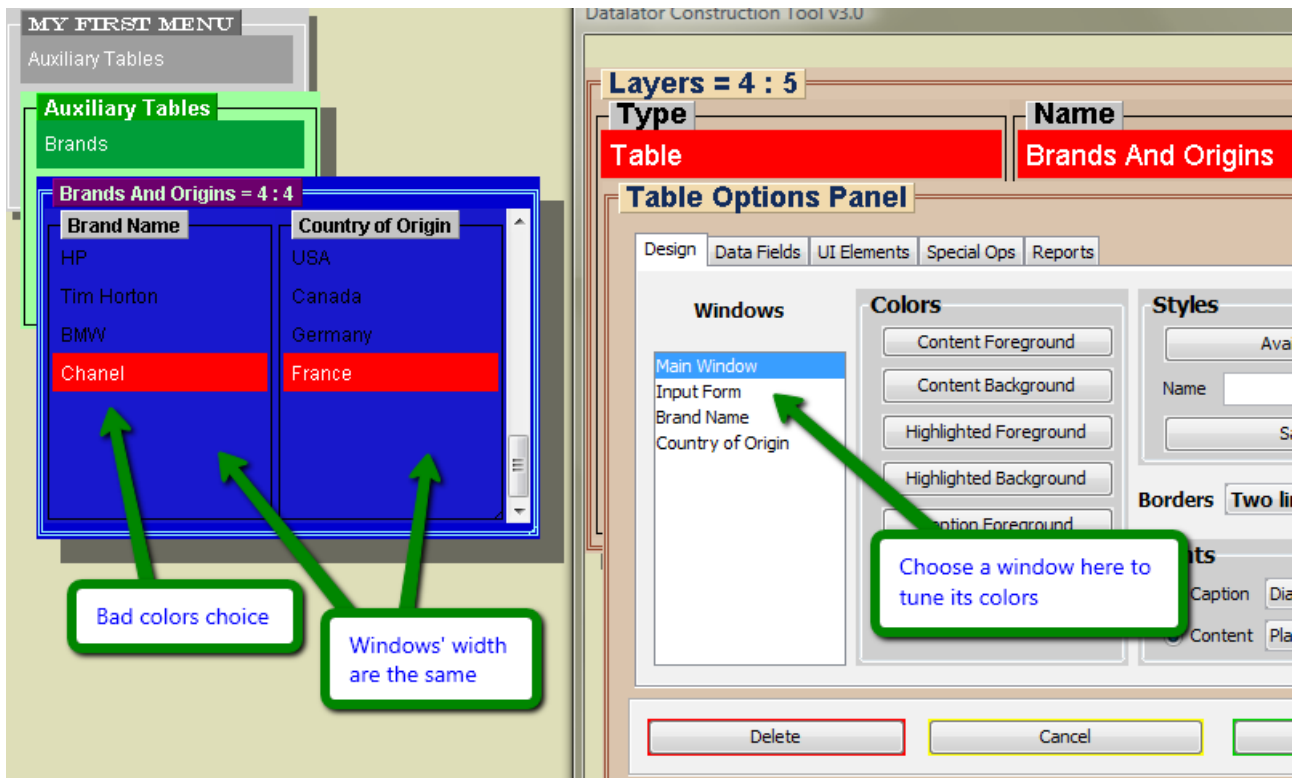
The screenshot shows a form titled "Brands And Origins" with two main sections: "Brand Name" and "Country of Origin". Below these sections is an "Input Record" dialog box. The "Input Record" dialog has two fields: "Brand Name" with the value "HP" and "Country of Origin" with the value "USA". The "Country of Origin" field is highlighted with a green border, indicating it is the active field.

Click on the "activation" button "Country of Origin" and we have the record inserted into the table:

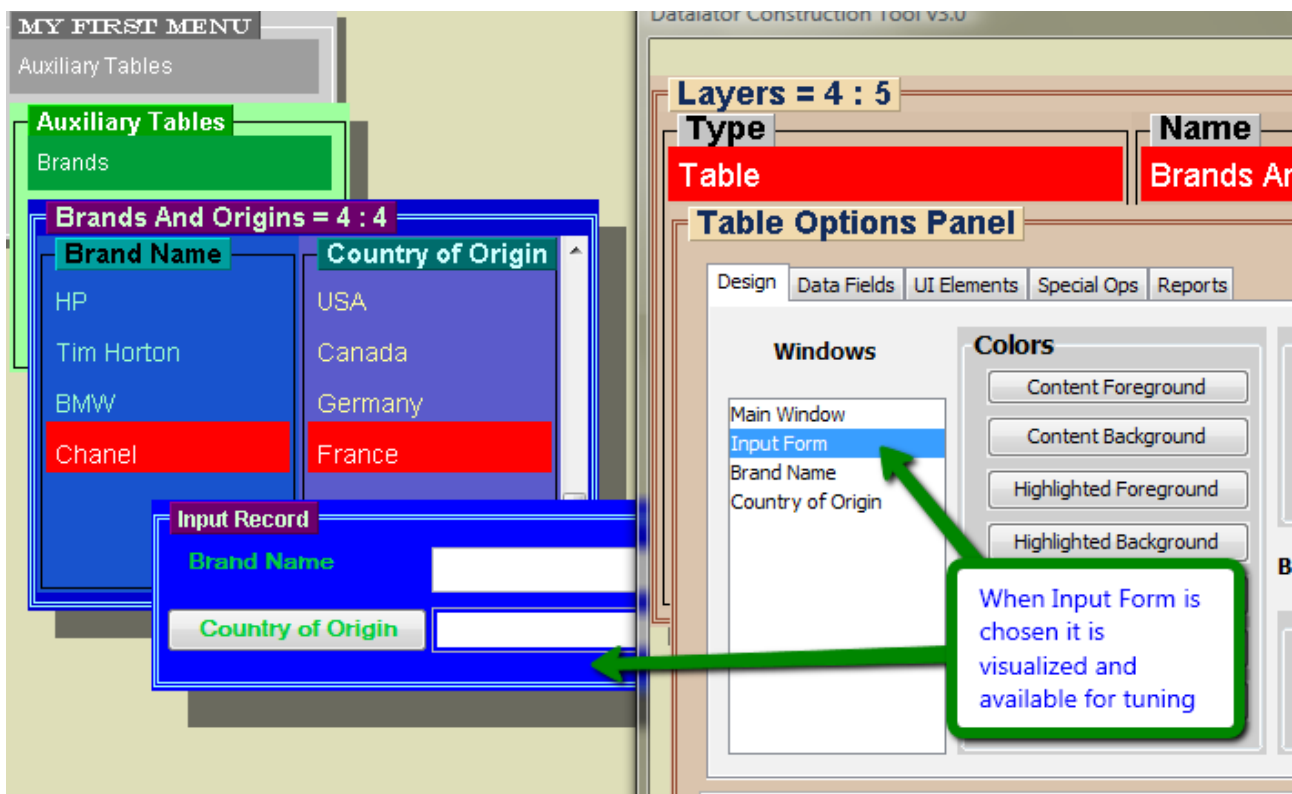
The screenshot shows the same "Brands And Origins" form, but now the "Brand Name" and "Country of Origin" fields are populated with "HP" and "USA" respectively. The "Input Record" dialog box is still open, but the "Country of Origin" field is now empty. The "Brands And Origins" caption is updated to "Brands And Origins = 1 : 1".

That linking technique will be also used when designing the Filter Form of the Filtered Table.

Now we'll learn how to adjust design of the table with live data in. Let's input a few more records, then close the Input Form and right-click on "Brands And Origins" caption to fix a few deficiencies with colors and fonts, choosing an element to tune from the "Windows" list of the Construction Tool :

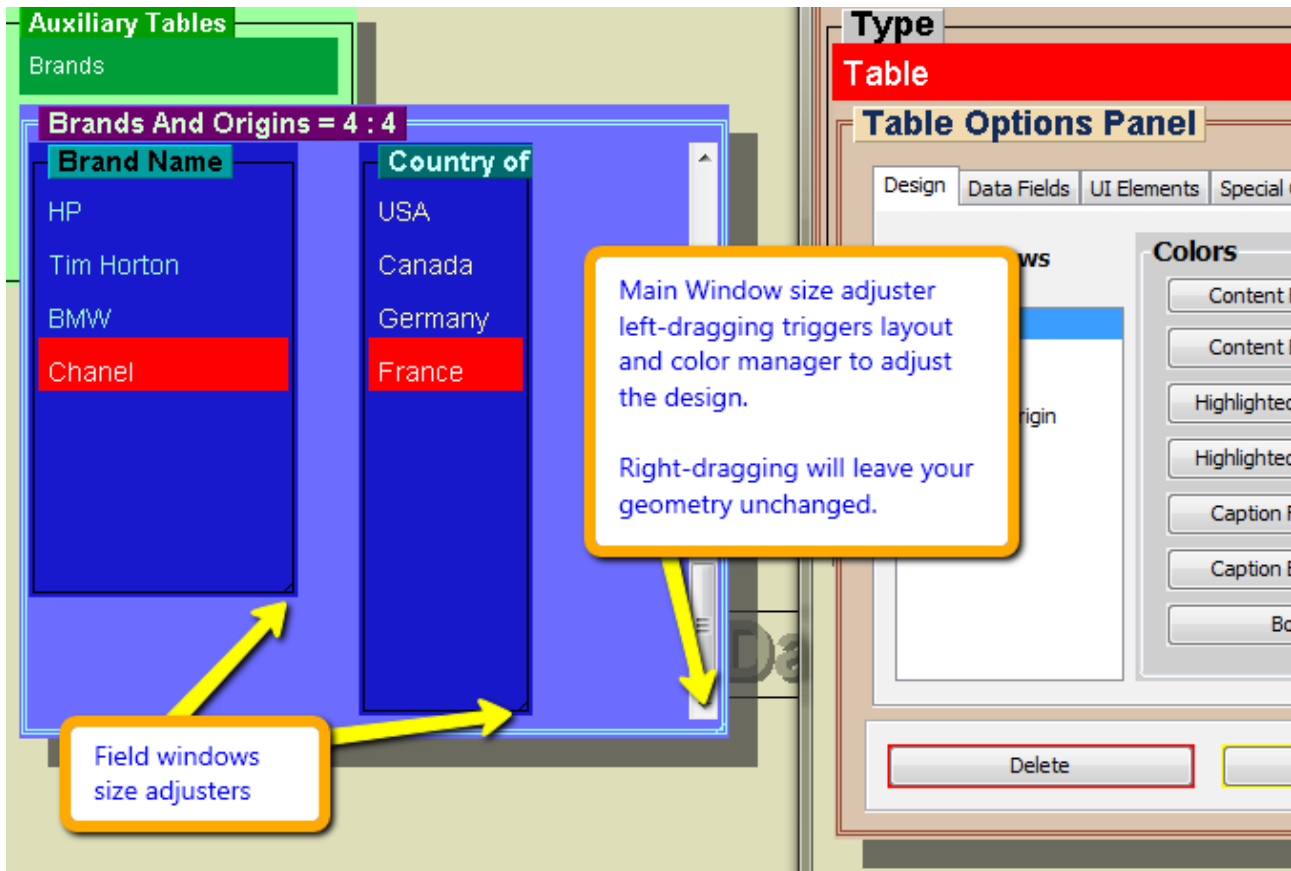


We could adjust the colors and fonts like that:

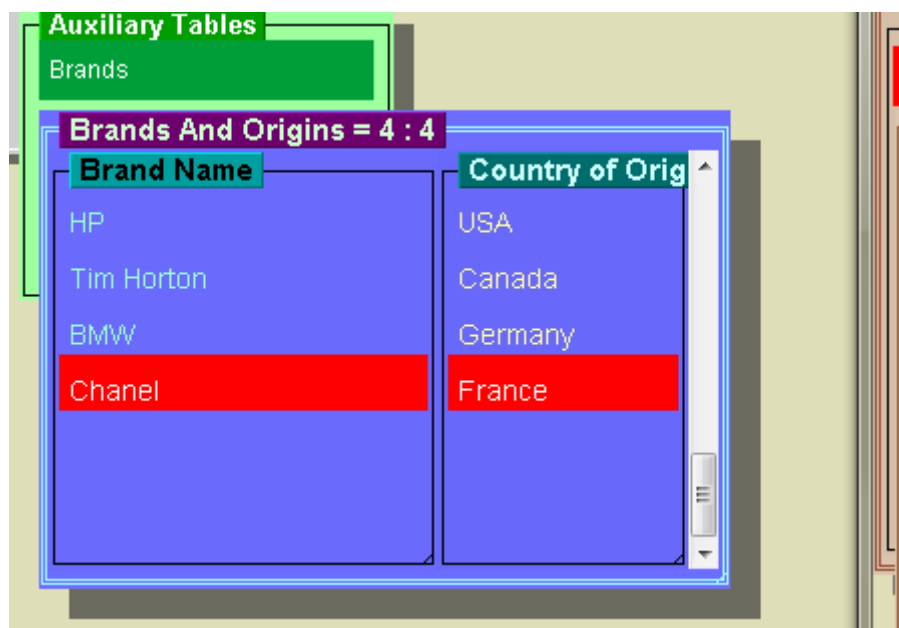


Without implying to be a design masterpiece it shows how you could possibly highlight the important information in the table and dim less valuable columns. Try to google "color matching" for better color choices and remember that you can always return to some time-proven color schemas for tables (dull gray), like in MS Windows, if you like it.

One more technique to show is how to set the widths of the table columns. Try to drag the right-bottom corners(size adjusters) of the field windows individually and then left-drag for just a bit the right-bottom corner(size adjuster) of the main window to trigger the layout manager intervention:

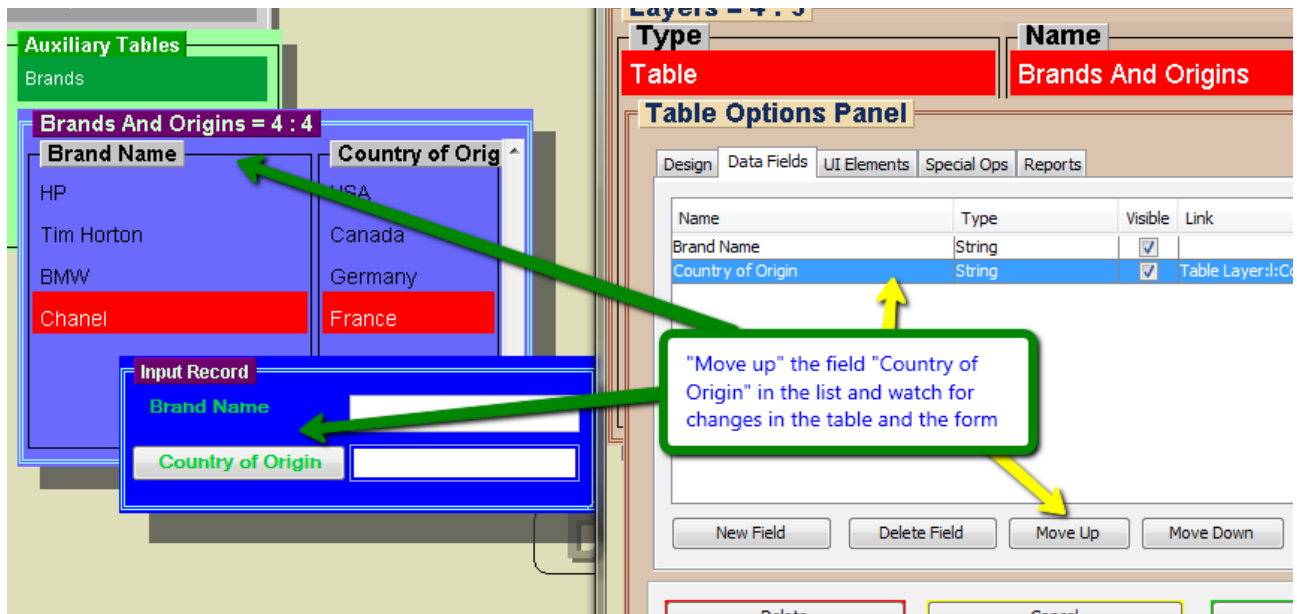


The adjusted layout will look like that:

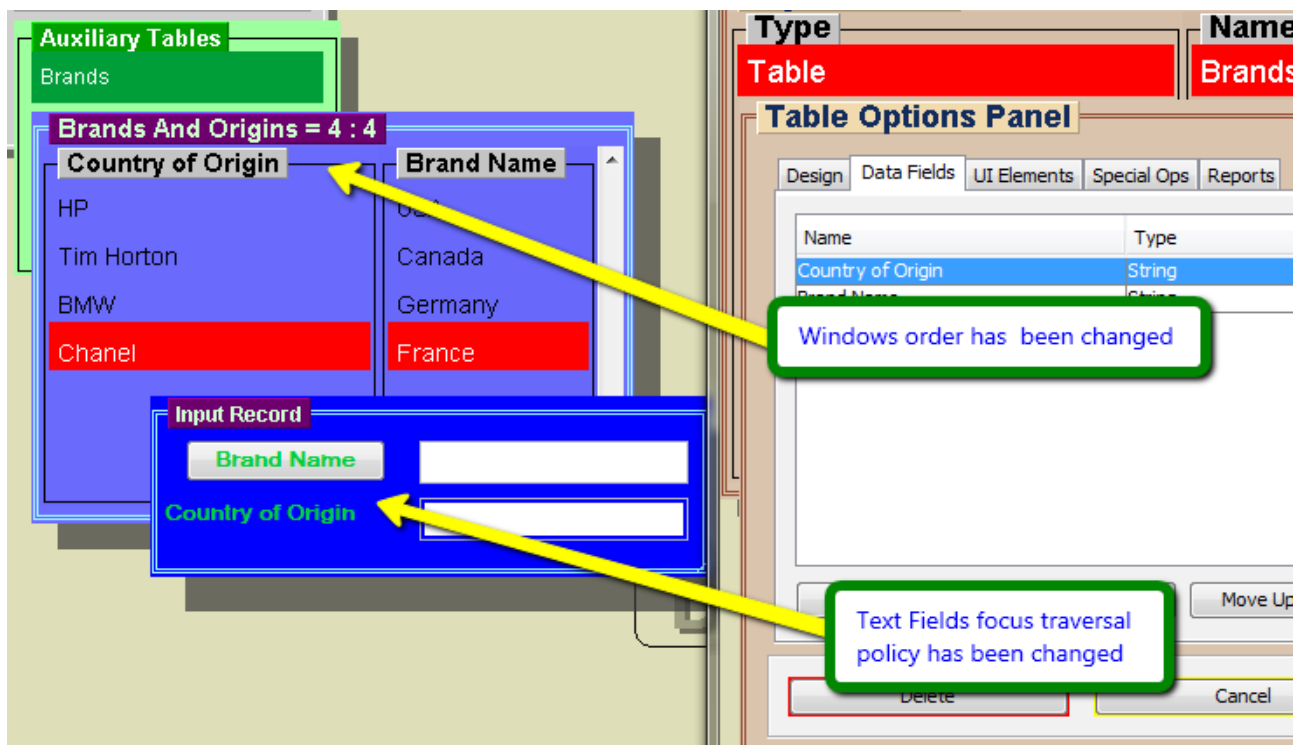


which is not flawless but educational – imagine you tune the geometry of the 10-fields-wide table. Just remember to hand-color them AFTER you satisfied with dimensions.

The last trick of tuning the tables is to change the fields order :



Hit the "Move Up" button and see the layout changes plus feel the text fields focus traversal policy changes in the Input Form:

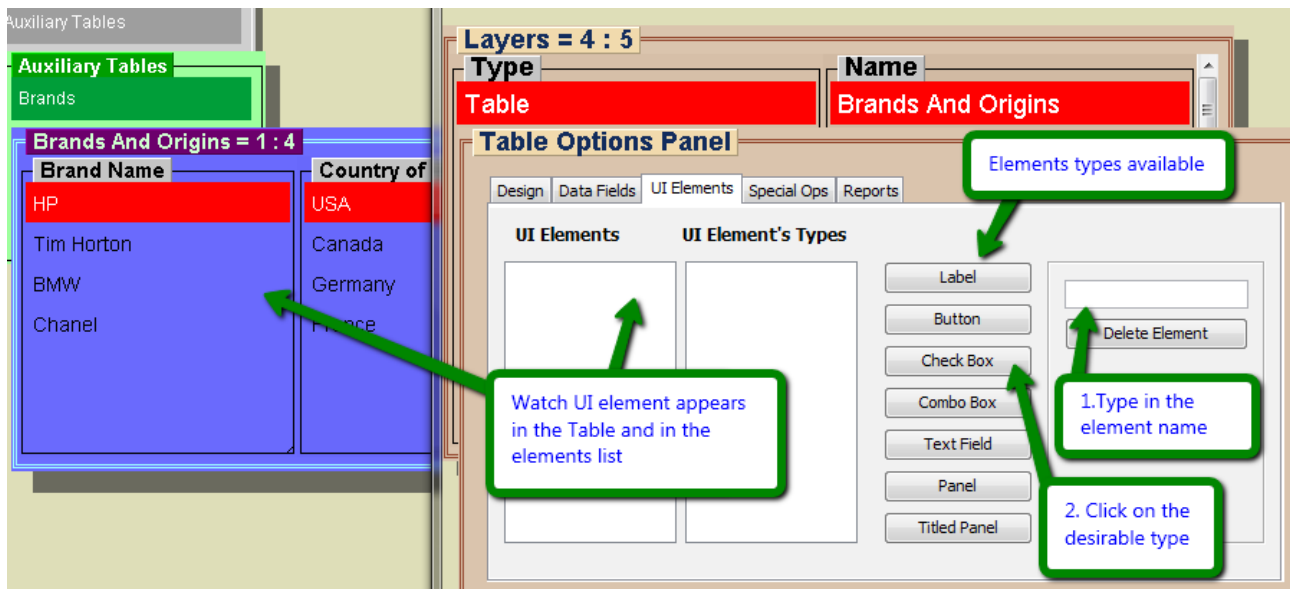


That mechanics helps to change the tables when testing your applications. The cheaper it costs to make changes the more changes it will provoke. Considered to be a Good thing in the best spirits of agile development methodology.

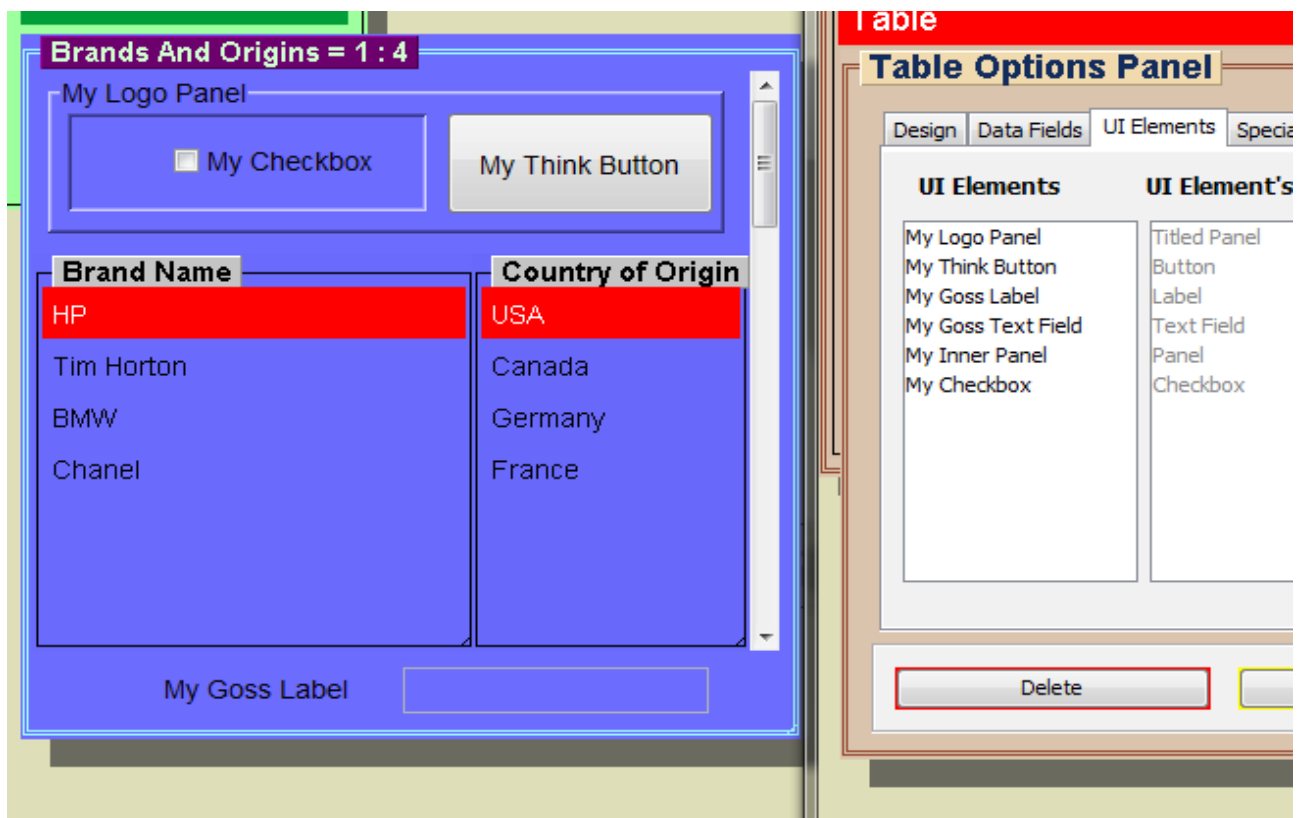
Note: current Datalator version cannot change structure of the populated tables. Export the content, export the design, delete all content, delete the layer, start making it from scratch but instead import the design. Change the table, save it and now import the data back. Not automated yet.

8. Building A More Complex Table: UI elements and Reports.

Lets see how we can enhance the table and it's form with UI elements. Right-click on the caption of the "Brands And Origins" table and click on "UI elements" tab of the Tool:



Lets place some UI elements on the form:



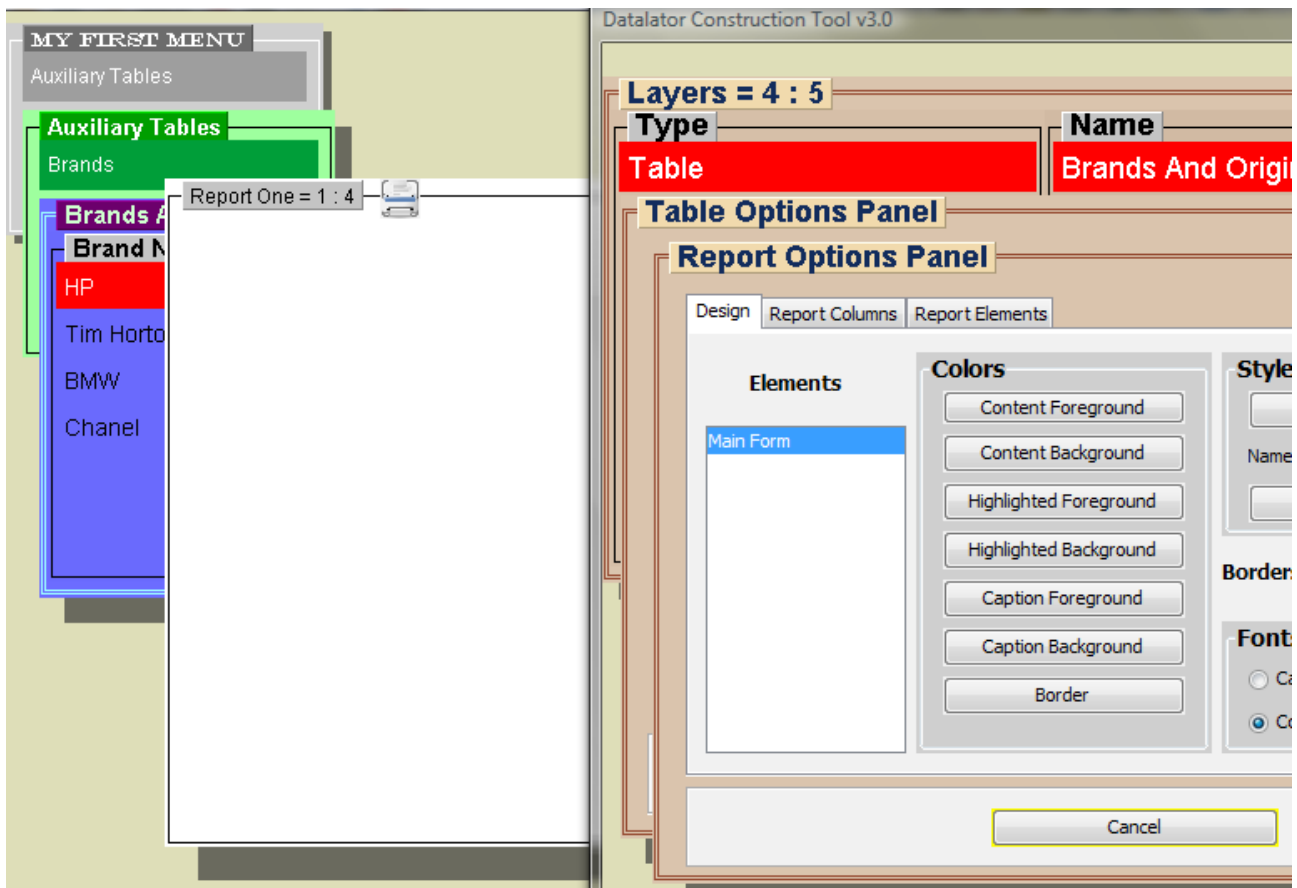
Elements are there and might be useful if we are going to add some functionality by programming them later. They are more useful when structuring the Input Form fields. And the real-life example is right here:

Input Record			
Trip		Customer	
Trip		Customer	
Driver		CSR	
Truck		Booked On	Customer #
Trailer		Customer Rating	
Dispatcher		Load	
Market		Skids	Spots
Carrier #		Weight	Status
Inbound		Price	
Pick Up		Base Price	Fuel
Shipper		Extra Stops	Detention
City From		Total Price	Currency
P/U due date	P/U number	Commodity	
P/U Appt. Time	P/U Appt. #		
Ready At	P/U Till	Border	
Shipper's hours		Customs Broker	
Shipper's notes		Paps-Pars	
Picked Up On	Picked Up At	Transaction	
Delivery		Border Status	
Receiver		Invoice	
City To		Invoice #	
Del. due date	Del. number	Invoiced On	
Del. Appt. Time	Del. Appt. #		
Preferred Time	Del. Till		
Receiver's hours			
Receiver's notes			
Delivered On	Delivered At		
Comments			

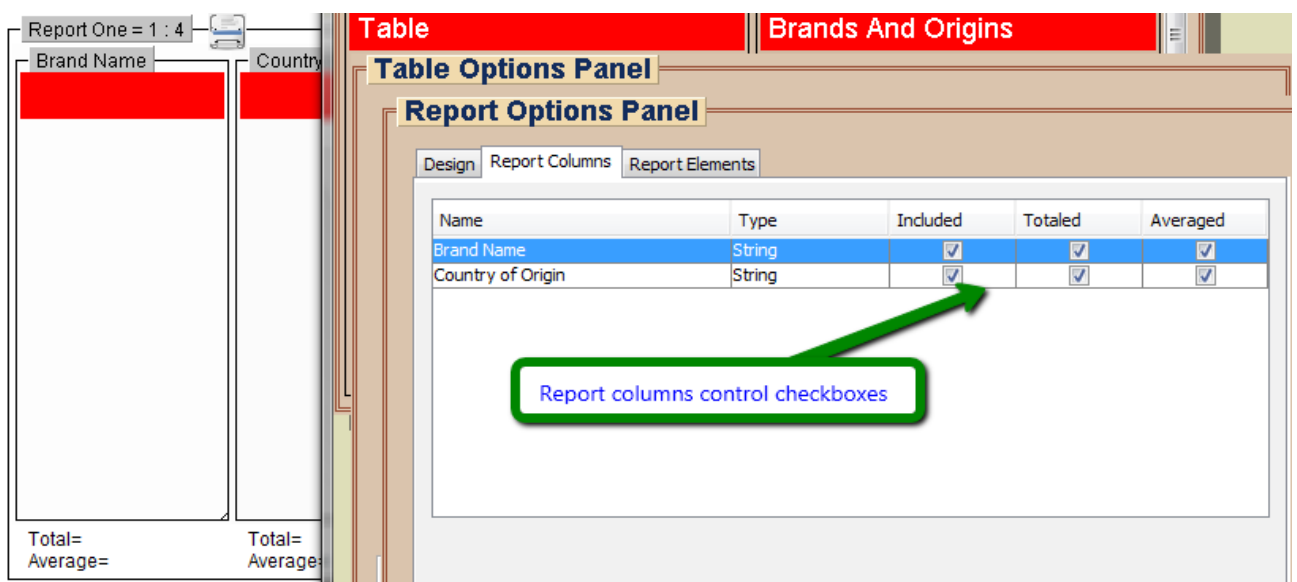
To print a table or a custom table view we have to prepare a report. Click on the "Reports" tab of the Construction Tool:

Auxiliary Tables		Layers = 4 : 5	
Brands		Type	Name
Brands And Origins = 1 : 4		Table	Brands And Origins
Brand Name		Table Options Panel	
HP		Design Data Fields UI Elements Special Ops Reports 1. Name a report 2. Create a report 3. Prepare or edit the report 4. Save the layer with associated reports	
Tim Horton		New Report Edit Report Delete Report	
BMW		Delete Cancel Save	
Chanel			

Lets name a report "Report One", create it and click on the "Edit" button:

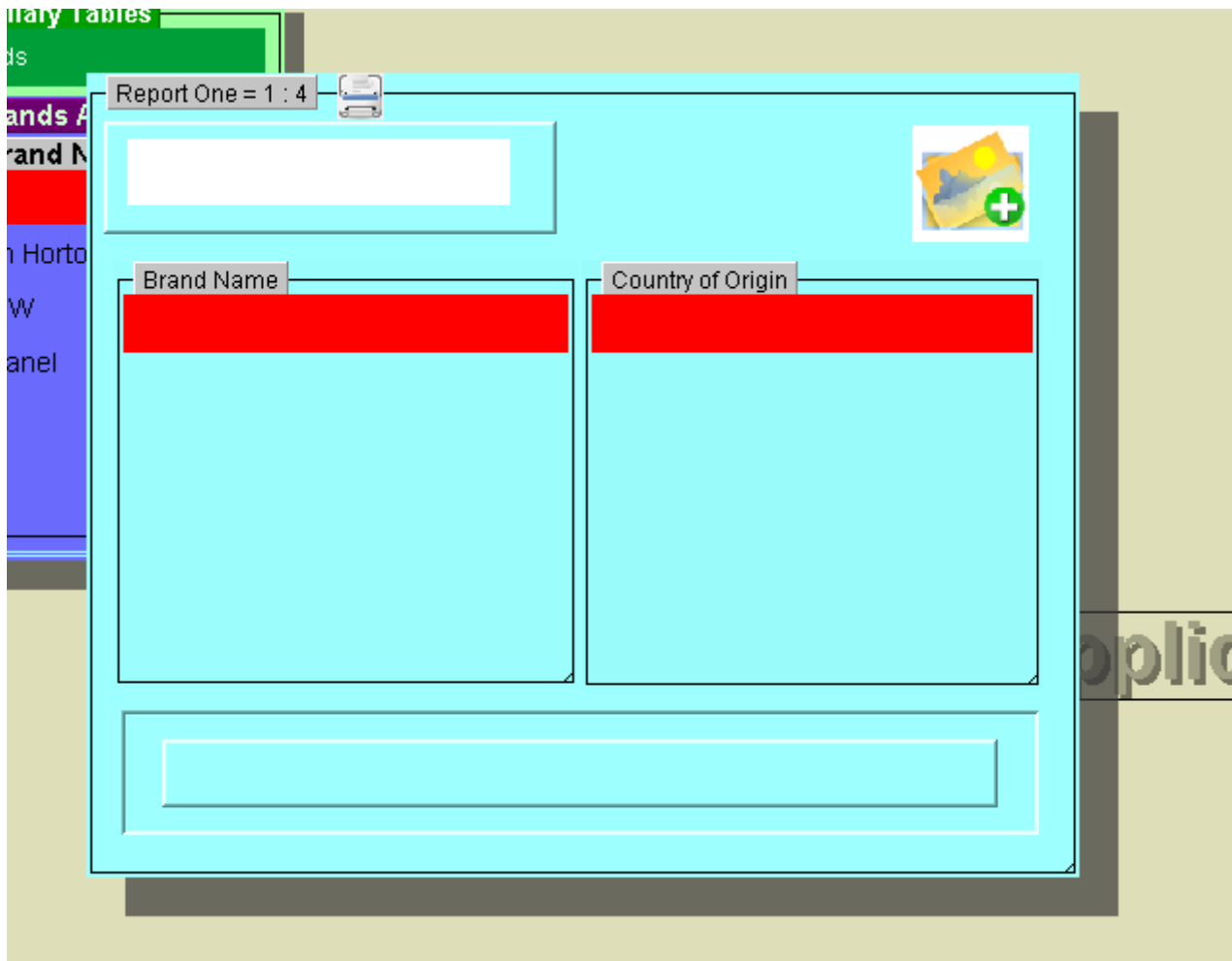


A Report is just another layer with a printer icon and its personal Options Panel at the Construction Tool. Geometry and design are controlled the usual way of tuning a layer. It might be wise to leave the background white if a report is meant to be printed on the white paper. Now we have to decide which columns from the "Brand And Origins" table will be included in our report. Click on "Report Columns" tab and try different combinations of the checkboxes:

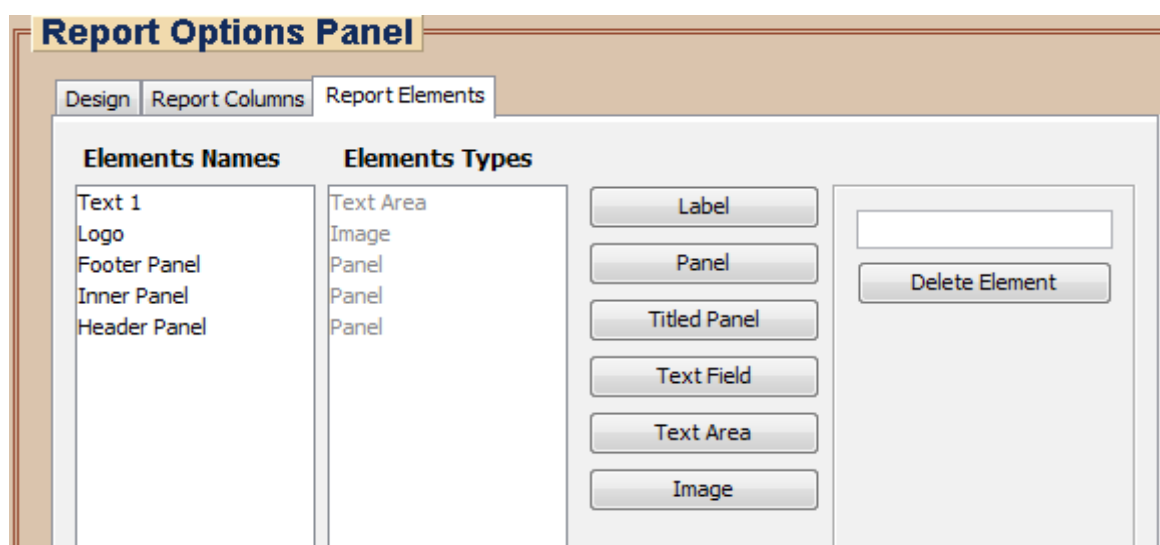


In our case "Total" and "Average" fields (a report will calculate them automatically) make no sense, because our columns do not contain numbers. Let's uncheck them and see what "Report Elements" tab holds and how it could help us to build the report. Just remember when placing white

elements on the white "paper" we may not be able to see them, lets color the background just a bit:

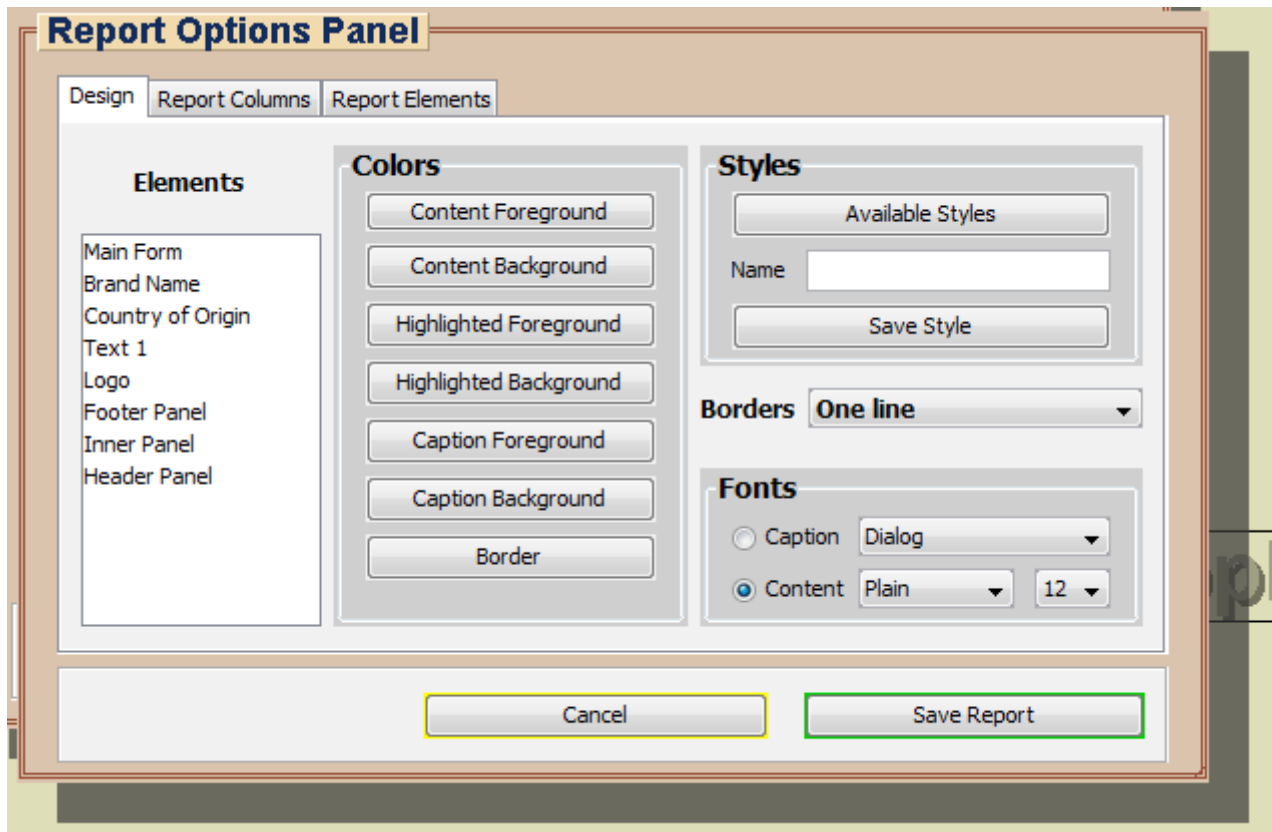


That is a report example with 3 panels, one text area and one image added. Panels are meant for beautification, text area will bear a title and image will bear a logo which could be set just before printing the report. The "Report Elements" panel will look like this:

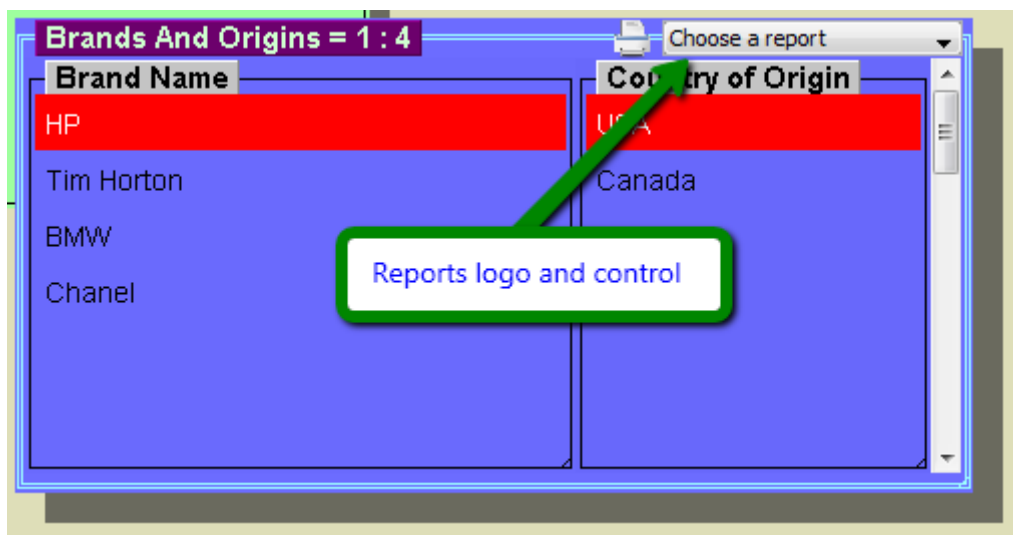


Each element was named, placed, left-dragged into position, right-dragged to its size, the Main

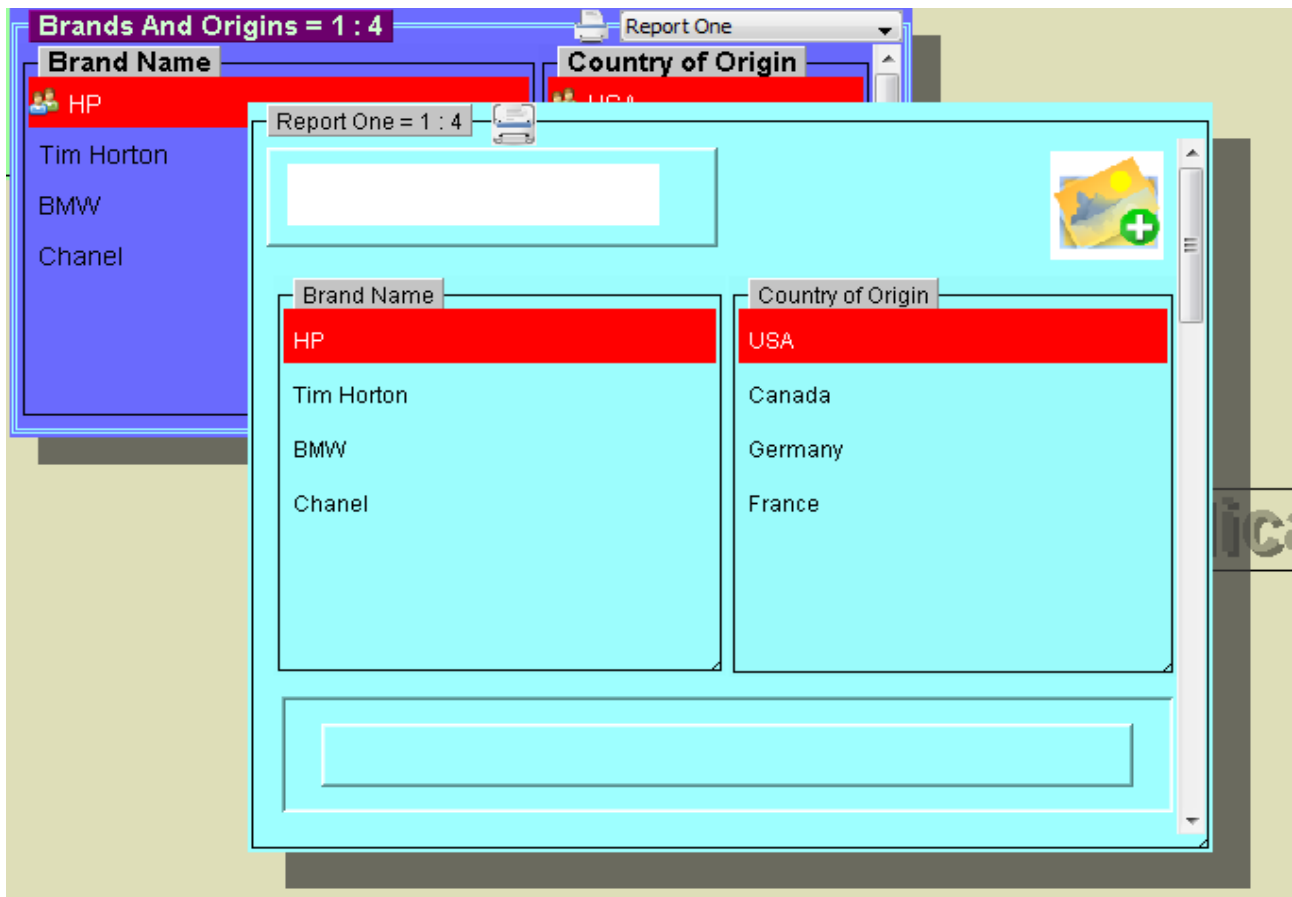
Window was resized to accommodate them all(right-resized to preserve Field Windows dimensions) and those Field Windows where placed where they belong to create the particular printable form. Every element now is available for individual color/fonts tuning in the Report's "Design" tab:



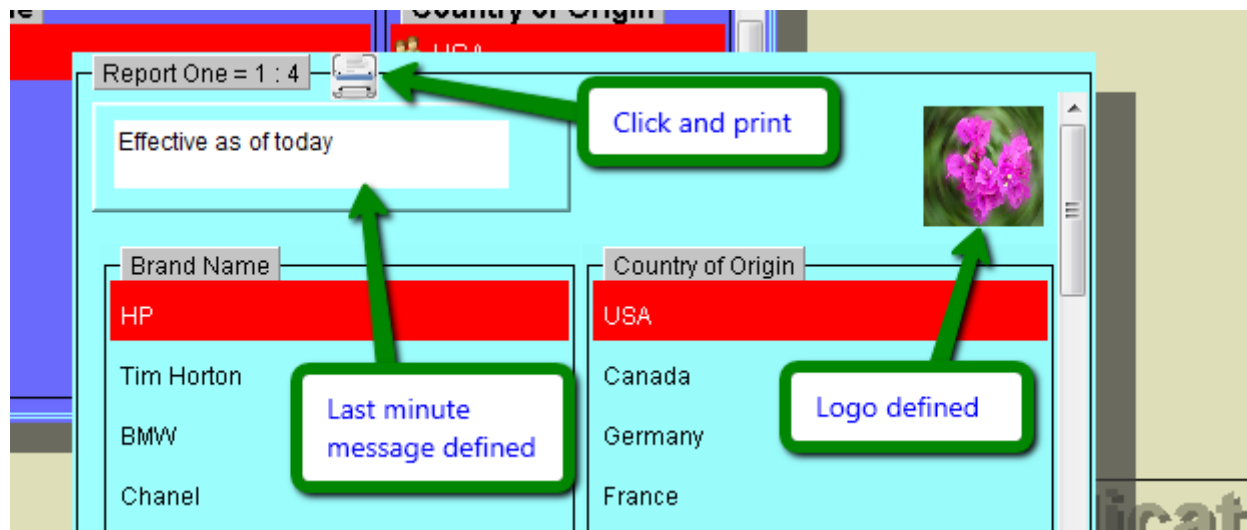
Lets just save the report and then save the layer to check how it works. Easy – the layer now bears the reports logo and the report control:



We have created only one report, lets see how it looks – choose the "Report One" from the drop down reports list:



The report is not quite ready for printing yet – it has neither the image nor the message. Click on the text area and type a message, click on the image placeholder and choose an image available on your computer:

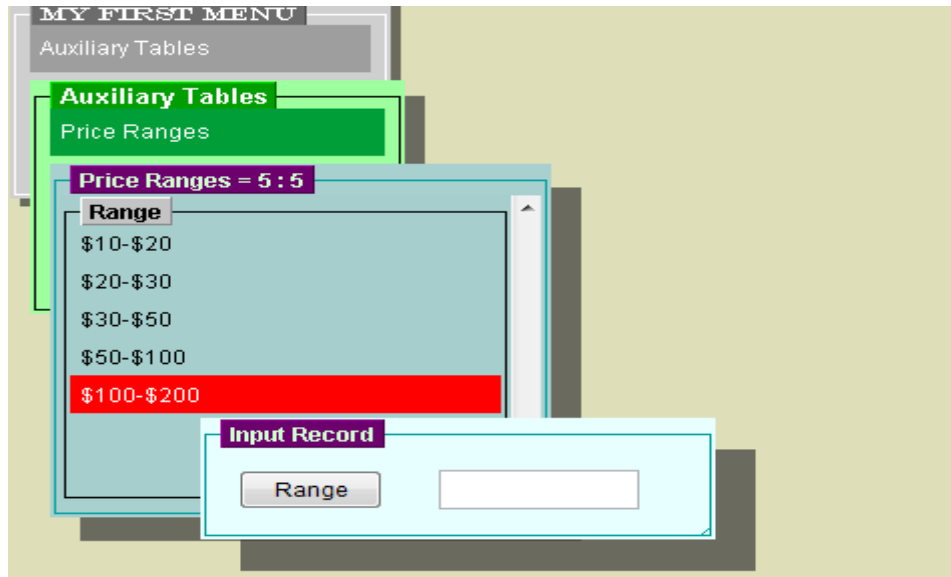


Click on the printer logo and get you report printed.

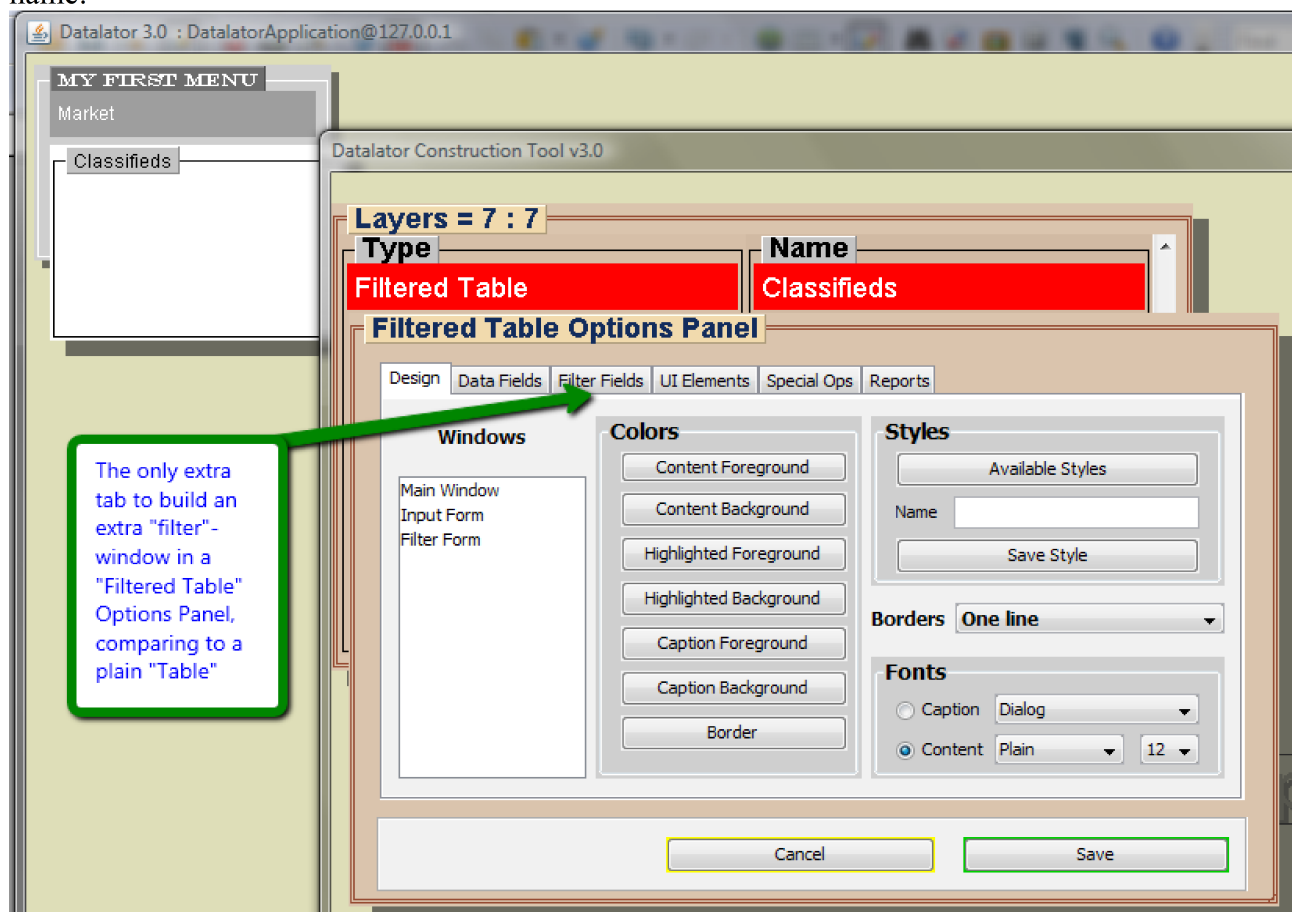
Those reports are meant to help in programming of "screen only" summaries and/or hard copy documents, even emailing them. But it is very possible you'll be able to design a nice looking document (say an invoice) with your logo, "Gross" field and multiple lines of orders on a few pages without programming at all. Happy printing!

9. Building A Filtered Table.

By now you should be able to design the last simple table of "Price Ranges" in no time. We need it to support the main goal – the Filtered Table "Classifieds". The "Price Ranges" might look and be populated like this:



Navigate to the main menu now, "enter" the line "Market" and start the Construction Tool by right-clicking on the "dummy" layer caption. Choose "Filtered Table" as a type and "Classifieds" as a name:



Lets build this table with Data Fields shown in the "Data Fields" tab, as we did building a Table:

Filtered Table Options Panel			
Design Data Fields Filter Fields UI Elements Special Ops Reports			
Name	Type	Visible	Link
Goods Category	String	☒	
Brand	String	☒	
Item name	String	☒	
Price	Currency	☒	
Date Posted	Date	☒	
Item Picture	Image	☒	

Lets place the Data Fields into the Input Form as shown (we intend to "hide" bottom fields):

Classifieds

Goods Category Brand Item name Price Date Posted Item Picture

Input Record

Item name

Price

Goods Category Date Posted

Brand

Item Picture 

Lets resize the Input Form, hiding "Goods Category", "Brand" and "Date Posted" to deny direct user access when making an add – our Application will populate those fields automatically:

Classifieds

Goods Category Brand Item name Price Date Posted Item Picture

Input Record

Item name

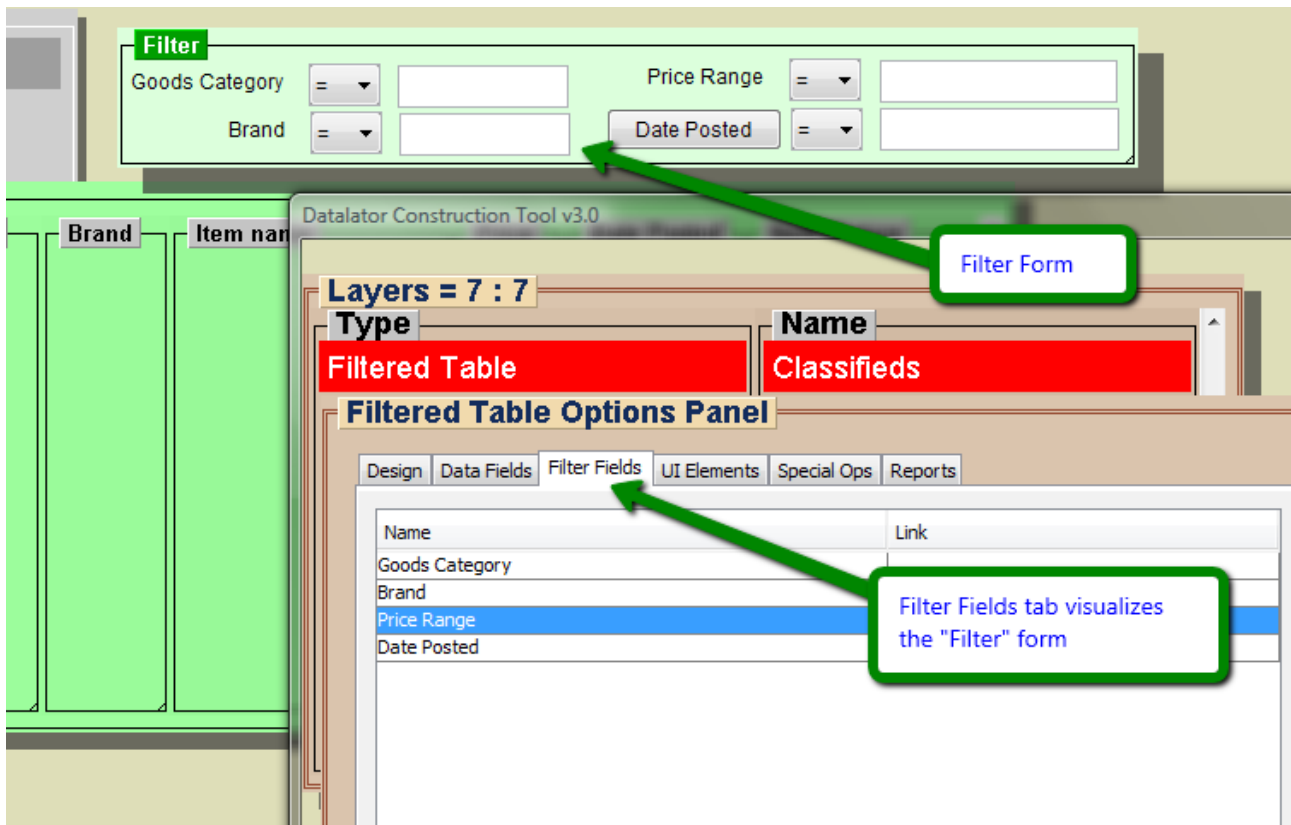
Price

Goods Category Date Posted

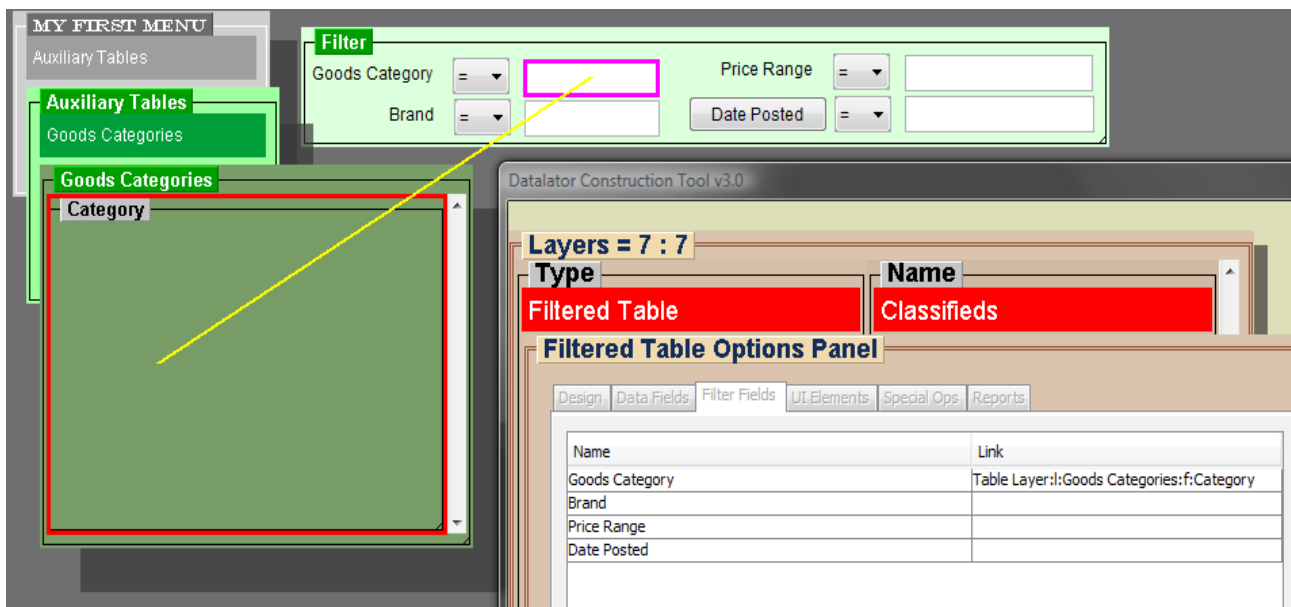
Brand

Item Picture 

Lets click on the "Filter Fields" tab in the Options Panel. That action will produce a "Filter" window, which will act as a "search" tool in application. We usually keep it on the top of the Main Table Window, as shown. Lets define the four "Filter Fields" and place them on the Filter Form:



Now we have to define the links – click on "Goods Category" row in the "Filter Fields" tab, click on "Set The Link" button and navigate thru the "new" Main Menu to the "Goods Categories" table; click on it – it will set the link:



Click on the "Stop/Cancel" button and repeat with three remaining Filter Fields.

The links should be defined like that:

Name	Link
Goods Category	Menu Layer::Auxiliary Tables
Brand	Table Layer::Brands And Origins:f:Brand Name
Price Range	Table Layer::Price Ranges:f:Range
Date Posted	(Calendar)

Go to "Special Ops" tab, export your newly created layer(safety first!) and click on the Save button. Your "Classifieds" are ready to be used now. Lets explore the possibilities. When the Filtered Table is activated, only the Filter window is visible:

MY FIRST MENU

Market

Filter

Goods Category = [dropdown] [input box]

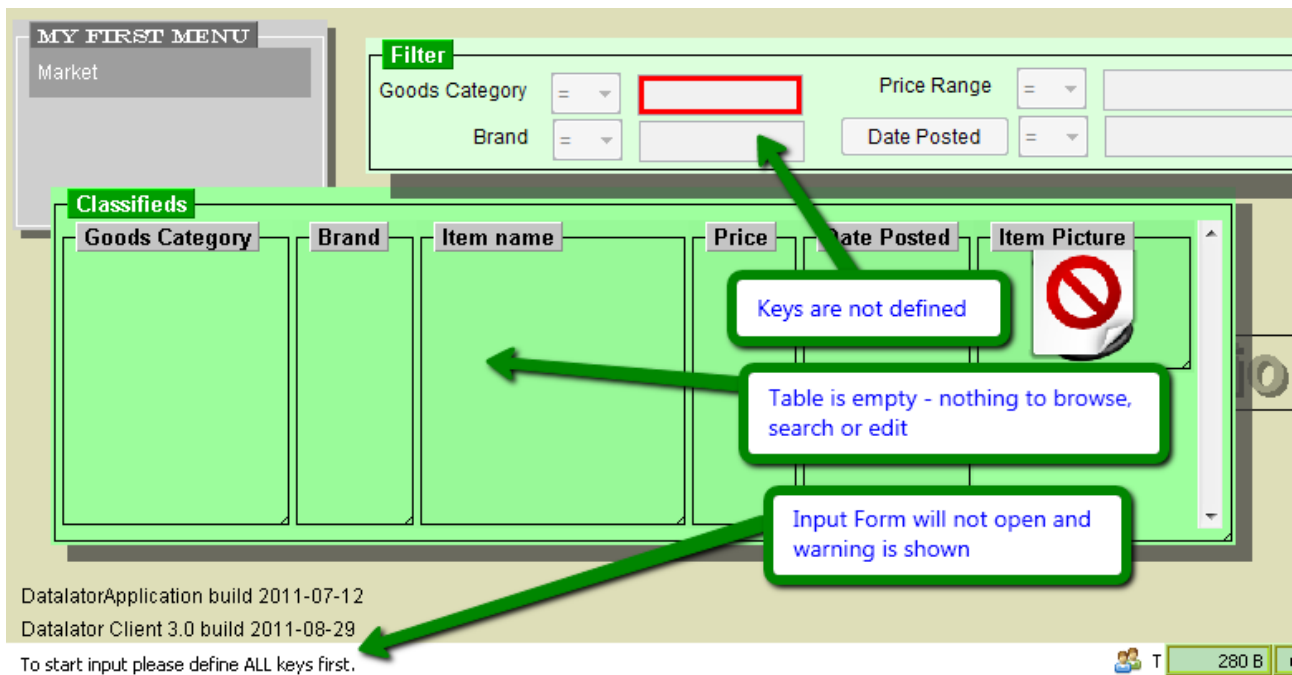
Brand = [dropdown] [input box]

Price Range = [dropdown] [input box]

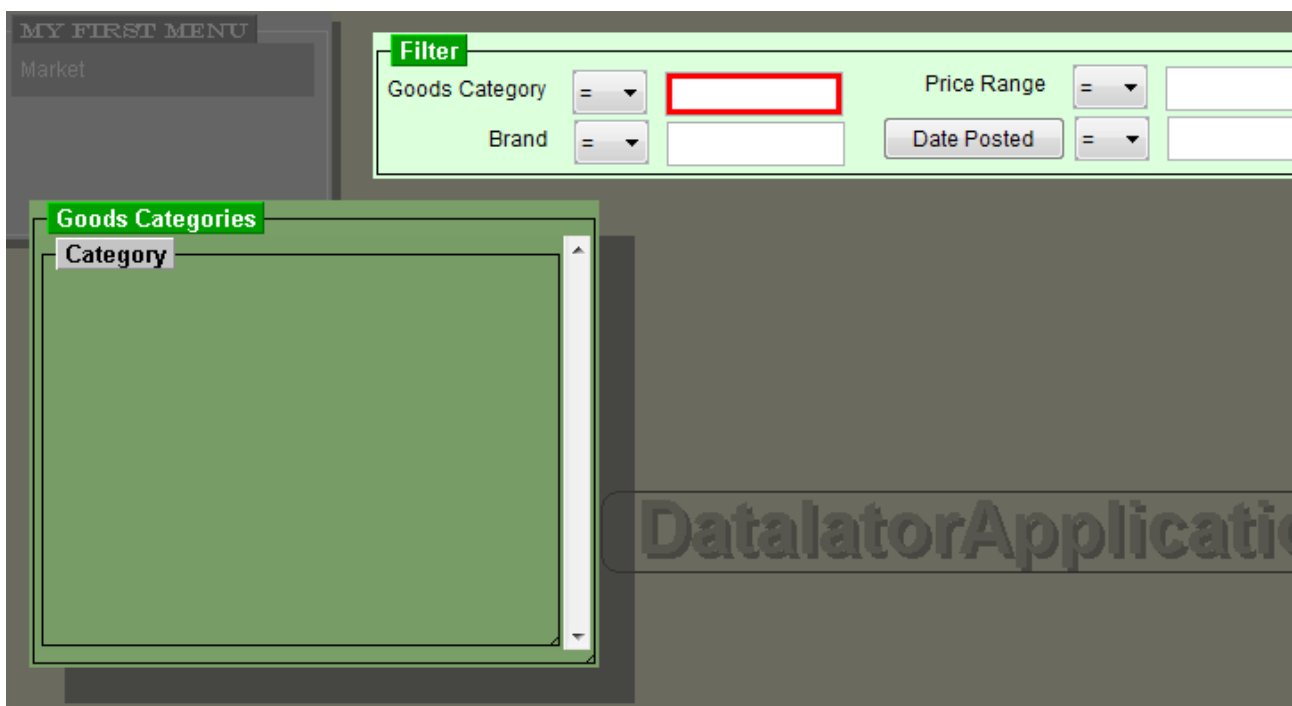
Date Posted = [dropdown] [input box]

You may want to click on the activator button (Date Posted – the last field) to check on the table content. It is empty – we and/or users haven't entered no information there yet.

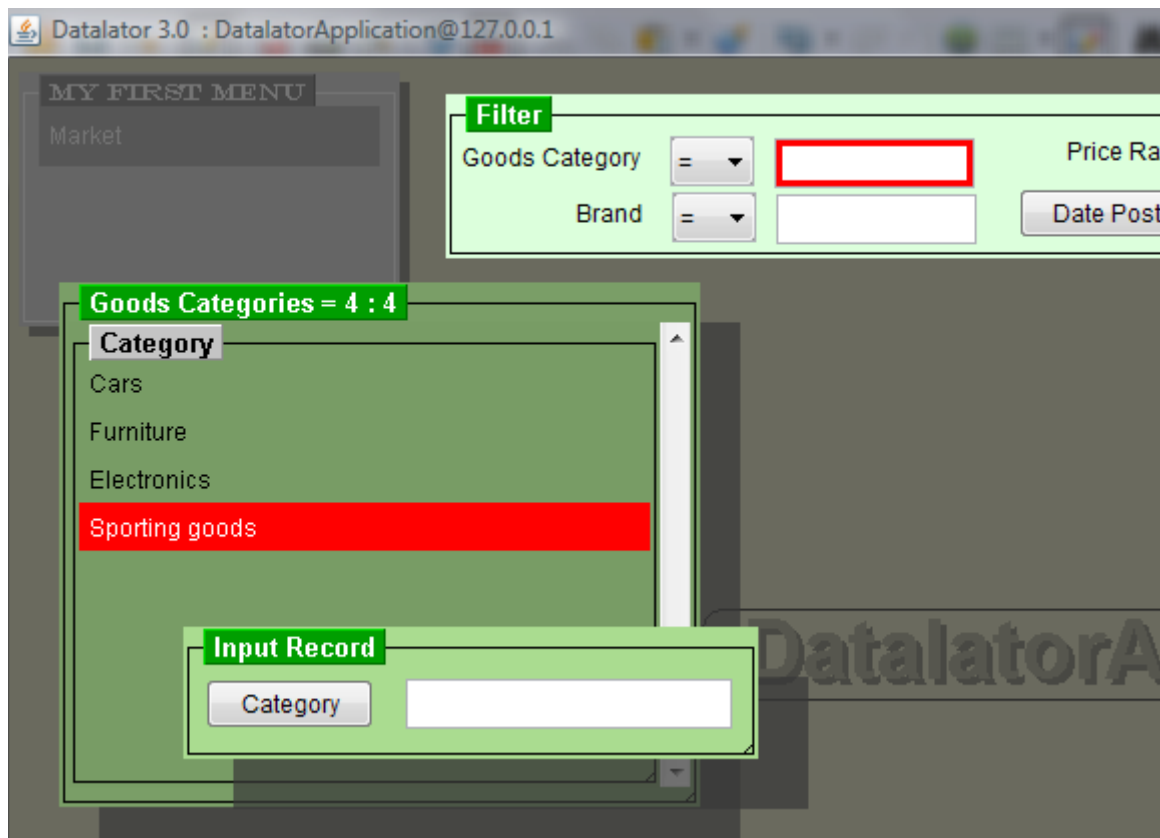
Note: When one or more keys(Filter Fields) are not given a value - input into the Filtered Table is denied. Browsing, search and editing of existent records are allowed.



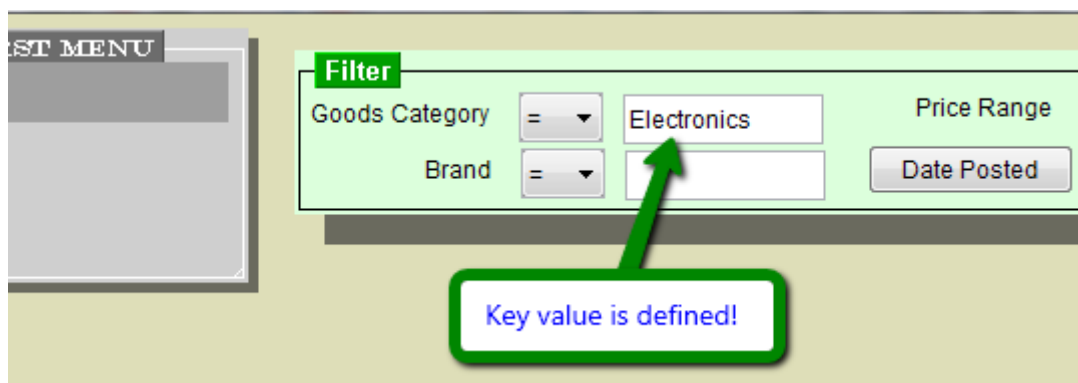
Lets define those keys; they are only accessible when Main window of the table is "escaped". Hit the "Escape" button or right-click on the application background (the table "Classifieds" will hide) and click on the "Goods Category" text field of the Filter window:



The empty table will pop up, because we did not defined any records here yet. Open "Categories" Input Form (F5) and enter a few records into the table:



Close the Input Form(Esc) and double-click on "Electronics" record – thus we defined a key!:



Do the same with "Brand" and "Price Range":



Click on the "Data Posted" filter text field and the built-in "Calendar" will pop up:

Filter

Goods Category = Price Range =

Brand = Date Posted =

Date Chooser

< September > << 2011 >>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Click on the appropriate date from the Calendar :

Filter

Goods Category = Price Range =

Brand = Date Posted =

Time to click on the "activator" button "Date Posted", get the "Classifieds" table, open the Input Form and type in something like that:

MY FIRST MENU

Market

Filter

Goods Category = Price Range =

Brand = Date Posted =

Classifieds

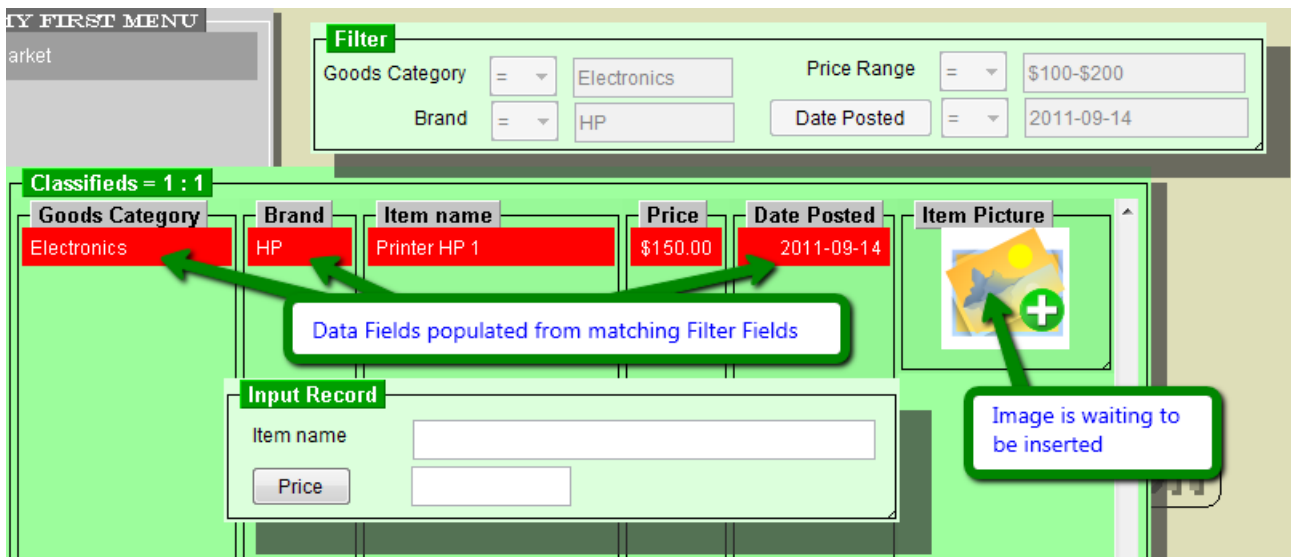
Goods Category	Brand	Item name	Price	Date Posted	Item Picture
					

Input Record

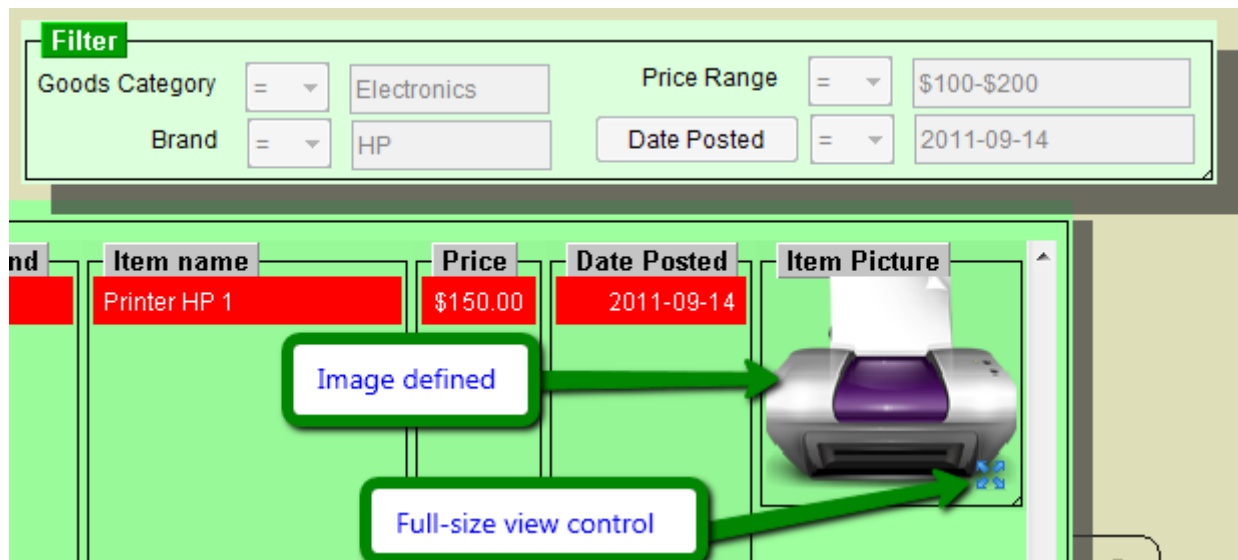
Item name

Price

Click on the "activator" button "Price" and insert the record like this:



To finalize setting up our "HP Printer 1" for "Classifieds", close the Input Form and click on the image placeholder. The "Open File" dialog will pop up; choose the image of the item and it is also inserted into the application Database:

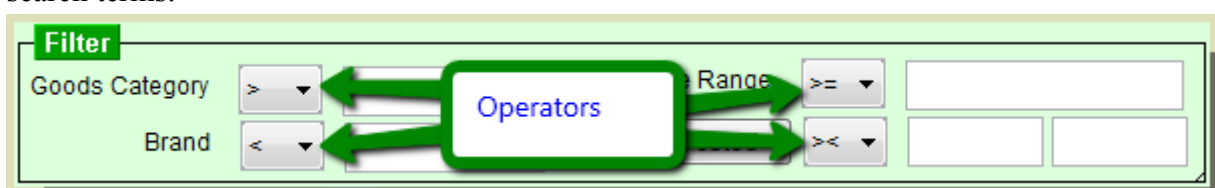


The development of the "Classifieds" Filtered Table is done. Lets check the possibilities of the Tables.

10. Filtered(and plain)Tables at Work.

Filtering Records

When entering a record you **must** set values of all keys of a Filter. When searching for a particular set of records you **may** set values of zero or more keys, using one of eight operators to define the search terms:



To better illustrate the power and simplicity of that Filter Window we'll show a few real-life examples. That 12-keys Filter (shown below) of "Orders" Filtered Table search request will produce no records at all:

The 'Filter' window for 'Orders' contains the following fields and values:

- Driver: = Jason
- Truck: =
- Trailer: =
- Dispatcher: =
- P/U due date: >< 2011-08-01 2011-08-31
- Inbound: = Inbound
- Customer: = Rogue Specialty Transport
- Shipper: =
- Receiver: =
- Customs Broker: =
- Invoice status: =
- Market: =

and this another one produces a set of 5 records:

The 'Filter' window for 'Trips' contains the following fields and values:

- Dispatcher: = Cassandra
- Market: =
- Driver: =
- Truck: =
- Trailer: = 1109
- Date: =

Trips = 1 : 5

Trip	Driver	Truck	Date	Price Out	Price In	Sto
VZ182	Vlad2	108	2011-05-08	\$500.00	\$1,830.00	
VZ181	Vlad2	108	2011-05-04	\$450.00	\$1,415.00	
VZ180	Vlad2	108	2011-05-02	\$660.00	\$1,210.00	
VZ179	Vlad2	108	2011-04-28	\$450.00	\$1,500.00	
VZ178	Vlad2	108	2011-04-26	\$785.00	\$1,425.00	

or another one – similar to our "Market" application – with search result of 2 records:

The 'Filter' window for 'Market' contains the following fields and values:

- Category: = Cars
- Type: =
- Kind: =
- Brand Name: = Toyota
- Price Range: =
- Posting Date: < 2011-09-15

Market = 1 : 2

Category	Type	Kind	Brand Name	Item Name
Cars	Pickups	Small pickups	Toyota	A smaller pickup or a car
Cars	Pickups	Full Size	Toyota	A full size truck

Sorting the Records

When Filter Window request produces too many records you may sort your big set by double clicking on any Field Window caption, like that:

Customers = 962 : 1016

Name	City	CSR	Phone
Agility Logistics	Mississauga, ON	Sandra	866-354
Ly	Mississauga, ON	Lawnie	905-858
AD	Mississauga, ON		-564
ShipNorthAmerica Transportation	Mississauga, ON		-593
Manitoulin	Mississauga	Aby	905-283
Teamwork	Mississauga	Andrew	905-795

That technique works with plain and Filtered Tables. Here is another sort example:

Shippers = 1 : 4608

Name	City	Hours	Phone
2 Source Mnf	Mississauga	H	905-361
3 River Ice Cream	Gibsonia, PA		724-759
31 Inc.	Newcomerstown, OH	8:00AM-NOON	800-438
31 Inc	Newcomerstown, OH	6AM - 4PM	740-498
3D Storage Systems	NewMarket, ON		05-830
511 Foods	Oakville, ON	8	05-829
5S Components	East Mckeesport, PA		412 967
711 Pool Products	Mississauga	8AM - ..	905-625

sorted 4608 records in 17138 msec

When you want to restore the original order - double click on that Field Window caption again.

Note: sorts are performed on your local computer and do not affect other users working with that table. To make your sort permanent and visible to others click on the "Save the Sorted Order" icon in the caption of the Main Table window.

Searching a Record

If a Filter and/or a Sort cannot help you to find a particular record, you may want to search the record by typing in a pattern(substring) into a text field of an Input Form. Here is how it works. Load a table of interest, open the Input Form(right-click on the table or use F5 button) and switch the input mode to one of search modes (use F5 again or click on the Input Form caption):

The screenshot shows a large table of customer records with columns: Name, City, CSR, and Phone. The table caption is "Customers = 1 : 1014". A dialog box titled "Find Record Containing..." is open, showing a list of fields to search by: Name, City, CSR, Phone, Fax, Email, Web, Report by, and Rating. Annotations include:

- "Big table to search" pointing to the main table.
- "Search mode" pointing to the dialog box.
- "Click on the Input Form caption to switch mode" pointing to the dialog box caption.
- "More fields available to search by then to sort by" pointing to the list of fields.
- "Type in one of text field to initiate the search" pointing to the text input fields in the dialog.
- "No 'activator' button means you cannot change the content in search mode" pointing to the dialog box.

Any symbol typed in any available text field will show the matching record as active(highlighted). No match – "active: pointer stays in place. Every time you type, the search starts from the top of the table. To check for your pattern further down the table (next match) use F3 button, see examples:

The screenshot shows the "Find Record Containing..." dialog box with the "Phone" field selected and the value "454" entered. A callout box says "Looking for a phone number containing '454'". The background table shows records with phone numbers, and the record with "454" in the phone field is highlighted.

The first match has been found(record 74 of total 1014 shown in the Main Window caption):

Customers = 74 : 1014

Name	City	CSR	Phone
R & R Transportation			6-410-9940
RoadLINX International			5-760-1141
Tempo Logistik			4-639-5123
XTC Logistics			9-749-1050
C & D Logistics			4-881-4440
Woodcock Transportat			3-6194
Transco Max			1-7522
Trans 58			5-7671
The Master's Group Ma			9-124-1600
Labyrinth Services Inc			5-629-454

Find Record Containing...

Name

City

CSR

Phone

Fax

Email

Web

Report by

Rating

Searching for a phone # containing "454" digits

Here is the match, to get a next match use F3 button

Another match found by pressing F3 button:

Customers = 725 : 1014

Name	City	CSR	Phone
Maple Gate Freight Sys			5-804-8213
Marbro Group Inc.			0-739-0939
Market Transport Servi			
Marketrans Transporta			0-599-2260
Martin Roy Transport (I			9-762-2620
Mc Freight Systems			0-700-9303
McArtur Express Inc.			
Megal Corp			
Megaliner Freight Syst			5-789-5912
Melburn Transpor/Mus			5-275-4545

Find Record Containing...

Name

City

CSR

Phone

Fax

Email

Web

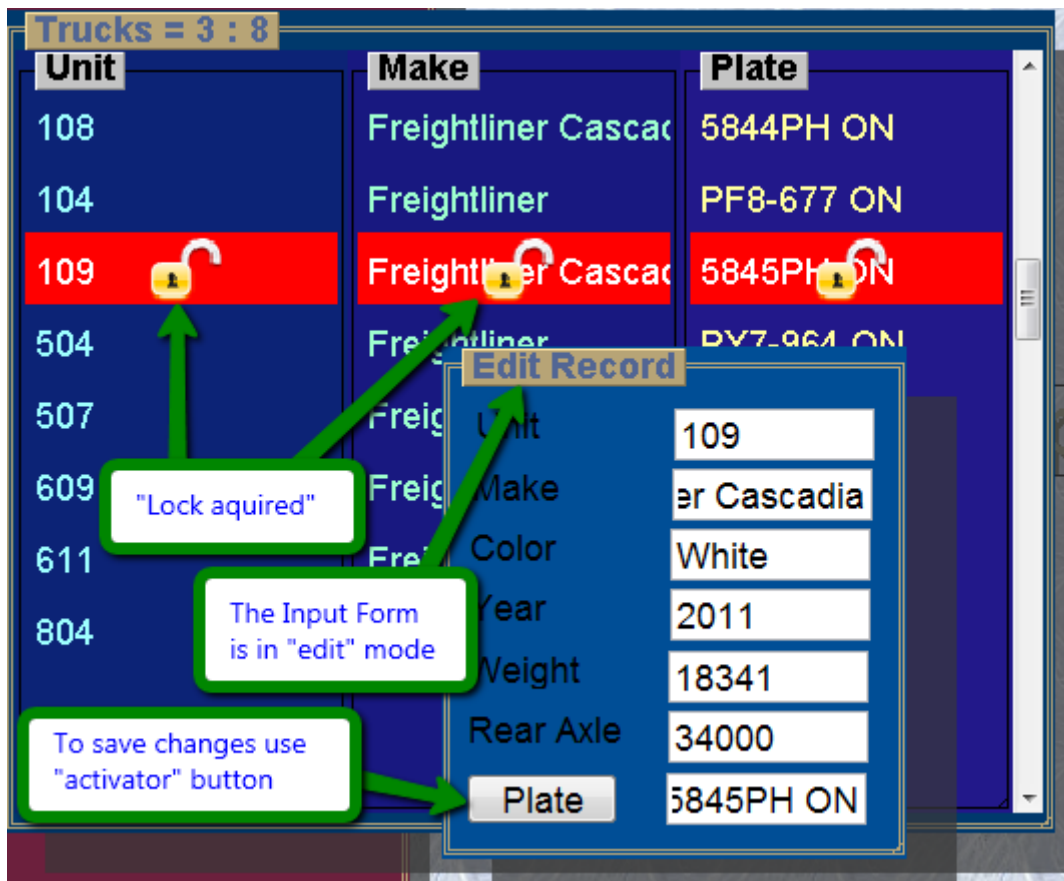
Another match found by using F3 button

Note: there are two search modes – by a pattern "**containing in**" and "**starting with**"; no wildcards supported – pattern to search is taken "as is".

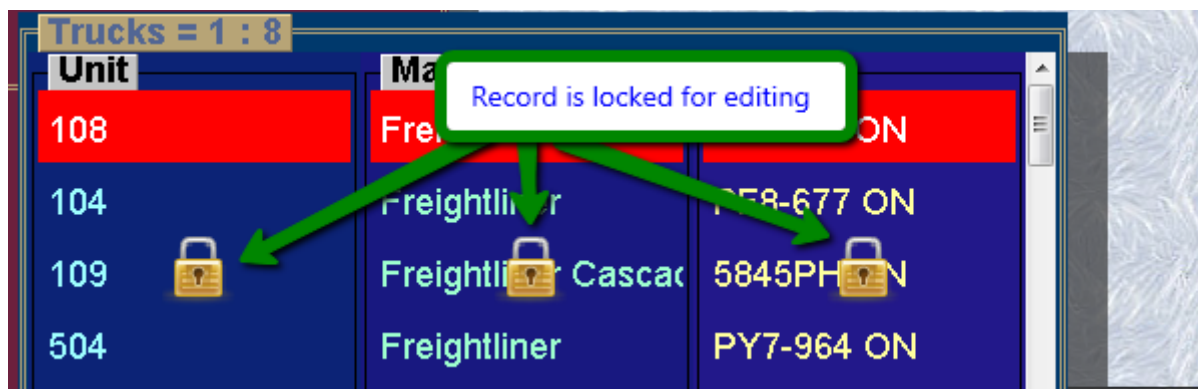
Note: search conducted only by the pattern from the textfield you are typing in. Other textfields values are not considered.

Editing a Record

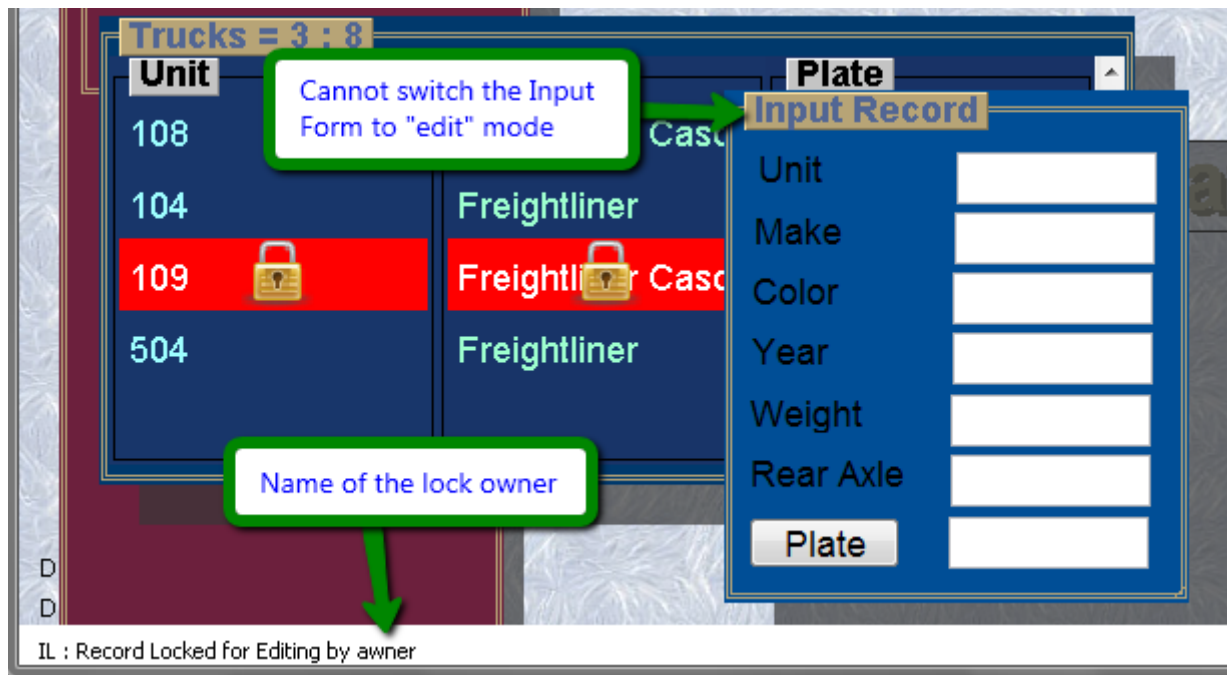
To change a field value of a particular record of a table, set the "active" pointer to that record by clicking on it, by searching for it or by scrolling to it. Open the Input Form (use F5 or right-click on the table) if it was closed and set the Input Form mode into "edit" by clicking on the Input Form caption or pressing F5 again. The Input Form will cycle its modes thru "input", "browse", search by containing in", "search by starting with" and "edit". Before switching to "edit" mode the form will request the server to lock the record. If the request is successful, the record will be marked with the icon "lock acquired", like that:



Other users, dwelling in the same table, will instantly see the icon "record is locked for editing":



Any attempt to edit that record by another user (using the Input Form in "edit" mode) will cycle the Input Form over to "input" mode and will show the message:



Note: "locked for editing" record is still accessible for browsing and searches.

After acquiring the lock, make desirable changes and click on the "activator" button of the Input Form. The change will be made in the database, the Input Form will unlock the record and switch it's mode to "browse". All the users happen to have that table open will be notified about content changes instantly – the locks will be gone.

Note: to quite editing just close the Input Form.

Note: all changes made are permanent – there is no "undo" function implemented.

Changing Records Order

Sometimes you want to keep your special order of records in the Table. By default the record will "remember" the position where it was inserted; you may place it at the end or beginning or in the middle of the Table. To move the position of the active(highlighted) record use a Ctrl/Up or Ctrl/Down keys combinations. Next example shows the original order of the table:

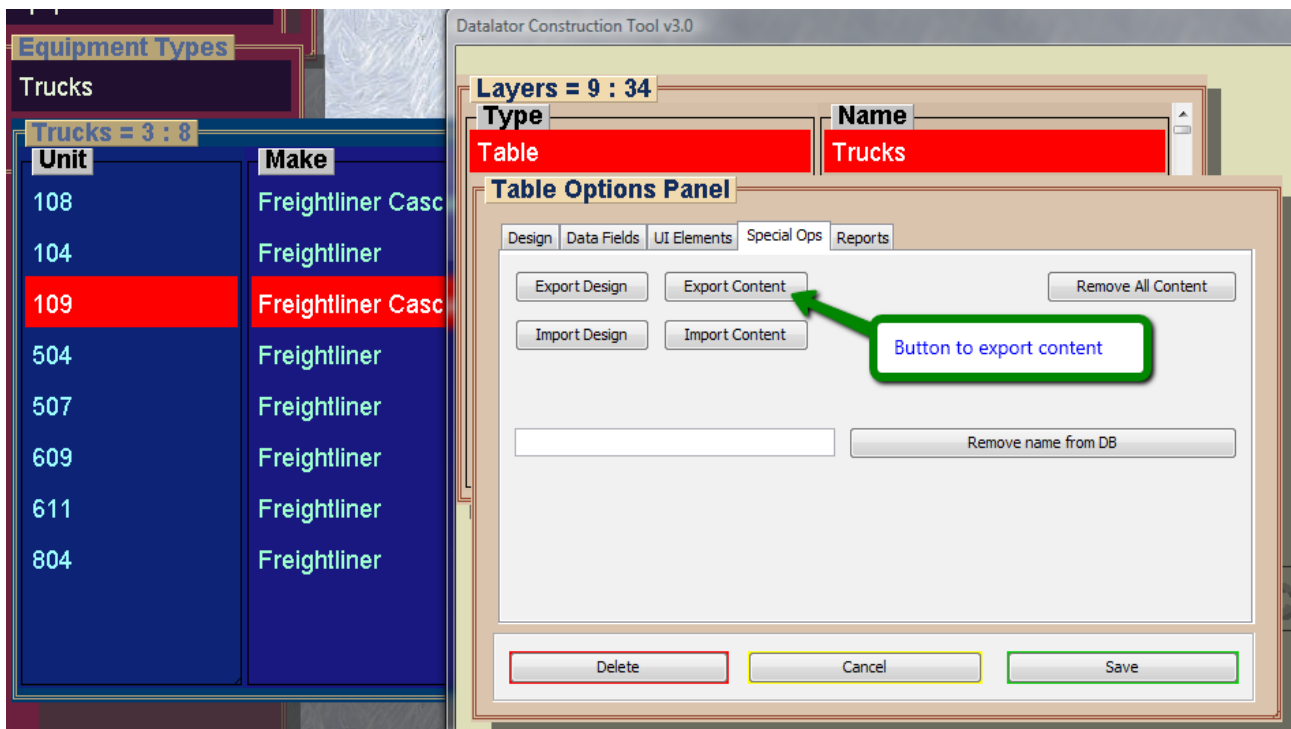
Trucks = 2 : 8		
Unit	Make	Plate
104	Freightliner	PF8-677 ON
108	Freightliner Cascadia	5844PH ON
109	Freightliner Cascadia	5845PH ON
504	Freightliner	PY7-964 ON

and the same table where the active record permanently moved up by Ctrl/Up:

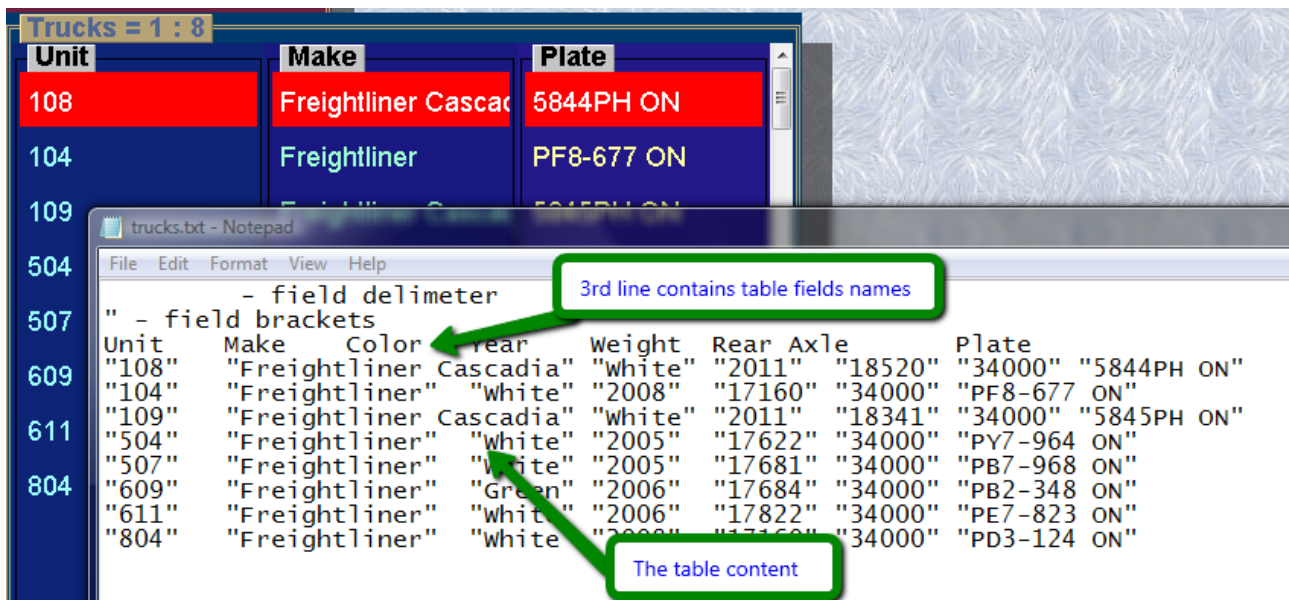
Trucks = 1 : 8		
Unit	Make	Plate
108	Freightliner Cascadia	5844PH ON
104	Freightliner	PF8-677 ON
109	Freightliner Cascadia	5845PH ON
504	Freightliner	PY7-964 ON

Exporting content of the table

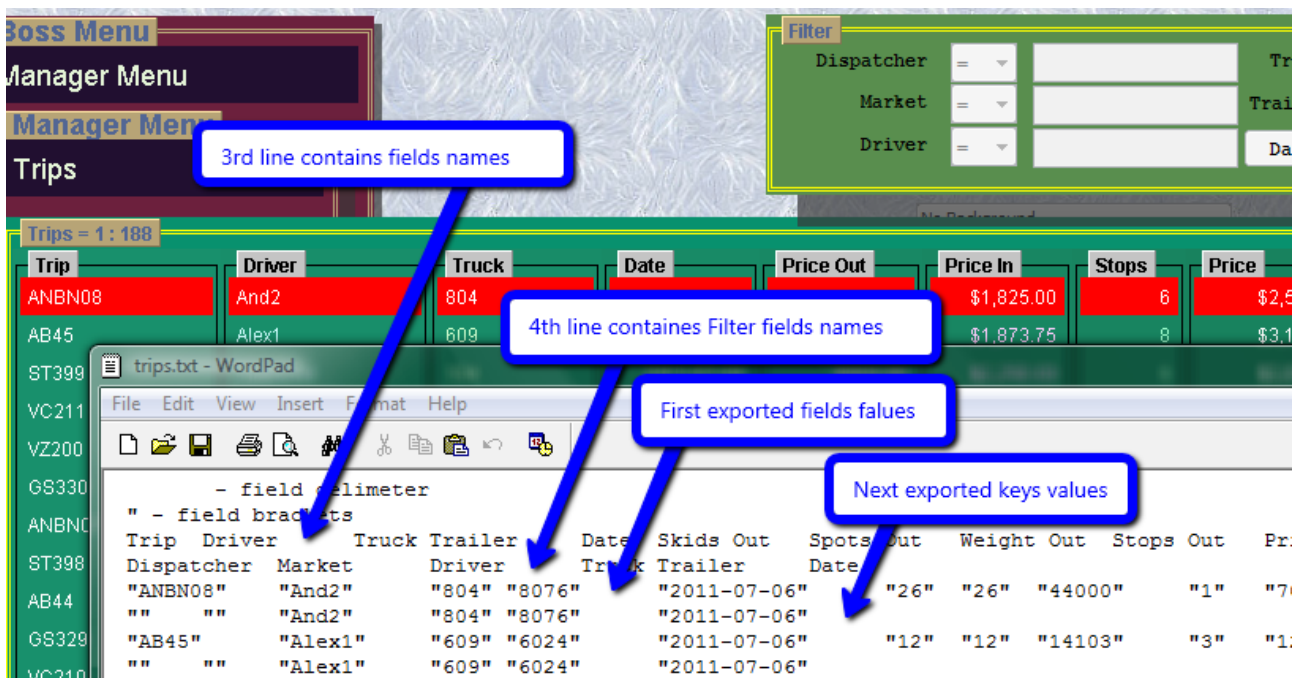
To download content of a particular Table to the local computer you have to have designer rights (given by administrator) or to be an administrator. If you are, right-click on the desirable table caption and start the Construction Tool, open the "Special Ops" tab, click on the "Export Content" button. A "Save File" dialog will pop up and you may save the content as a text file on your computer. Lets export the content of the "Trucks" table:



Format of the resulting text file is CSV format, with fields values "quoted" and separated by a "tab" character (Horizontal Tab with a decimal code 9 to be precise) and records kept in separate lines. Two first lines are artifacts and skipped when importing content back into the table. The third line contains names of the table fields, next lines contain the table row values:



When exporting content of a Filtered Table, the Table fields names will be followed by the Filter fields names and each record content will be followed by a line with key fields content:



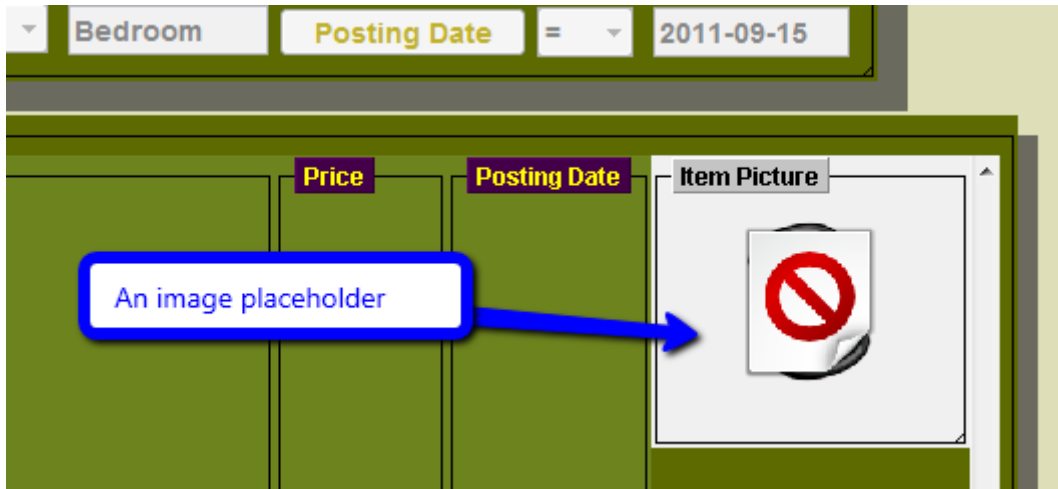
Knowing the format of exported tables may help if you want to export information from a Datalator application into another application type, such as a spreadsheet or a database.

Importing content of the table

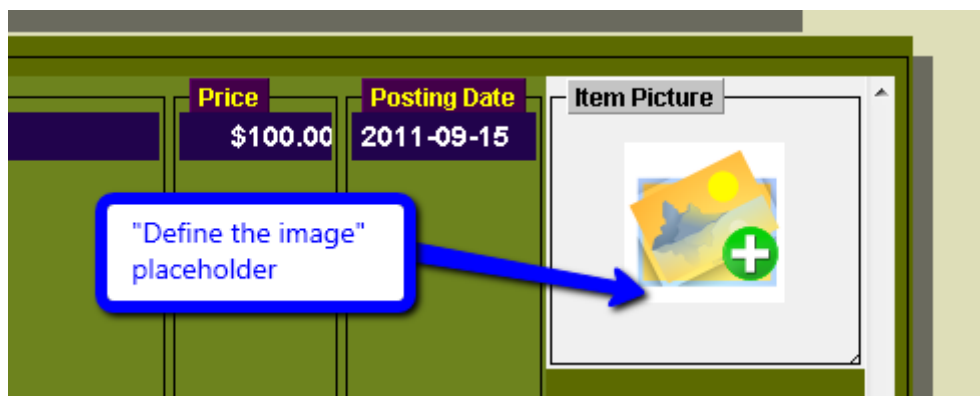
When there is need to populate your table(s) with information exported from another table(s) of another application, the Construction Tool is here to help. Open a table, right-click on the table caption – get the Construction Tool, click on the "Special Ops" tab and after on "Import Content" button. In the "Open File" dialog choose the file with the content of the table, exported before. If content was exported from another table, only the fields matching the names of the table fields will be imported.

Working with images

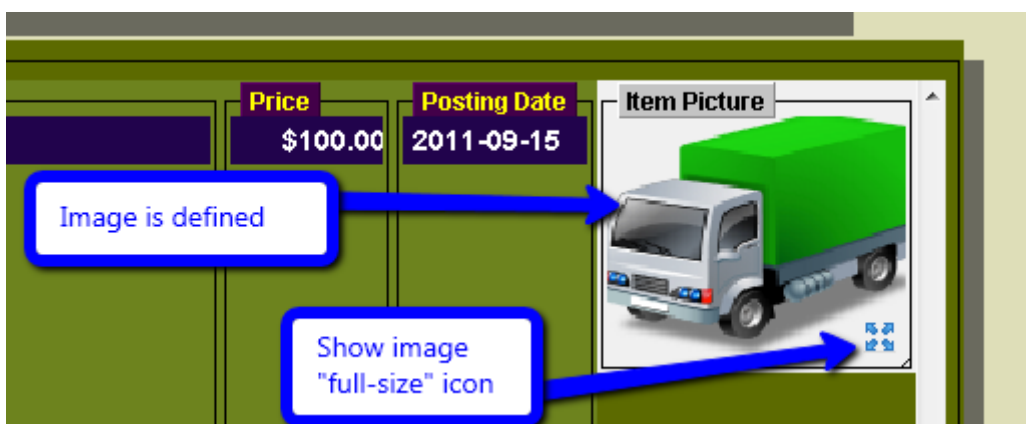
When constructing a table you may set a Data Field type as an "Image". That field content cannot be accessed thru the Input Form. An image must be associated with a Table record. When no records exist yet in the particular Table, the image is replaced by a "do not touch" placeholder:



When a record created, the placeholder changes to a more friendly type – "define the image":



Left-click on that placeholder and the "Open File" dialog will pop up. Choose an image to be uploaded to the database and associated with the record; watch the results:



All images are encoded and inserted into the Database – part of your content now.

You may redefine an image at any time by left-clicking on it – an "Open File" dialog helps you to choose the new image, which will replace the old one in the Database. All users will see changes instantly.

You may see you image in "full-size" mode. Click on the "full-size" icon inside the image and a new frame will pop up with the image shown in original size.

To download an image right-click on it and the "Save File" dialog will make it happen. When downloading only "png" format is available to save the image.

Note: images are kept in a database and encoded, thus consuming about 30% more room than on a storage device. By inserting a lot of huge images you will inflate your database pretty quickly.

Note: records are local – images are remote. That means an image is loaded from the server when active(highlighted) record of the table is redefined: click on another record and another image is loaded. If you scroll fast, only the last image is loaded, conserving the bandwidth. Still loading an image is not an instant operation, even in the LAN environment.

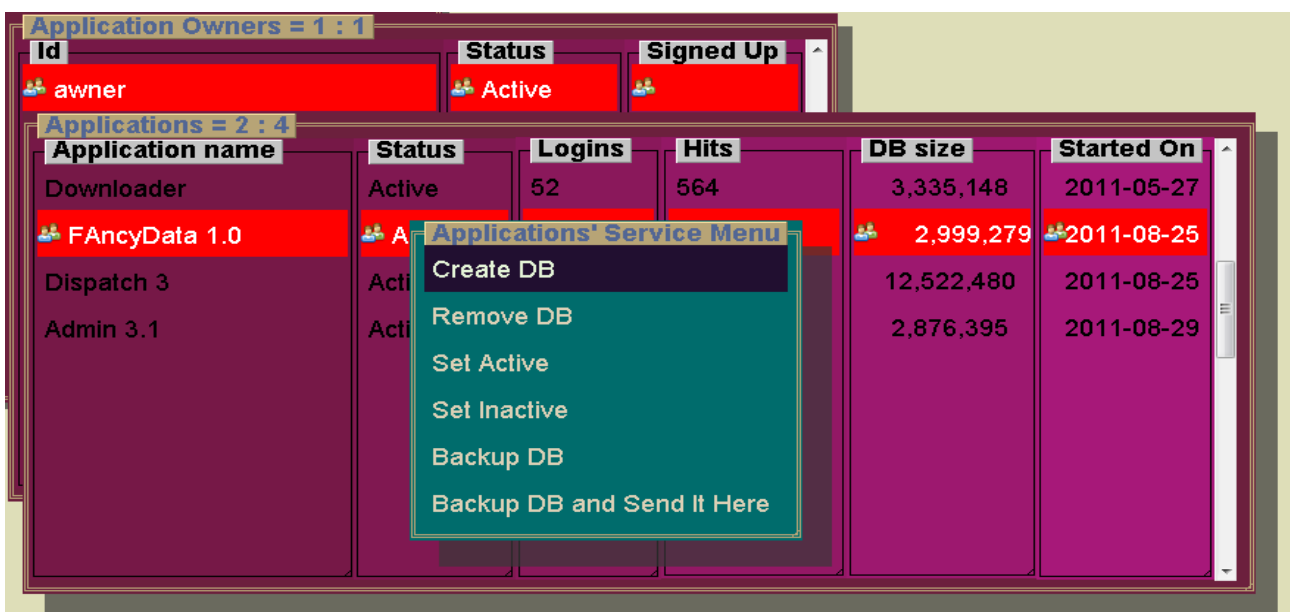
Note: there are no thumbnails when working with images; an image is loaded from the server with all the pixels available. It makes for a slower loading but instant "full-size" magnification.

11. Context Menus.

Every Table could be complemented by a Menu, meant to control some non-default activities, applicable to the whole Table or just to the "active" Table record. They are not all that useful without additional programming. Those "context" or "service" menus are just another way to "enter" a Menu from a Table – not with an "Enter" key or a mouse left double-click – but with the F2 key. Except for the activation method service Menus are just the regular ones. You may equip them with submenus or even with a different Tables if it makes sense in your application.

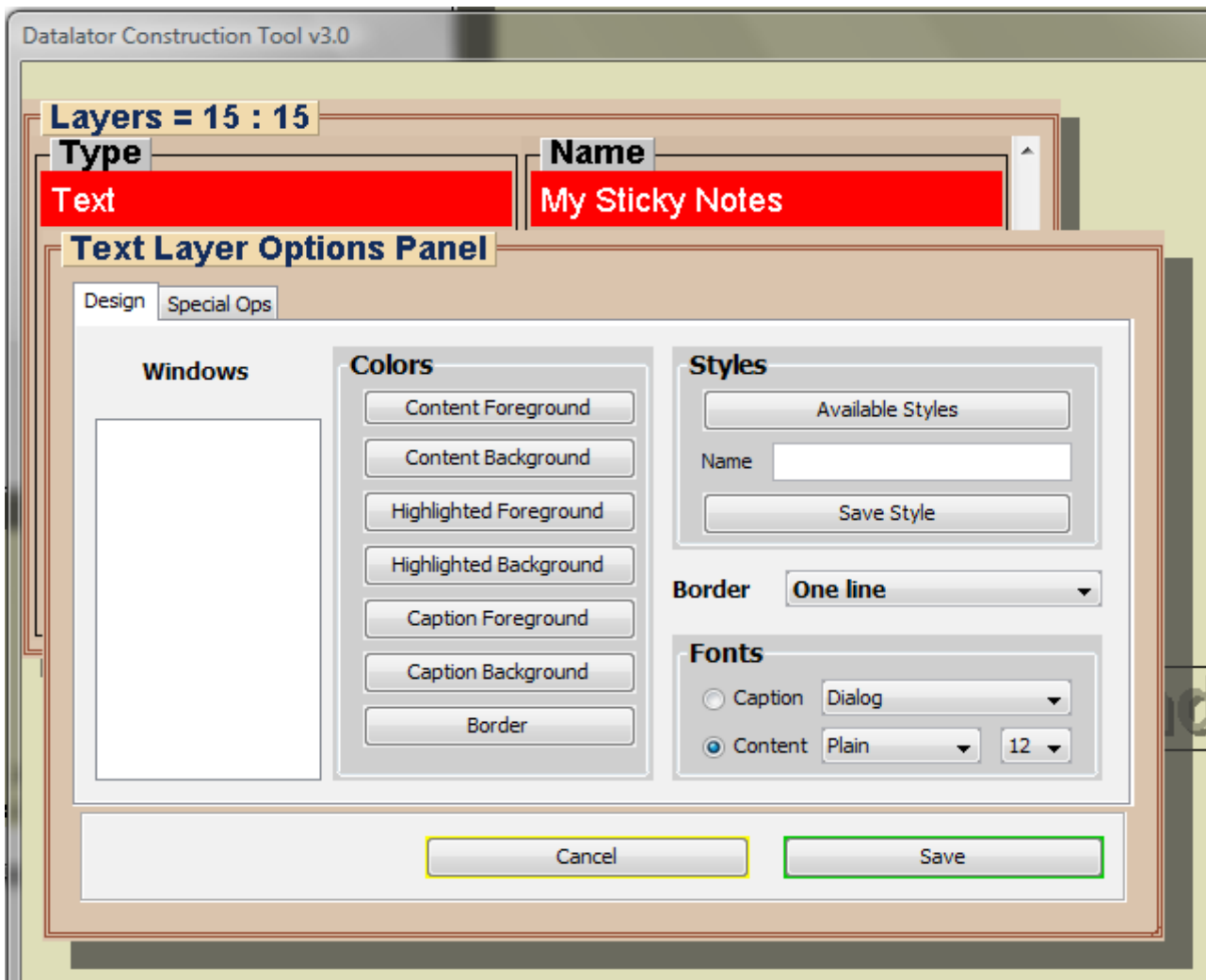
Anyways, to create a Service Menu use the F2 key when in a Table or Filtered Table. The gray "dummy" layer will pop up, ready to be customized the usual way. Type it as a "Menu" and name it as you like, customize and save; call it by pressing the F2 button. In the current Datalator version a Table has no indications of a Service Menu attached (likely coming in future Datalator releases).

Here are an example of a service menu:



12. Text Layer.

Text Layer is useful to make plain-text notes and associate them with Tables or Menu records. Every record in the database could be complemented by a personal "sticky" note. Every type of Table could have a different style of Text Layer. And every Menu line of every Menu could be equipped with a "custom text" as well. To create a Text Layer "enter" a Table or a Menu line, get the gray "dummy", start the Construction Tool, type the Dummy as "Text" and name it as you pleased. The Text customization panel is pretty straightforward:



Use the "Design" tab to beautify and the "Special Ops" to export/import design.

Examples of Text Layer applications are:

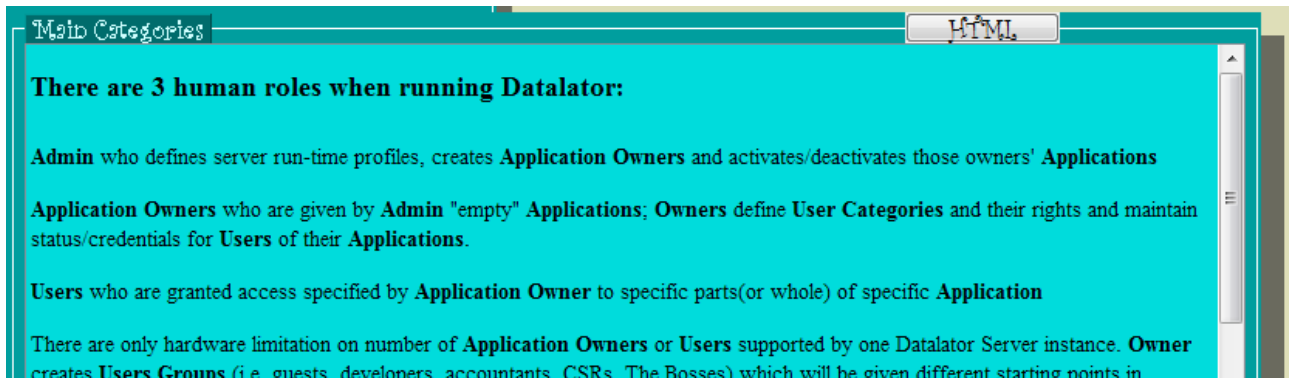
- a) Log Files delivery from the server and visualization (requires programming)
- b) An item description in a Market application
- c) Employee history keeping in HR type application
- d) Bugs description in Bugzilla type applications
- e) Forums, chats and "contact us" applications (requires programming)

Note: a Text Layer by default equipped with a "Save" button (visualized when keyboard input detected). Without programming you cannot prevent anybody to change your texts.

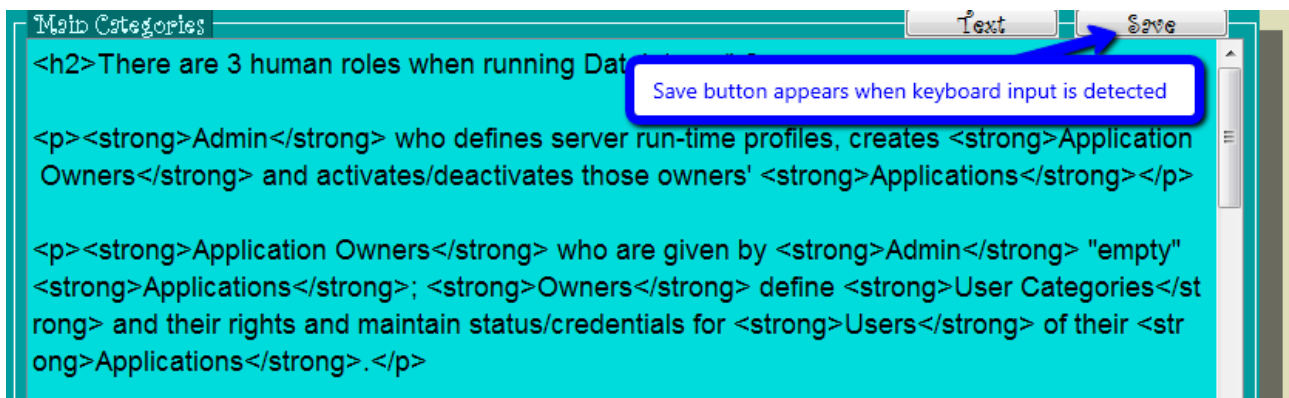
13. HTML Layer.

HTML Layer is derived from a Text Layer and found in the same Use Cases. The difference is that HTML Layer will allow you to type in html code as plain text and visualize it as an HTML page. You may insert pictures by coding URLs the usual html way.

Here is an example of HTML layer in "html" mode:



And here we have the same editable content in "plain text" mode:



The functionality of the HTML layer is defined by abilities of Java Swing JEditorPane without processing html links. This is not a web browser, just an simple HTML editor. Although with some programming it is possible to make that Layer much more able and sophisticated.

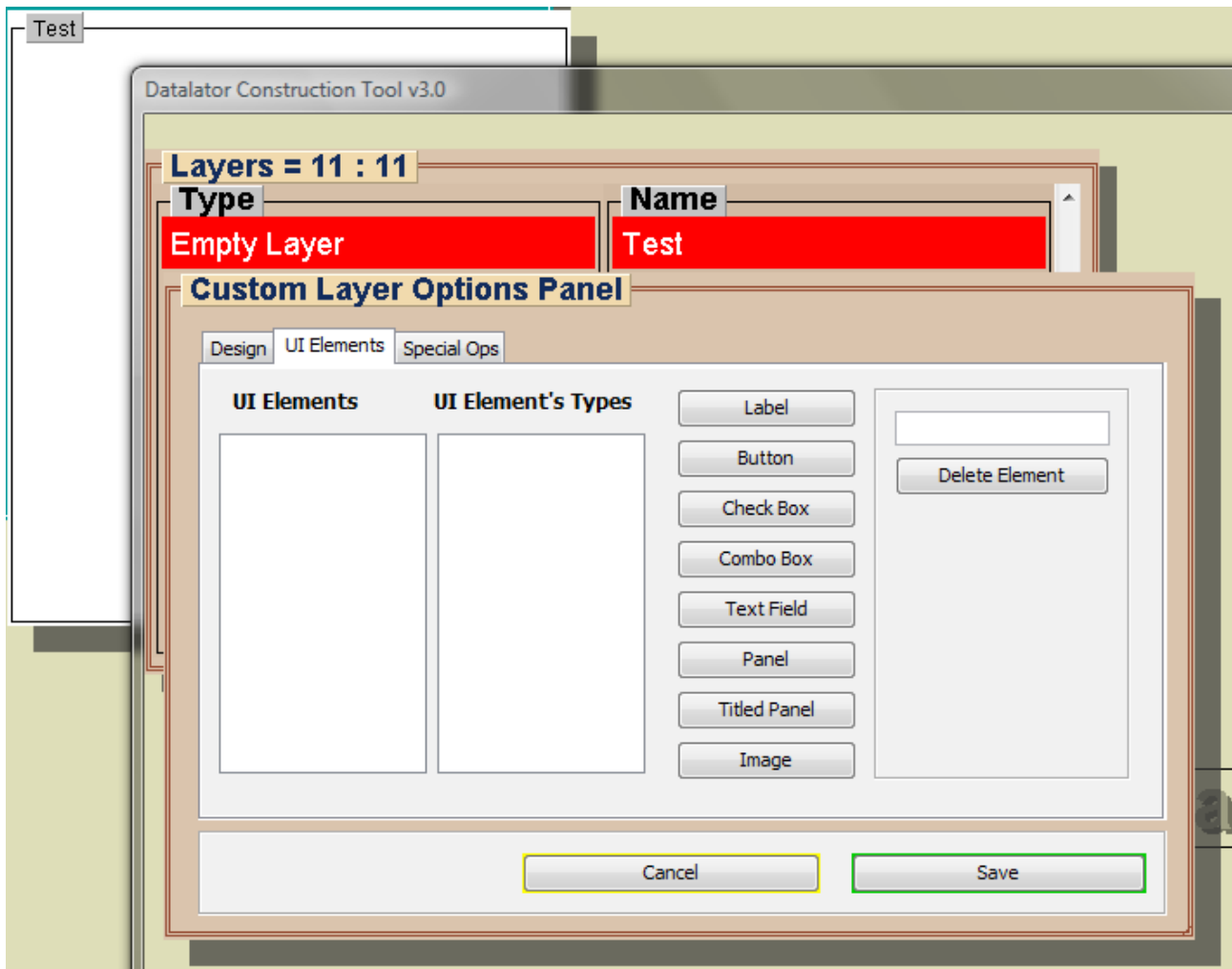
Note: as with the Text Layer you cannot prevent people to change your HTML content without programming (a few lines of code but still...) - disabling the "mode and "save" buttons.

Note: as with the Text Layer the content of HTML layer is encoded and kept in the application database. The pictures (coded with <image> tag) have to reside outside the database and to be served by a separate WebServer.

Note: as the Text Layer, the HTML Layer is a "terminal" one – you cannot define "enter" operation into another layer, well, without programming. A Service Menu is supported for Text and HTML Layers.

14. Custom (Empty) Layer.

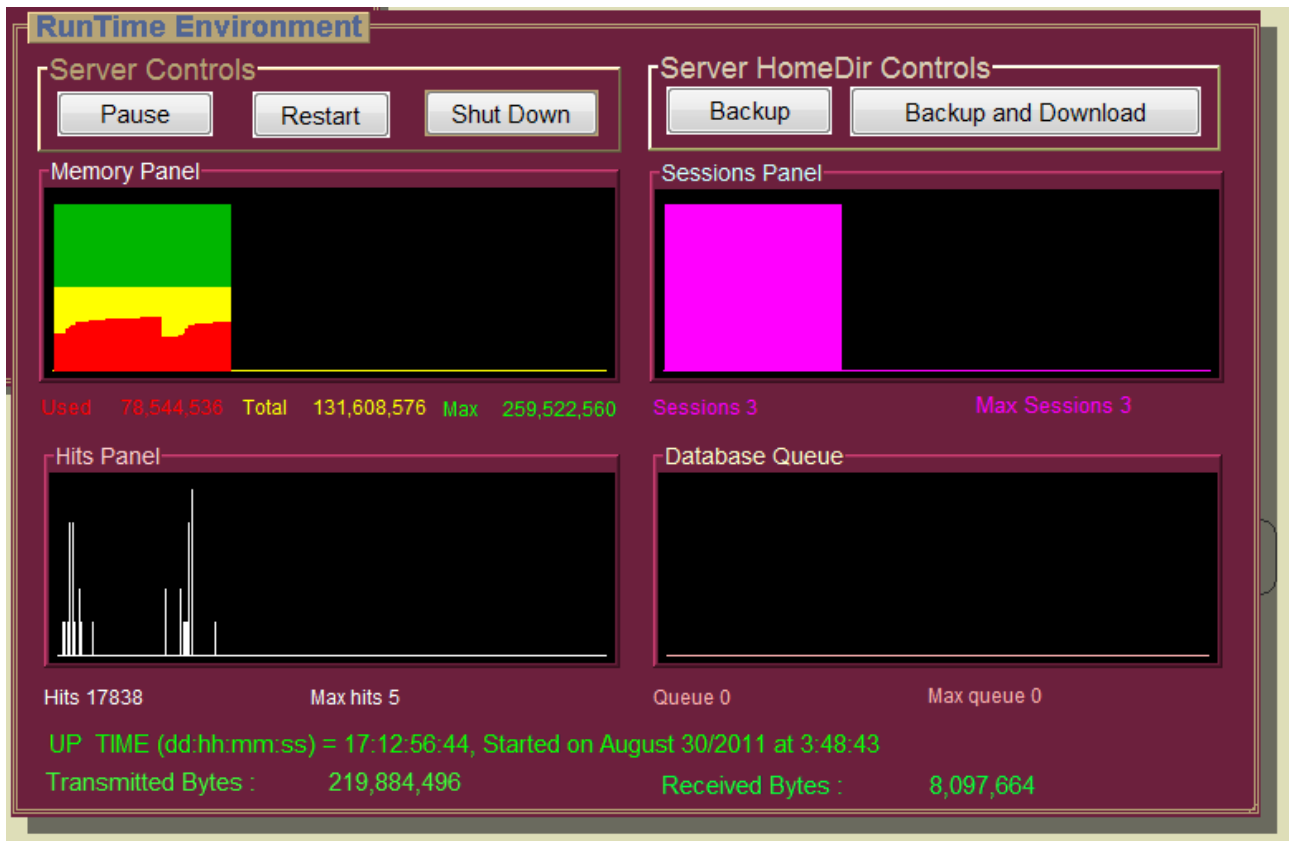
You may define a type of a Layer as "Empty" – and it will do nothing for you, will just "opens" when you "enter" the line where you placed it and will "escape" back. It is pretty useless without programming at all. We cannot imaging what you ought to use it for, although you might place a lot of controls on the panel of that layer – check the "UI Elements" tab of the Construction Tool:



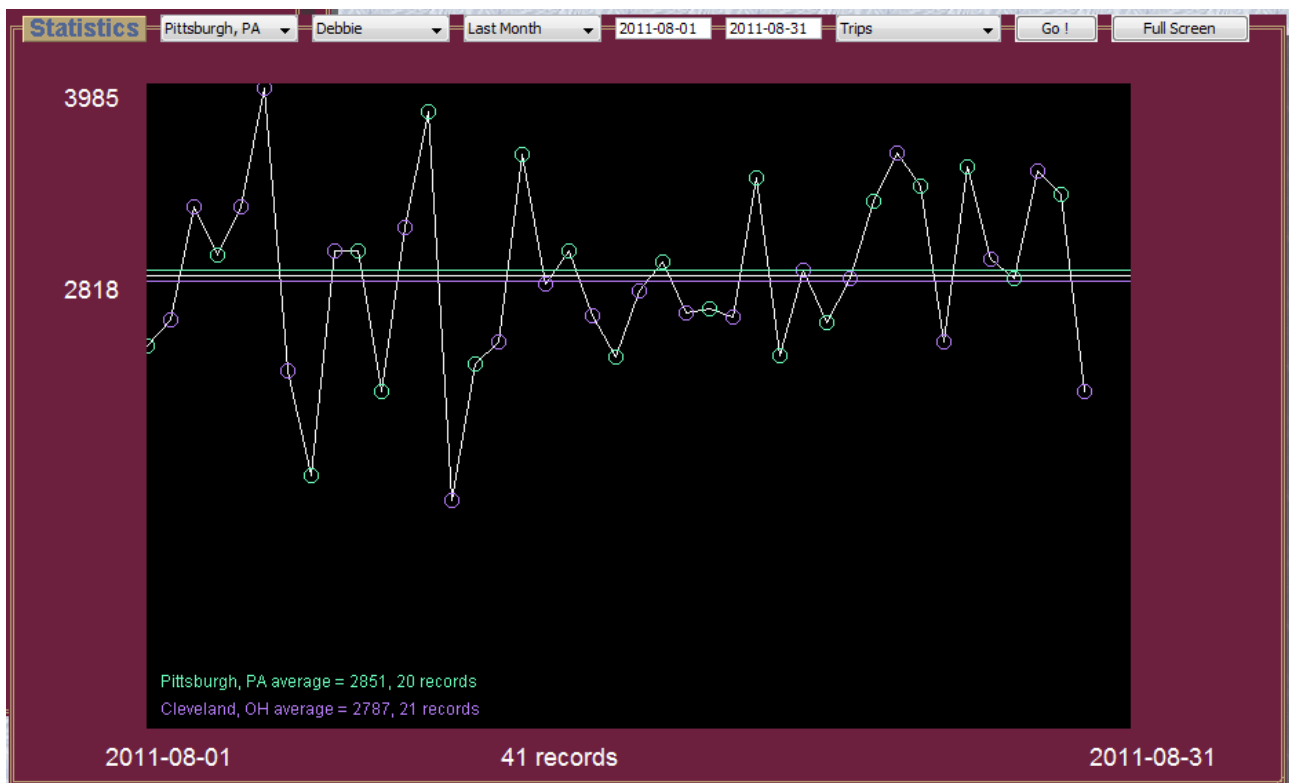
But it is very useful when you are willing to program the layer. All controls might be accessed from the Java code and all user actions processed and responded to. The Custom Layer is the best shot when you have to visualize your data not in the form of a Table, but beyond. Whatever data type is not supported by existent 6 types of layers might be expressed in a Custom("Empty") one – by programming it of course. The best known usages are the:

- a) dashboards
- b) chartboards
- c) gameboards

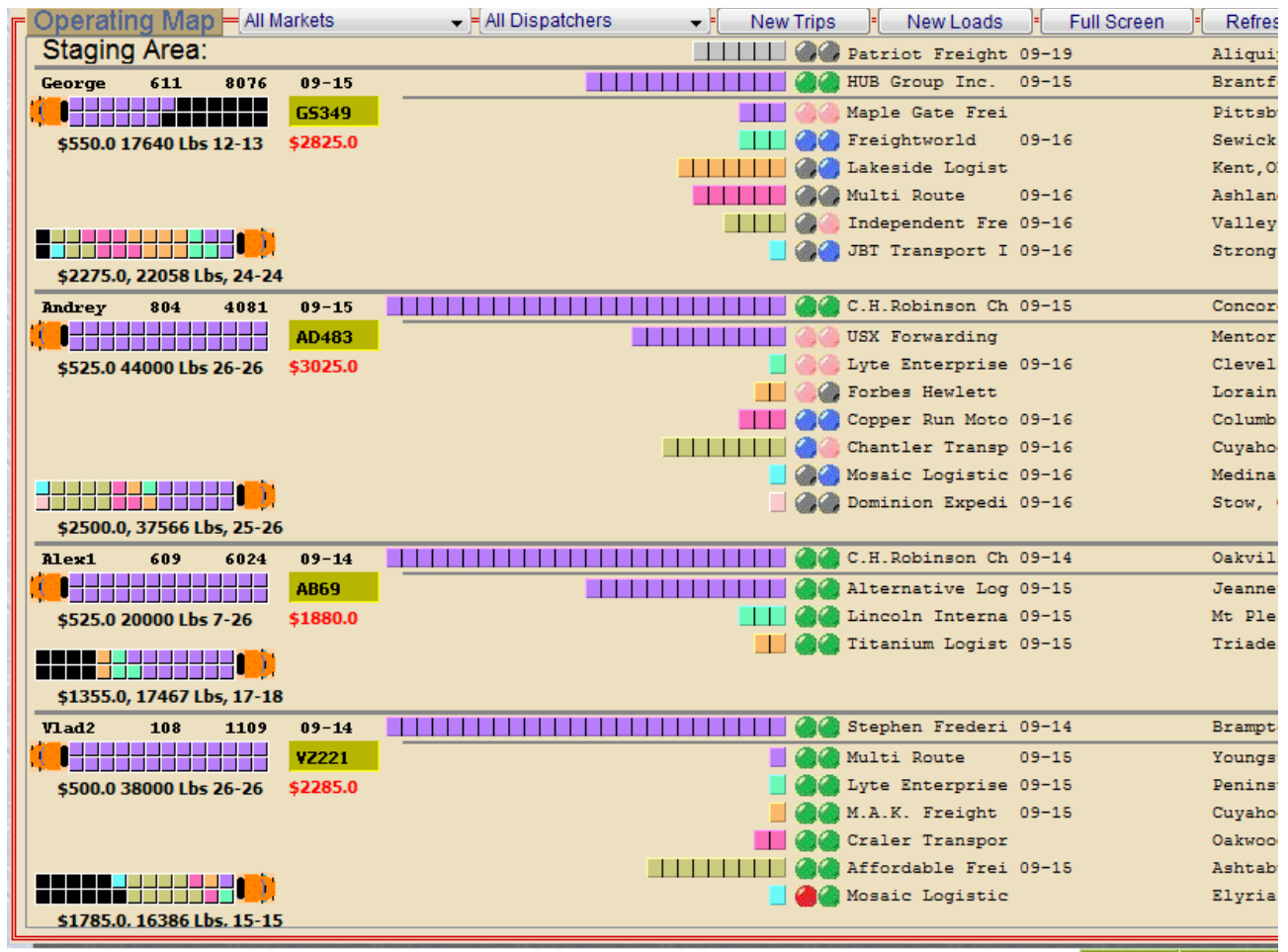
Here is an example of a dashboard:



and a chartboard:



and a gameboard (in this case its a real-time multiuser "transport dispatcher "game):



15. Layers under development.

We have discussed how to use 6 types of components: Menus, plain and Filtered Tables, Text, HTML and Custom Layers. First meaningful commercial applications were built times ago, when there were only Menus and plain Tables. Obviously, the components set is not full or ideal now and probably will never be. Still it was useful before and is getting better with every new release.

There are a few useful types of Layers (or components) in different stages of development right now. Some of them of general purpose like a "tree" component to better display hierarchies and "dialog" component to better control an application business logic. Some are about accessing more data types, like a player of Adobe (www.adobe.com) "pdf" and flash files. A player of "MP3" music and "MP4" movies seems to be handy. And others are about better collaboration: real-time whiteboards and notebooks, voice and video chats.

Sky is the limit...