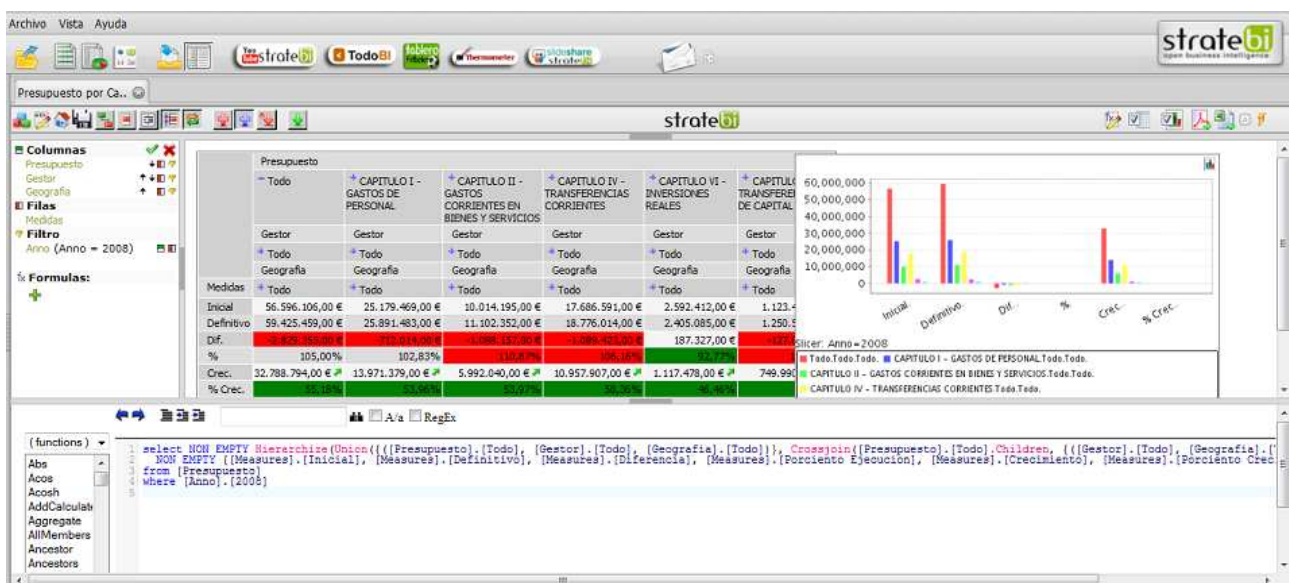


Introducing STPivot



January 2012

Copyright (c) 2011, Stratebi. All Rights Reserved!

This document is copyright © 2011 Stratebi. No part may be reprinted without written permission from Stratebi. All trademarks are the property of their respective owners.

About This Document

If you have questions that are not covered in this guide, or if you find errors in the instructions or language, please contact the Stratebi Technical Publications team at info@stratebi.com. The Publications team cannot help you resolve technical issues with products.

Support-related questions should be submitted through the STPivot Home Page at

<http://code.google.com/p/stpivot>

Limits of Liability and Disclaimer of Warranty

The author(s) of this document have used their best efforts in preparing the content and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these programs or the documentation contained in this book.

The author(s) and Stratebi shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims.

Company Information

Stratebi (www.stratebi.com/en/) are the main experts in Open Source Business Intelligence solutions in Spain, with a lot of client references, and their own developments.

They hold the main blog and social web about Open Source Analytics: Todobi (www.todobi.com) and Redopenbi (www.redopenbi.com)

You can check several **Online Demos** based on Open Source BI here: <http://sample.stratebi.es> and <http://www.tablerofutbolero.com>

Table of Contents

1.	Introduction	3
2.	Pentaho's Pivot viewer pros & cons	4
3.	STPivot Features	5
3.1	Rearranged user interface	6
3.2	Ajaxified user interactions	9
3.3	Easier Chart manipulation	10
3.4	Quick and easy stats	11
3.5	Enhanced MDX Query editor	12
3.6	Formula Editor dialog	14
3.7	More features	15
4.	Roadmap	16
5.	Final Thoughts	17

1. Introduction

STPivot is an open source web based OLAP viewer, developed by Todo BI – Stratebi, based on default Pivot viewer provided by Pentaho, written on top of JPivot. The idea behind this project was to improve JPivot user's experience, by taking advantage of free user interface libraries and technologies (such as jQuery and Ajax).

Although default Pivot viewer in Pentaho have successfully pass the test of time (fully simple, functional and crossbrowser), in Stratebi our clients constantly asked for improved versions (closer to the state of the art). That's why we develop STPivot, giving special attention to user interface and interactions (foreground), while the background remains basically the same.

This document have been created to introduce STPivot, so you can learn how it improves JPivot and hopefully choose to try it in your own environment.

2. Pentaho's Pivot viewer pros & cons

JPivot was the first open source olap client that communicates with mondrian (the leading open source olap server). It has been for years the tool of choice for many projects that wanted to interact with data cubes, and is the foundation for Pentaho's Pivot Viewer in the Community Edition version. It was also the olap viewer for the Enterprise Edition, until the arrival of Pentaho Analyser.

Default Pivot viewer evolve for years to become a very complete and stable solution, that allow you to freely interact with OLAP data served by Mondrian. From a technical point of view, and considering the history written by JPivot, it is very awkward for us to criticize this solution. Nevertheless, most users continuously ask for fresher alternatives near to the state of the art, to navigate their data cubes. Most recurrent complains included:

- “It is not easy to use”
- “It looks very old”
- “It lack features we can find in privative software”

Most complains were always subjective, but they expose feelings indicating that something must be changed.

3. STPivot Features

When initially defined, STPivot's goal was to construct a fresher web based OLAP client solution, specially focused on ease of use. JPivot's users will find lot of coincidences when using STPivot, and that's because we reused most of hard work done by JPivot, while changing some parts of it surface.

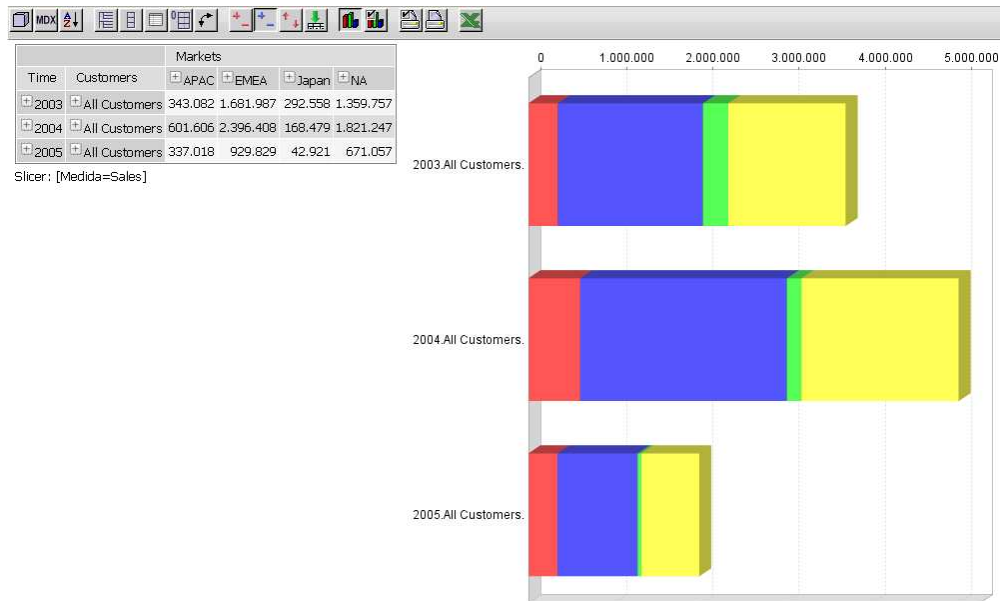
Main features include:

- Rearranged user interface
- Ajaxified user interactions
- Easier Chart manipulation
- Quick and easy stats
- Enhanced MDX Query editor
- Formula editor dialog

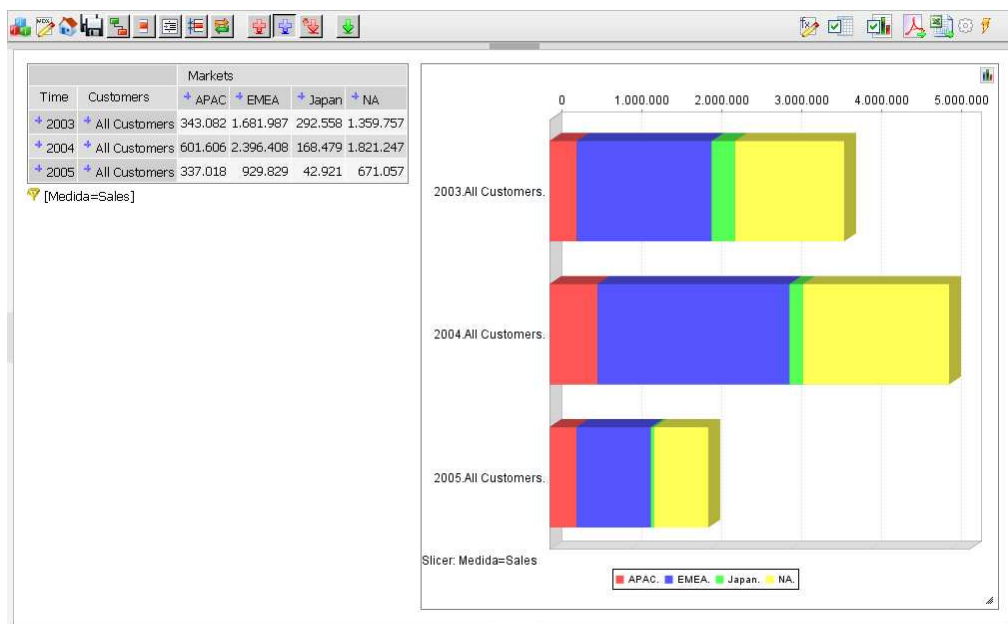
Now you'll learn more about each of them, including some additional minor features.

3.1 user interface

Appearance was identified as a problem, giving an image like following picture.






















Almost every feature available is presented to users in the tool bar, drawing forms on top of page to change configuration. STPivot look very similar at first sight, reordering buttons and moving some of them apart. Next picture shows how it looks.

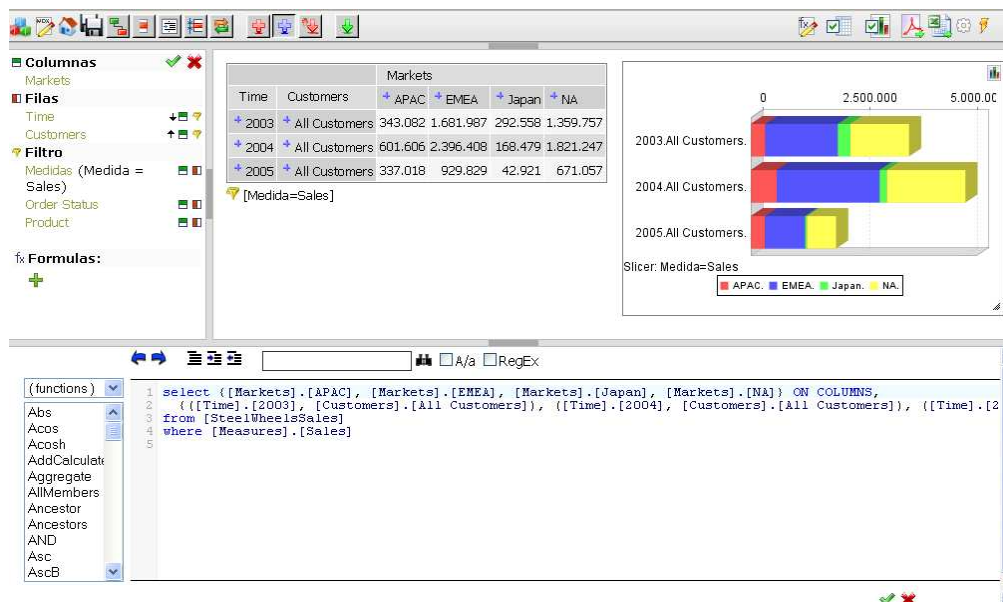


Some buttons have gone, while others have been added. Most of them remains the same, despite the different image used (fresher look). Next table contains equivalences between both STPivot and JPivot tool bar buttons. It is not the purpose of this document to explain each functionality, but you'll find it in STPivot's User Manual.

STPivot	JPivot	Description
		Similar, except for that STPivot show/hide Navigator panel.
		Similar, except for that STPivot show/hide MDX editor panel.
	-	New in STPivot, returns to the first run Query
	-	Included in Pentaho's Pivot viewer, reproduced in STPivot.
		Exactly the same.
		Exactly the same.
		Exactly the same.
		Exactly the same.
		Exactly the same.

		Exactly the same.
		Exactly the same.
		Exactly the same.
		Exactly the same.
	-	New in STPivot, shows/hide the Formula Editor dialog.
	-	New in STPivot, shows/hide the grid (MDX Table)
		Exactly the same.
		Exactly the same.
		Exactly the same.
		In STPivot opens a dialog with diferent (rarely used) configuration forms.
	-	New in STPivot, clears cache

Another important addition is the use of sliding panels for Cube Navigator and Formulas (left panel), MDX Editor (bottom panel) and the Toolbar (top panel). This way users can maximize the available space for table and/or chart. You can see all panels opened in the following picture.



3.2 Ajaxified user interactions

Ajax stand for “Asynchronous Javascript and XML”; it is not a product, but a collection of technologies that can be used to build web pages that are more interactive than has been possible in the past. Web pages built with AJAX can be made to feel more like application than a web page. Technology comprises:

- Asynchronous Javascript: This means that script within the web page can make calls to a server to get information without re-loading the web page. This significantly improves the usability and performance of complex web sites because the contents of the page can interact with the user without having to reload HTML, style-sheets, and images.
- XML: This refers to the fact that the result of the call to the server can be XML. Actually it can be any kind of text such as HTML or Javascript object notation (JSON) or just a response message.
- Dynamic HTML: It allows the web pages to be dynamically altered by the script based on its conversation with the server. This can be used to change images, add or remove rows to tables, hide and show tabs, and move or change text on the page.

Almost every user interactions in JPivot causes full reload of page, but this behavior becomes unacceptable when we decided to include JavaScript libraries in STPivot (penalizing initial load of page, looking for a better user experience). That's why we reload only the parts of the page that should change with each type of user interaction.

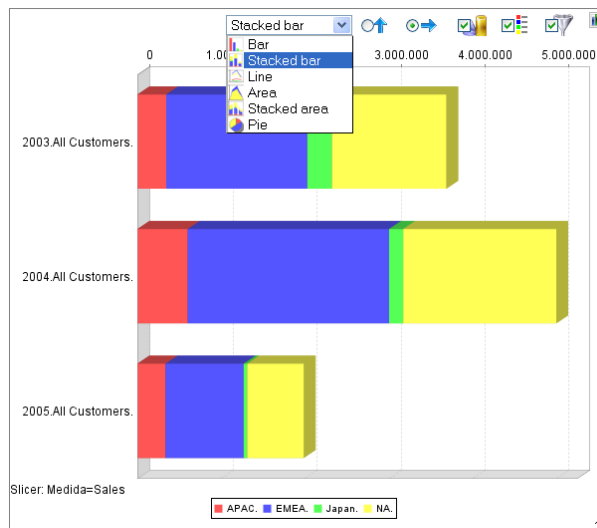
Now users experience a minor extra delay when opening the viewer by the first time (compare to classic Pivot viewer), but they will witness a better performance when interacting with STPivot.

3.3 Easier Chart manipulation

In order to manipulate the Chart output, JPivot presents the user with a complex form that you can also find in STPivot, as shown in the following pictures.

STPivot	JPivot
<div><div>Chart Axis Sort Print</div><div>Chart Type: Stacked Horizontal Bar 3D</div><div>Chart Title:</div><div>Chart Title Font: SansSerif Bold 18</div><div>Horizontal axis label:</div><div>Vertical axis label:</div><div>Axis Label Font: SansSerif Plain 12</div><div>Axis Tick Label font: SansSerif Plain 12 30°</div><div>Show Legend: <input checked="" type="checkbox"/> Bottom</div><div>Legend Font: SansSerif Plain 10</div><div>Show Slicer: <input checked="" type="checkbox"/> Bottom Left</div><div>Slicer Font: SansSerif Plain 12</div><div>Chart Height: 213 Chart Width: 374</div><div>Background (R, G, B): 255 255 255</div><div>✓ ✗</div></div>	<div><div>Chart Properties</div><div>Chart Type: Stacked Horizontal Bar 3D</div><div>Enable Drill Through: <input type="checkbox"/></div><div>Chart Title:</div><div>Chart Title Font: SansSerif Bold 18</div><div>Horizontal axis label:</div><div>Vertical axis label:</div><div>Axis Label Font: SansSerif Plain 12</div><div>Axis Tick Label font: SansSerif Plain 12 30°</div><div>Show Legend: <input checked="" type="checkbox"/> Bottom</div><div>Legend Font: SansSerif Plain 10</div><div>Show Slicer: <input checked="" type="checkbox"/> Bottom Left</div><div>Slicer Font: SansSerif Plain 12</div><div>Chart Height: 600 Chart Width: 600</div><div>Background (R, G, B): 255 255 255</div><div>OK Cancel</div></div>

Except for the Drill Through option (disabled in STPivot for challenges in implementation), both forms are very similar. But some of these options, those most frequently used, have been moved to the chart dialog, so it is easier for users to change them. These shortcuts are not always visible, only when clicking in the small image at the top-right corner (in the chart dialog). Next picture shows these options when visible.

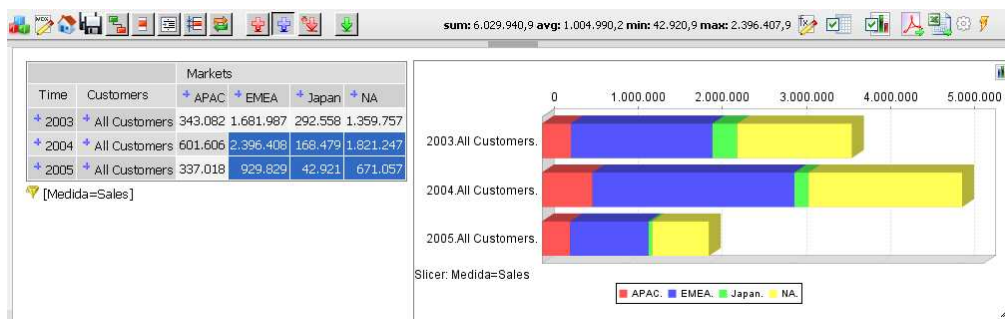


JPivot's form lets you choose amount 16 different types of charts. This new form lets you choose amount 6 types of charts, together with orientation (horizontal/vertical) and 3D attribute; which results in a more easy way to get type of chart desired.

Now you can quickly show/hide legend, as well as the slicer. But perhaps the most appealing improvement is the possibility of resize current chart by dragging the edges of it dialog. Users can also drag&drop the entire chart dialog to the desired location, according to their needs.

3.4 Quick and easy stats

One handy feature you can find in desktop OLAP clients (such as La_Azada, formerly Jrubik), is the possibility to select some cells in the table and get important stats such as sum, average, minimum and maximum. Next picture shows how STPivot looks when selecting multiple cells.

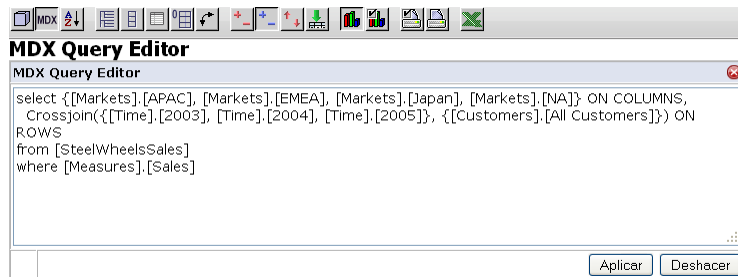


Currently statistical summary goes into the tool bar, limiting space and layout, which is why we plan to move this to a floating dialog, perhaps accommodating more relevant information.

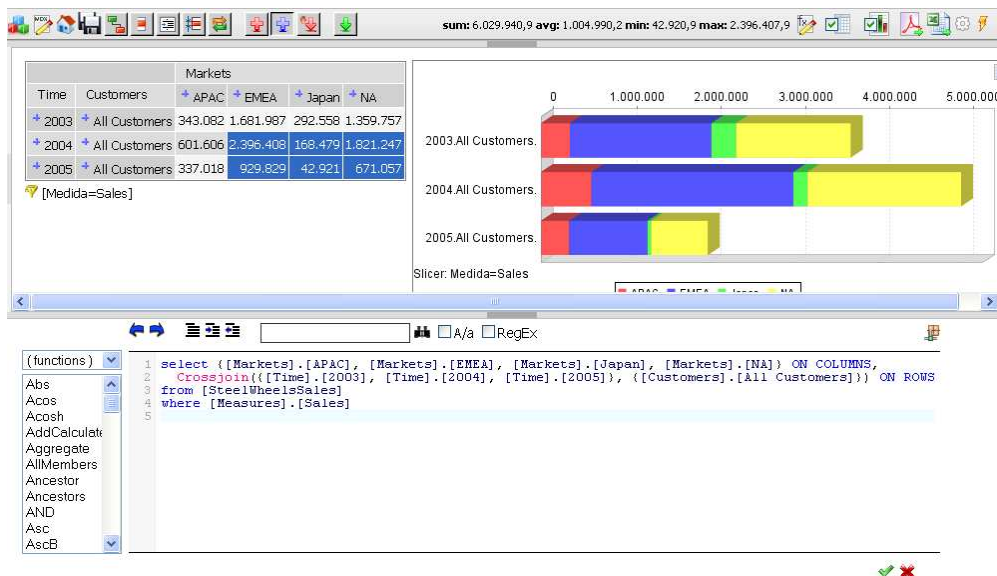
3.5 Enhanced MDX Query editor

Original MDX Query editor consist of plain form with text area to read/write current query. This was enhanced in STPivot's beta version using CodePress, which uses a custom MDX language parser to highlight reserved words and symbols. This solution had faults in some navigators (like Chrome) and left some problems unsolved. That's why current version uses CodeMirror, adding new advanced features to the editor.

This is how original MDX Query editor looks like:



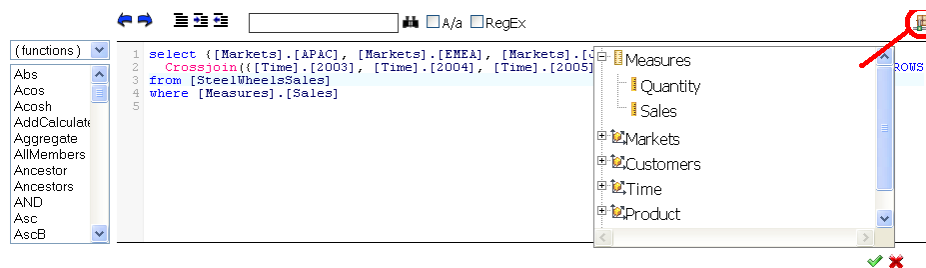
And this is how current (STPivot) MDX Query editor looks like:



As stated before, the editor was moved to the bottom, into a panel that users can show/hide. But also:

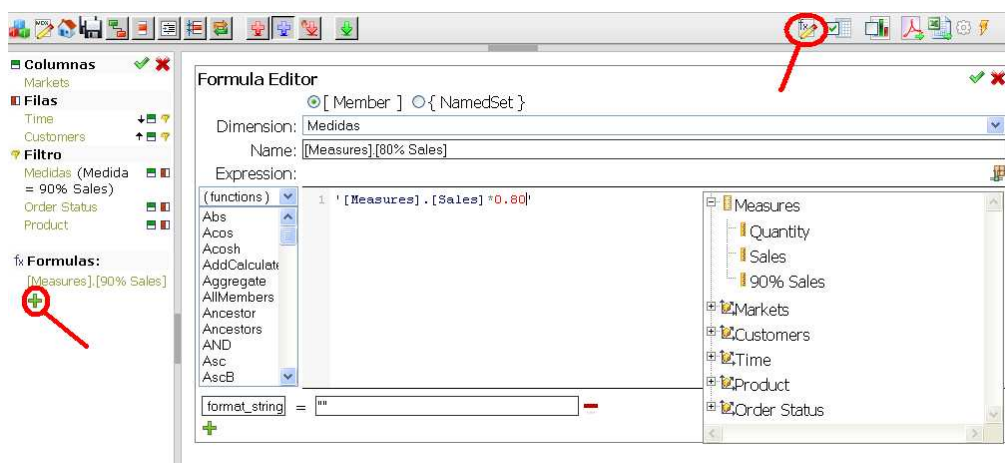
- Highlighted syntax results in a more easy to read/write MDX query.
- Common editor options like undo/redo, indentations and search are now available. Optionally the search can be case sensitive, or be based on a regular expression pattern.
- Code completion allows to quickly find functions and reserved words ready to use, just by pressing ctrl + space
- There is a visible selector (left side) for MDX functions, categorized conveniently according to its result type (one function may fit into many categories).
- An interactive cube navigator have been added at top-right corner, allowing users to explore dimensions, hierarchies, levels and even members, with the possibility to add their name by double clicking. They also replace a selection range by new text selected, as expected in any desktop editor.

What follows is a picture where cube navigator is open and Measures dimension explored.



3.6 Formula Editor dialog

This feature allows users to add/edit calculated members or named sets, that are added to the query in the form of “with ... member ...”. This entities list, also known as formulas, appears below the cube navigator in left panel. In the picture you can see the Formula Editor dialog, along with red marks indicating where are the buttons to open it.



If the formula is a calculated member, users can indicate its dimension to have a consistent name proposal (that he can also change). But if it is a named set, no dimension will be necessary.

The expression editor is similar to the MDX Query editor, only a bit simpler, including the cube navigator and formula selector features. No other code completion is currently available.

Calculated members may have multiple properties, such as FORMAT_STRING and SOLVE_ORDER for instance; so users can add them directly in the expression editor, or optionally use the dynamic list of property+value at the bottom.

3.7 More features

There are additional features perhaps less important and hard to use, but potentially useful for the experienced/advanced users. This features include:

- **Possibility to externally interacting with the viewer.** It means that you can inject a new query into STPivot, especially design using a customized form for an specific domain, refreshing the entire view without reloading the entire page. This is used in what we call Guided Adhoc Analysis, where the pivot view is inside an iframe, in a page that holds the custom form.
- **Parameterized links on members.** JPivot allows you to link members using different patterns, but you need to define a custom viewer in order to use this feature. STPivot expect necessary information in an special input variable, and dynamically build the list of clickable members. You can even set multiple links on the same member, appearing in the form of pop up dialog links.
- **XML/A access mode.** Default Pivot viewer assumes Mondrian as dimensional data source, but JPivot allows you to access XML/A sources. That's why STPivot is now able to interact with XML/A sources, simply indicating the right information y certaing parameters.
- **New set of icons.** We are constantly looking for better icons to represent each function and promote usability.

Some features in STPivot appears hidden and hard to uncover, which is way they'll be fully explained in a set of sample and tutorial documents, that hopefully will help its adoption.

4. Roadmap

We are constantly looking for improvements to STPivot, putting special attention in feedbacks from final users. Current roadmap includes:

- Release a new branch independent from Pentaho, so other projects using JPivot can also benefit from STPivot.
- Upgrade charting engine, perhaps using some new javaScript library such as HighCharts.
- Build a new PivotComponentView for Pentaho, to enhance definition of clickables and XML/A mode
- Improve stats summary when selecting table cells
- Rewrite the whole frontend, using dojo as the foundation for user interface.
- Replace jpivot tags and libraries for those provided by Olap4j, along with the possibility to interact with a wider range of OLAP providers.

5. Final Thoughts

STPivot has come to live only because of the great work done by JPivot and its classic Pivot viewer. We should also recognize the work done by the team of JRubik and La_Azada (The Hoe), as reference implementation for a desktop based OLAP client, which gave us many ideas to improve JPivot.

In Stratebi, our team knows that there is a lot of work to do, regarding web based OLAP client, but we are very happy releasing this project and contributing to the Open Source and Business Intelligence communities.

Please, feel free to visit the project home (<http://code.google.com/p/stpivot/>) to give your feedback and contributions.