

Paper 379-2013

The Dynamic Cube Viewer - OLAP Made Easy

Raymond Ebben, OCS Consulting

ABSTRACT

The Dynamic Cube Viewer is a bespoke browser-based application that offers an intuitive interface for business users to query OLAP cubes, without the need to have an understanding of OLAP cubes. It has originally been developed as a benchmarking tool for the Association of Dutch Insurers and has been further developed by OCS Consulting to make it more generic. The application only reads OLAP cube metadata and uses this to build the user interface.

INTRODUCTION

This paper describes the history and the evolution of the Dynamic Cube Viewer, followed by the functional and technical overview.

THE HISTORY

The Dynamic Cube Viewer finds its origin in a bespoke application that was initially developed in SAS® 9.1.3 for one of our customers; The Association of Dutch Insurers. Its purpose is to simplify and extend the functionality offered by the SAS® Web OLAP Viewer (see Display 1. SAS® Web OLAP Viewer). The client had reviewed the SAS® Web OLAP Viewer and concluded that this application was not suited for the intended end users.

The screenshot shows the SAS Web OLAP Viewer interface. The main window displays a table with the following data:

Month		Dec
		Total Actual
Country	State/Province	
Canada		\$155,964.10
Mexico		\$110,468.70
U.S.A.	California	\$51,997.90
	Colorado	\$55,317.00
	Florida	\$54,794.80
	Illinois	\$272,218.70
	New York	\$52,138.20
	North Carolina	\$47,844.60
	Texas	\$44,770.10
	Washington	\$52,165.60

Display 1. SAS® Web OLAP Viewer

The intended end users are insurers who have provided input to the Association of Dutch Insurers. The information provided by all participating insurers is processed and from this a market total is derived. The purpose of this application is to make this information available to the insurers again.

These insurers are information consumers who are used to process and analyse data, but do not necessarily have affinity with OLAP cubes.

The main requirements for the initial version were:

- The application must be intuitive and user friendly;
- For some cubes, certain combinations of dimensions are not allowed. This must be implemented in the application;
- The application must allow the users to “benchmark” the information they provided against the derived market total;
- As this is sensitive information, security and traceability needs to be implemented.

This resulted in the initial version of the Cube Viewer, which is displayed below:

Display 2. Initial version of the Dynamic Cube Viewer

The different elements on the screen are:

1. The list of all available dimensions. Hierarchical dimensions are listed as individual items (e.g. The dimension year and the dimension year, month are two separate entries);
2. Dimensions dropped here by the user will be presented in the table row. The order in which they are displayed is the order in which they are presented;
3. Dimensions dropped here by the user will be presented in the table column. The order in which they are displayed is the order in which they are presented;
4. Select which measures will be included in the requested table;
5. Select if the measures will be presented in the row or in the column of the requested table;
6. Select the company to report on (Security ensures that only the companies the user has access to will be available);
7. Select if the market total needs to be included in the requested table (benchmark the results);
8. Select the output type: On screen (HTML), Excel or PDF;
9. Generate the requested table.

The Traceability has been implemented in a SAS® program that deletes the market value if the criteria for traceability are met. When a combination is requested by the user that is not allowed, the application displays a warning message to the user and does not process the request.

THE EVOLUTION

Since the initial version, several enhancement requests have been received. The most important enhancement requests are:

From a user's perspective

- Allow the user to filter (slice) the dimensions. This means that a list of distinct values will be presented when a user clicks the new dropdown button, added to the dimension container;
- Extend the drag and drop functionality by adding buttons to the dimension container to move a dimension to a row or a column;
- Extend the drag and drop functionality by adding buttons to the dimension container to change the display order within a row or column.

From an administrator's perspective

- Ensure the preparation process for a cube is generic;
- The ability to exclude one or more cube dimensions from the application;
- The ability to exclude one or more cube measures from the application;
- The ability to setup a dimension as a single selection dimension. This means that the user can select only one value to filter the dimension;
- The ability to setup a dimension as a mandatory dimension. This means this dimension must be included in the users' selection.

These enhancements have been requested and implemented over a period of more than 5 years. Some of these enhancements were implemented in a cost effective and efficient manner, but would have been implemented differently if we had included this functionality from the start.

For example the process to prepare a cube has been made generic, but the application reads all the required information from a SAS dataset. Often this dataset is not updated correctly due to the many switches and settings and has no direct link to the actual OLAP cube.

This knowledge, together with the new capabilities of the software used (SAS®, HTML, JavaScript and CSS), inspired us to develop a brand new version of the Dynamic Cube Viewer, utilising the new capabilities of the software. This application is described in the next chapters.

THE OCS DYNAMIC CUBE VIEWER – FUNCTIONAL OVERVIEW

The OCS Dynamic Cube Viewer is a bespoke browser-based application that offers an intuitive interface for business users to query OLAP cubes, without the need to have an understanding of OLAP cubes. The application reads only OLAP cube metadata and uses this to build the user interface. An example is shown below:

View data in cube: Orion Sales

The interface is divided into several sections:

- Dimensions:** A list of dimensions (Product Line, Product Category) with a red circle 1 next to it.
- Compare with:** A checkbox labeled "All product groups" with a red circle 5 next to it.
- Columns:** A list of columns (Product Group) with a red circle 3 next to it.
- Rows:** A list of rows (Year, Quarter) with a red circle 2 next to it.
- Measures:** A list of measures (Total Retail Price, Profit, Quantity) with a red circle 4 next to it.
- Show measures in:** A radio button selection for Columns (selected) and Rows.
- HTML table:** A table displaying data with a red circle 6 next to it.

		Total Retail Price	
		All product groups	Badminton
Year	Quarter		
1999	1999Q1	4825072.51	18203.80
	1999Q2	7463957.09	12360.10
	1999Q3	6881995.06	17102.50
	1999Q4	5854399.32	30234.50
2000	2000Q1	5758640.42	25382.10
	2000Q2	8857843.56	17076.50
	2000Q3	8690722.13	25360.80
	2000Q4	7166970.35	37390.20
2001	2001Q1	4848709.81	17447.90
	2001Q2	7681910.78	15037.50
	2001Q3	7276656.94	16688.30
	2001Q4	6104915.30	36130.60
2002	2002Q1	5026353.06	18493.30
	2002Q2	8359585.28	16947.40
	2002Q3	8526526.15	22909.30
	2002Q4	8022446.14	39778.90

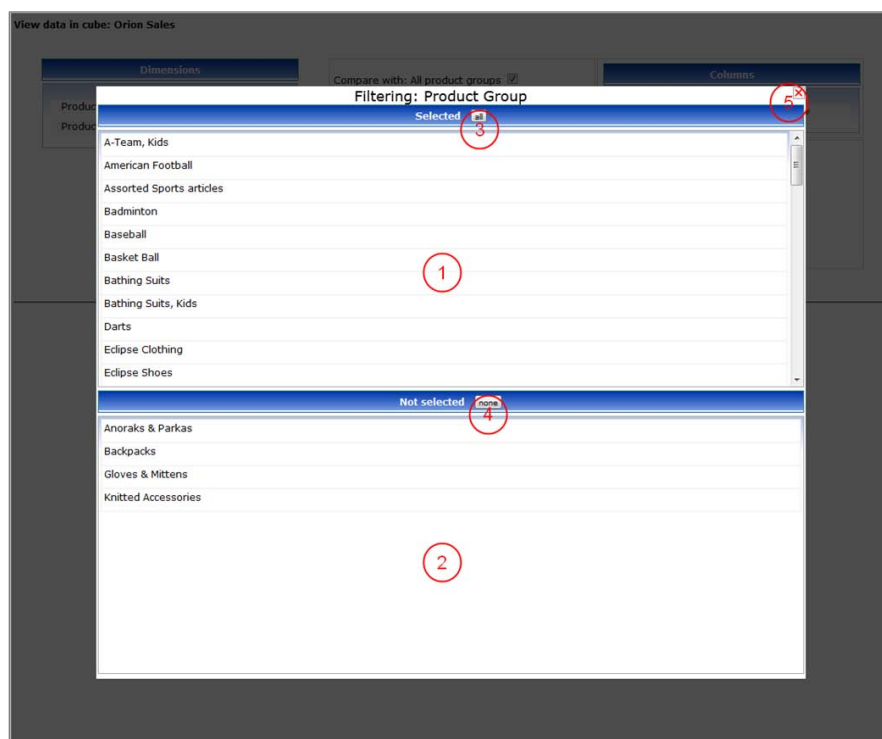
Display 3. The OCS Dynamic Cube Viewer

The user interface displays the following elements:

- ① Dimensions: Contains the list of available dimensions. Hierarchical dimensions are handled as a bundle (see ② Year Quarter as an example).
- ② Rows: Dimensions dropped here by the user will be presented in the table row.
- ③ Columns: Dimensions dropped here by the user will be presented in the table column.
- ④ Measures: The user can select which measures will be included in the requested table.
- ⑤ Output selection: The requested output type (HTML, Excel, PDF or RTF). The "Compare with" checkbox will allow benchmarking (In this case an individual shop is compared against the total of all shops).
- ⑥ HTML table: The requested table in case HTML has been selected as the requested output type.

The dimension values can be filtered by clicking on the filter icon, available for each dimension.

When for instance the product group dimension is clicked, a filter screen is displayed allowing the user to filter on the product groups to be displayed in the results table.



Display 4. The filter screen

- | | | |
|---|--------------------|--|
| ① | Selected area: | Contains all currently selected values. Clicking on any line here will move it to the Not selected area. ② |
| ② | Not selected area: | Here you see all currently excluded values of a category. Clicking on any line here will move it to the Selected area. ① |
| ③ | All button: | Selects all values (placing them in the selected values area ①). |
| ④ | None button: | Deselect all values (placing them in the excluded values area ②). |
| ⑤ | Close button: | Saves selection and closes filtering interface. Clicking in the grey area around the Filter interface has the same effect. |

The text of the individual display elements (headers, box names, tool tips etc.) are stored as part of the application in a separate list and supports the use of multiple languages.

Next to these basic functionalities additional functionality is available, so that the viewer can be fine-tuned for a specific cube. This functionality can be controlled using the extended metadata properties of an OLAP cube. These functionalities are:

- | | |
|-----------------------|---|
| Exclude Dimensions: | Dimensions, present in the cube can be excluded from the cube viewer. |
| Exclude Measures: | Measures, present in the cube can be excluded from the cube viewer. |
| Single Selection: | Dimensions that can only have 1 value selected in the Cube Viewer. |
| Mandatory Dimensions: | Dimensions that have to be selected in the cube viewer. |
| Benchmark Hierarchy: | Next to the measures for the selected values (for instance a specific product group), the totals for this dimension are also displayed (for instance the total of the product group). |
| Don't Combine: | For these dimensions a message will be generated when they are selected together. |

THE OCS DYNAMIC CUBE VIEWER – TECHNICAL OVERVIEW

This chapter zooms in on the technical aspects of the OCS Dynamic Cube Viewer.

The application itself is a SAS® stored process that uses SAS® Base code and SAS® Macro code to generate the HTML, CSS and JavaScript that makes up the user interface.

This interface uses AJAX® technologies to ensure only these areas on the screen that need to be updated, are updated. This makes the application really responsive and makes reloading the entire page each time a request has to be posted to SAS® obsolete.

The stored process is called with the name of the cube to be viewed and the language in which the user interface is to be presented as its only parameters.

With the cube's name, the application reads the SAS® metadata, including the bespoke extended attributes to determine the specific application setting for this cube.

If the application detects that the bespoke extended attributes are not available for the cube to be processed, the application will generate these attributes. This functionality has been added to assist the OLAP cube builder. Once a cube has been defined in SAS® OLAP cube studio, the developer loads the cube in the OCS Dynamic Cube Viewer, so that all available extended attributes are generated automatically, preventing possible typing errors. Then these values can be assigned from SAS® OLAP cube studio.

#	Field Name	Value	Description
1	ExcludedDimensions		
2	ExcludedMeasures		
3	SingleSelectDimensions		
4	MandatoryDimensions		
5	BenchmarkHierarchy		
6	BenchmarkText		
7	DontCombineDimensions		
8	WedgeProgramBeforeScreen		
9	WedgeProgramBeforeTable		
10	FormatsDirectory		

Display 5. The extended attributes

The first 7 attributes have already been detailed in the functional overview. Attribute 8 and 9 allow the user to specify a name of a SAS® program that will be executed either prior to generating the user interface or before generating the table. For instance, attribute 9 could be utilised to include a SAS® program that clears the values that meet the traceability criteria as mentioned in the initial client request. Attribute 10 holds the location of the formats, used to display the results in the table.

Using the extended attributes and generating them upon loading the user interface, should they not exist, allows for additional extended attributes to be added for future functionalities.

CONCLUSION

When an application is being actively used for a longer period of time and has undergone several enhancements during its life cycle, it is worthwhile to review the application to see if the application could be improved using today's knowledge and technology.

ACKNOWLEDGEMENTS

I would like to thank Khaled Al Haj Saleh from the Association of Dutch Insurers, our main contact within the company for which the original application was developed, my colleague Harro Vos, who is the developer of the OCS Dynamic Cube Viewer and has provided valuable input to this paper and my colleagues Jules van der Zalm and Yves Poriau who have provided the necessary feedback on this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Raymond Ebben
Enterprise:	OCS Consulting
Address:	Ruwekampweg 2G
City, State ZIP:	Den Bosch, 5222 AT, the Netherlands
E-mail:	sasquestions@ocs-consultng.com
Web:	www.ocs-consulting.com/nl

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.