

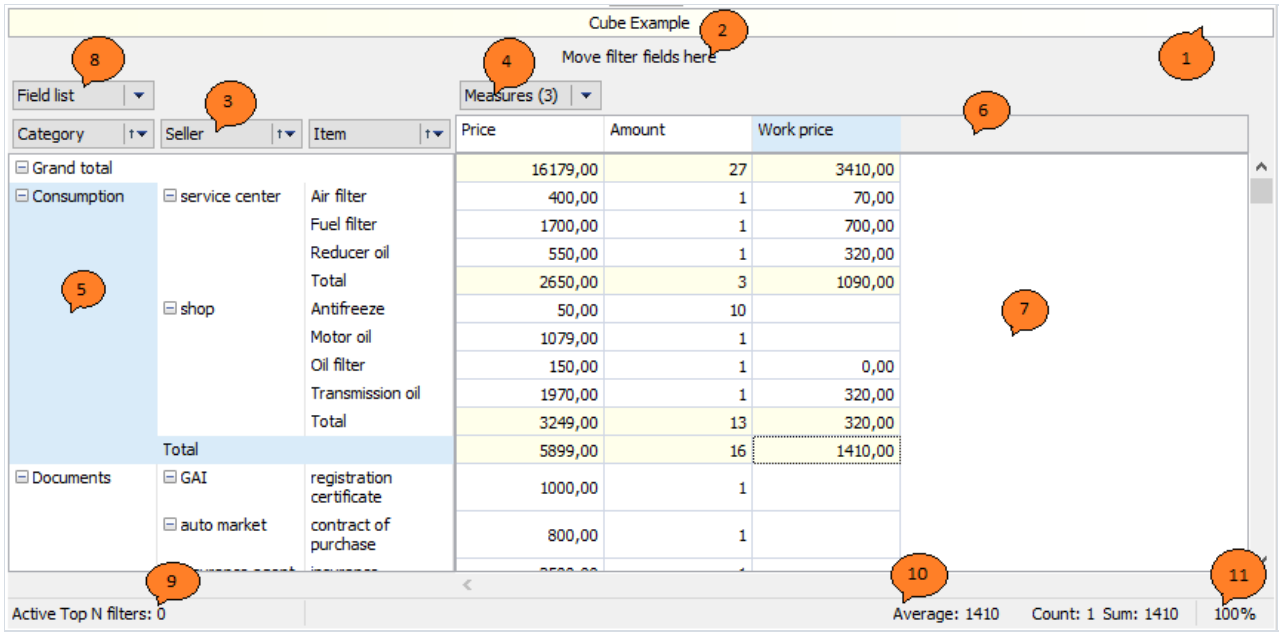
FastCube VCL User Manual

Version 2026.2

© 1998-2026 Fast Reports Inc.

The Grid

The Grid is an interactive tool for the presentation and analysis of data. There are multi-level headers in the left and top parts of the grid. The 'headers' are filled with the dimension values; there are both horizontal and vertical 'headers'. The central part of the grid displays the values of the measures.



The grid consists of several regions, identified by numbers above:

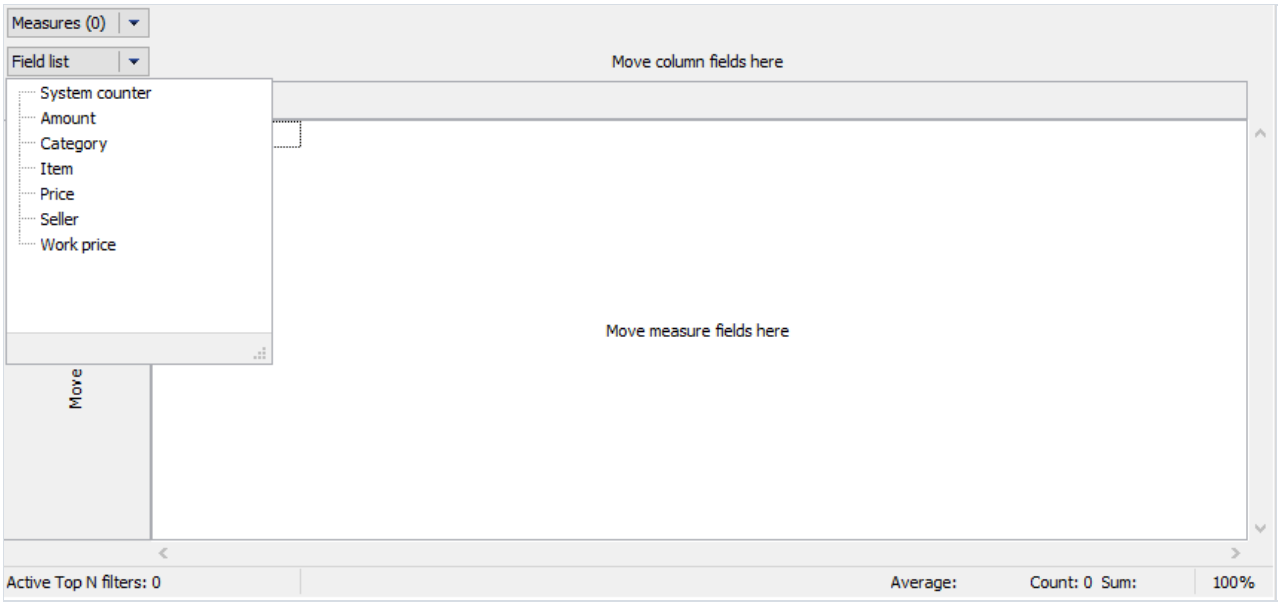
1. the grid title
2. the filter region : dimensions included in this region can be used for filtering the grid data
3. the vertical dimension headers : dimensions included in this region form the grid's vertical header
4. the horizontal dimension headers : dimensions included in this region form the grid's horizontal header
5. the vertical header of the grid
6. the horizontal header of the grid
7. data region
8. the list of fields : the fields required for processing are dragged from the drop-down list : fields not required are dragged back to this list
9. list of Top-N filters : filters can be modified through the context menu for this field
10. the region showing aggregates for the selected cells : can be modified through the context menu
11. the scale editor

Setting up the Grid Structure

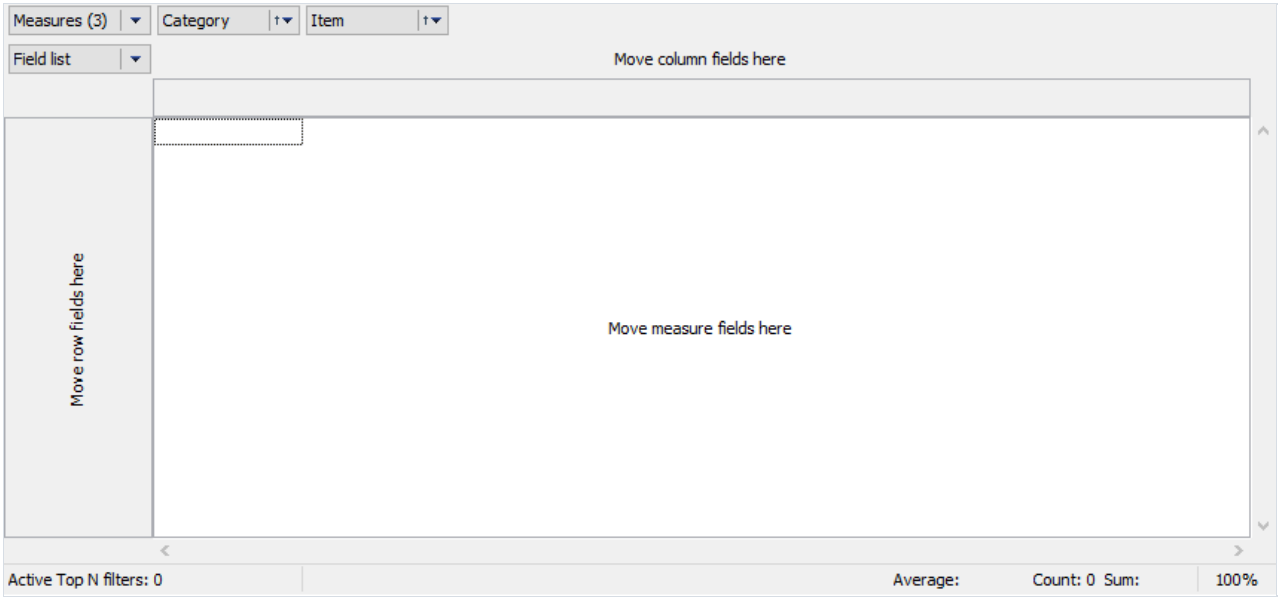
The grid structure many be defined in several different ways: by code, by loading from a cube or scheme file or by manual setup by the user. This paragraph deals only with manual setup of the grid structure.

Dimensions and measures are added to the grid by dragging from the drop-down list (shown below) or by using the Field List dialog. They are added to one of the following regions:

- the filters region
- the row fields region
- the column fields region
- or the measure fields region (the data region)



Let's add the fields "Category" and "Item" to the filters region, and drag the "Price" field to the data region. The grid now looks like this:

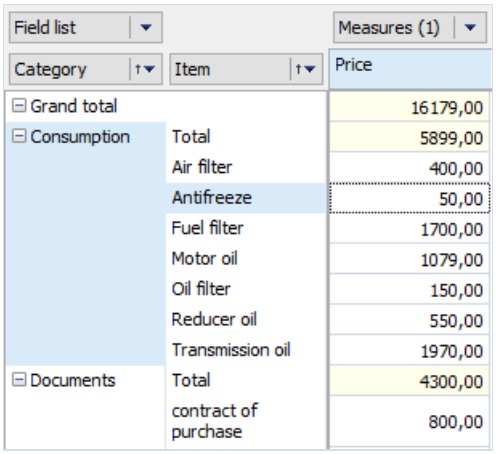


The number in brackets in the "Measures" dimension indicates the number of measures defined for this grid. Here the number of measures is 1, since we dragged the "Price" field to the data region. The "Measures" dimension is in fact not a simple dimension. Unlike all the others, "Measures" contains measures, and not elements of the database

row as its values. This implementation of the measure list as a pseudo-dimensions allows a consistent approach in the user interface - dragging to any dimension region and value filter.

No data is yet displayed in the grid above as the grid only displays data when it contains at least one measure and the "Measures" dimension is located either in the row fields region or in the column fields region.

For example, if the "Category" and "Item" dimensions are dragged into the row fields region and the "Measures" dimension into the column fields region, then the grid will look like this:

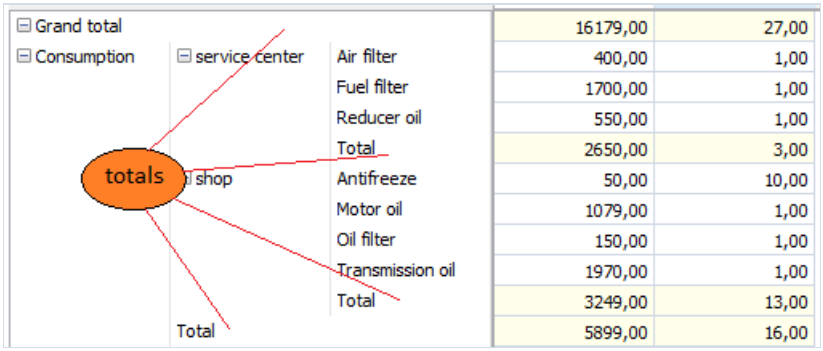


Field list		Measures (1)
Category	Item	Price
Grand total		16179,00
Consumption	Total	5899,00
	Air filter	400,00
	Antifreeze	50,00
	Fuel filter	1700,00
	Motor oil	1079,00
	Oil filter	150,00
	Reducer oil	550,00
	Transmission oil	1970,00
Documents	Total	4300,00
	contract of purchase	800,00

Totals

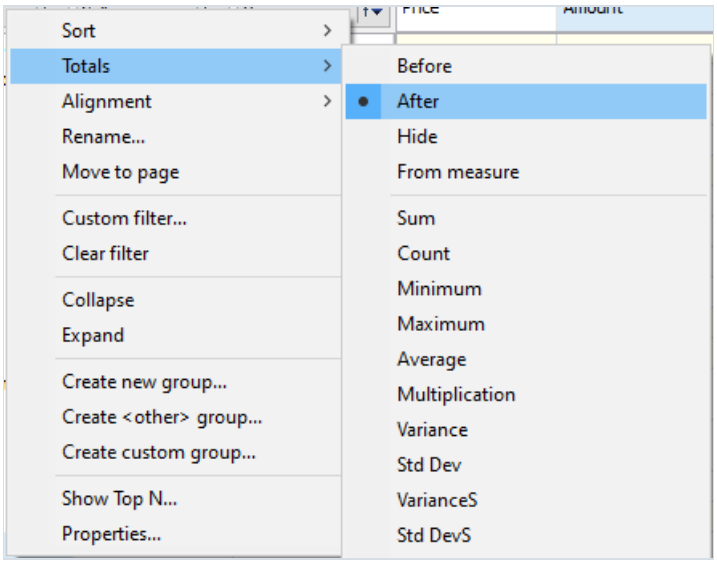
As well as dimension values the grid can also show Total values. Total value is an aggregated measure value over a group of dimension values. A Total value is calculated using the default aggregate function for the measure.

The various levels of Total are shown in this partial grid:



Grand total			16179,00	27,00
Consumption	service center	Air filter	400,00	1,00
		Fuel filter	1700,00	1,00
		Reducer oil	550,00	1,00
		Total	2650,00	3,00
	shop	Antifreeze	50,00	10,00
		Motor oil	1079,00	1,00
		Oil filter	150,00	1,00
		Transmission oil	1970,00	1,00
		Total	3249,00	13,00
	Total		5899,00	16,00

Use the dimension context menu to change the position of Totals (before or after), to hide Totals and to manage the aggregate functions.

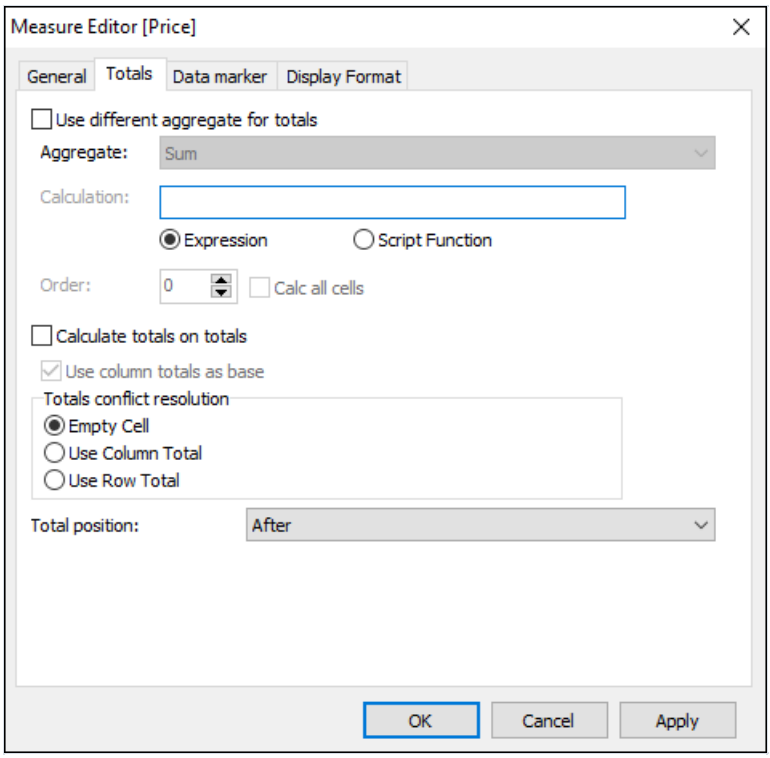


If in the example above we turn off the Total for the "Seller" dimension and also the Grand Total, set the Total position for "Category" dimension to "Before" and add an aggregate "Count" to it, we achieve this final grid:

Category	Seller	Item	Price	Amount	Work price
Consumption	Total		5899,00	16	1410,00
	Total Count		7,00	7	5,00
	service center	Air filter	400,00	1	70,00
		Fuel filter	1700,00	1	700,00
		Reducer oil	550,00	1	320,00
	shop	Antifreeze	50,00	10	
		Motor oil	1079,00	1	
		Oil filter	150,00	1	0,00
		Transmission oil	1970,00	1	320,00

The context menu item "From measure" sets Totals to show a value defined on the "Total position" parameter for the measure. This item only has a value for the axis which contains the "Measures" field and is ignored for other axes.

A measure's Total parameters can be modified in the measure editor:



A Total value is calculated using the default aggregate function for the measure. The aggregate function to be used can be changed or can even be implemented using a script, if required.

The Total calculation can be made either on the source dataset values or on total values of a child level. This can be set using the "Calc totals on to totals" checkbox. If this option is checked then the axis from which to take the Total value (either columns or rows) must be set. This can be set using "Use column totals as base".

Sometimes a situation can arise where, according to the dimensions parameters, each axis requires a different aggregate function with a different values for particular cells. So there is a conflict of which aggregate function to use. The "Totals conflict resolution" group-box has options for solving the conflict. A conflicting cell will not be calculated by default.

The "Total position" parameter defines how to show the Total for the axis with the "Measures" field for those dimensions which have the Totals position set to "From measure".

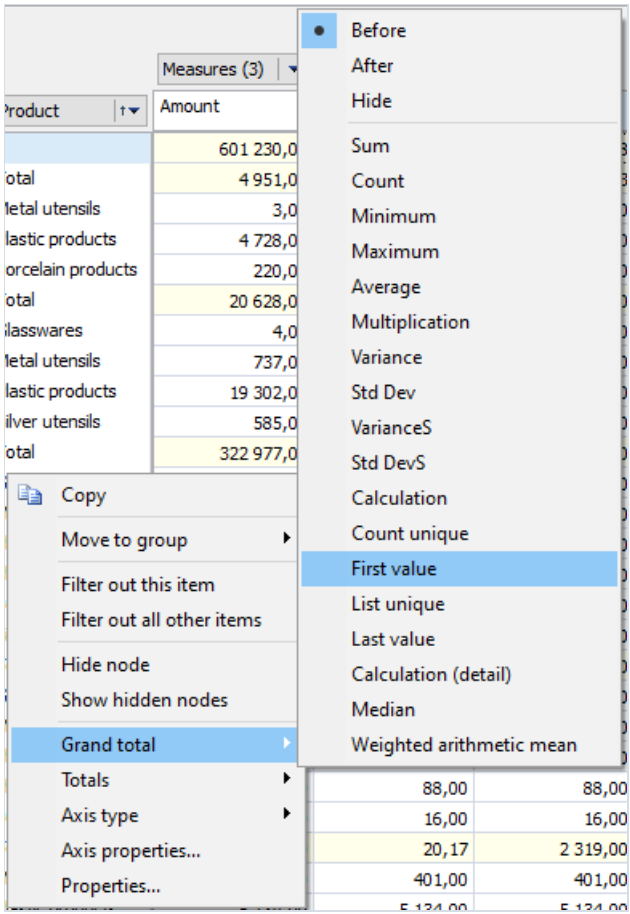
The example below shows the dimension "Manager" Totals set to "From measure", the "Amount" measure Totals set to "Before", the "Cost" measure Totals set to "Hide" and the other measures Totals set to "After". It can be seen that the "Cost" Total is not shown for the "Manager" dimension:

Manager	Product	Measures (4)	
Amount			601 230,00
Cost			11 706 187,56
Paid			5 219 178,52
Balance			6 487 009,04
Baldwin Janet	Amount		4 951,00
	Metal utensils	Amount	3,00
		Cost	643,00
		Paid	643,00
		Balance	0,00
	Plastic products	Amount	4 728,00
		Cost	240 260,35
		Paid	6 453,60
		Balance	233 806,75
	Porcelain products	Amount	220,00
		Cost	6 416,30
		Paid	1 257,60
		Balance	5 158,70

The next example shows the outcome when Total aggregate functions are changed and Total values are calculated on child totals. All three measures are built using the main aggregate function "Sum" by field "Amount". The Total for measure "Amount" is not customized. For measures "Amount1" and "Amount2" the Totals aggregate function is changed to "Average" and the measure "Amount2" uses the "Calculate totals on totals" option:

Field list		Measures (3)		
Manager	Product	Amount	Amount 2	Amount 3
Grand total		601 230,00	186,83	11 619,28
Baldwin Janet	Total	4 951,00	29,65	1 650,33
	Metal utensils	3,00	3,00	3,00
	Plastic products	4 728,00	4 728,00	4 728,00
	Porcelain products	220,00	220,00	220,00
Burbank Kevin	Total	20 628,00	45,24	5 157,00
	Glasswares	4,00	4,00	4,00
	Metal utensils	737,00	737,00	737,00
	Plastic products	19 302,00	19 302,00	19 302,00
	Silver utensils	585,00	585,00	585,00
Glon Jacques	Total	322 977,00	1 845,58	53 829,50
	Glasswares	3 087,00	3 087,00	3 087,00
	Metal utensils	119,00	119,00	119,00
	Plastic box	289 740,00	289 740,00	289 740,00
	Plastic products	1 316,00	1 316,00	1 316,00
	Porcelain products	28 711,00	28 711,00	28 711,00
	Silver utensils	4,00	4,00	4,00
Guckenheimer Scott Jr	Total	59 936,00	922,09	11 987,20
	Glasswares	1 400,00	1 400,00	1 400,00
	Metal utensils	82,00	82,00	82,00
	Plastic box	58 350,00	58 350,00	58 350,00
	Plastic products	88,00	88,00	88,00

Grand totals can only be managed through the context menu:

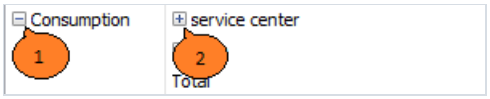


Drill Up/Down

Grid data can be collapsed (drilled up) to exclude the values of the dimensions located below from processing or can be expanded (drilled down). These operations are performed using the "-" and "+" buttons located on both headers of the grid.

To group the data by the value of a certain dimension, you need to click the "-" button of the corresponding dimension. And conversely, to refine the data click the "+" button. These buttons are identified as '1' and '2' below:

1. drill up button
2. drill down button



Drilling up or drilling down changes the structure of the grid header and the data region. The two illustrations below show how the grid looks after drilling down on "service center" and then after drilling up on "Consumption":

⊟ Consumption	⊟ service center	Air filter	400,00	1,00	70,00
		Fuel filter	1700,00	1,00	700,00
		Reducer oil	550,00	1,00	320,00
		Total	2650,00	3,00	1090,00
	⊞ shop		3249,00	13,00	320,00
	Total		5899,00	16,00	1410,00

⊞ Consumption	5899,00	16,00	1410,00

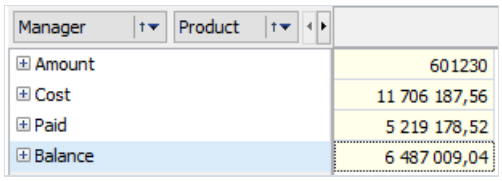
Only the main Total shows when a dimension item is collapsed.

It is possible to collapse the grand totals. The result depends on whether or not the "Measures" field is placed in the collapsed axis. An axis without measures collapses with the hiding of all the cells except the grand total cell. An axis with measures can collapse grand totals for each measure independently. The collapse of measure grand totals hides all the cells of that measure. Collapsing the grand totals of all the measures hides all the cells except the grand total cells for each measure.

In this example the grand totals for measure "Amount" and "Cost" are collapsed:

⊞ Amount			601230
⊞ Cost			11 706 187,56
⊟ Paid			5 219 178,52
⊟ Balance			6 487 009,04
⊟ Baldwin Janet	Paid		8 354,20
	Balance		238 965,45
	Metal utensils	Paid	643,00
		Balance	0,00
	Plastic products	Paid	6 453,60
		Balance	233 806,75
	Porcelain products	Paid	1 257,60
		Balance	5 158,70
⊟ Burbank Kevin	Paid		536 548,98
	Balance		885 426,09
	Glasswares	Paid	483,25
		Balance	0,00
	Metal utensils	Paid	132 447,40
		Balance	79 728,10

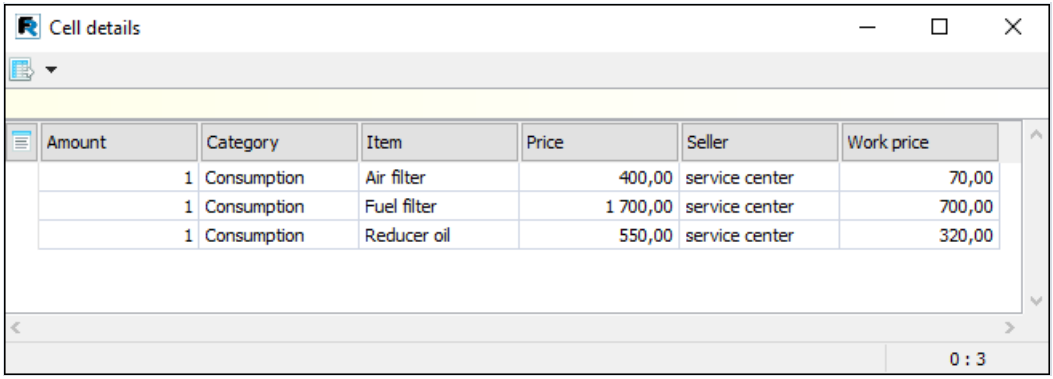
And here the grand totals for all the measures are collapsed:



Manager	Product	
⊞ Amount		601230
⊞ Cost		11 706 187,56
⊞ Paid		5 219 178,52
⊞ Balance		6 487 009,04

Drill Through

Double-clicking on a data cell opens a window showing a Detail Table which contains the source data rows used for calculation of the selected cell:



This Detail Table can be manipulated by:

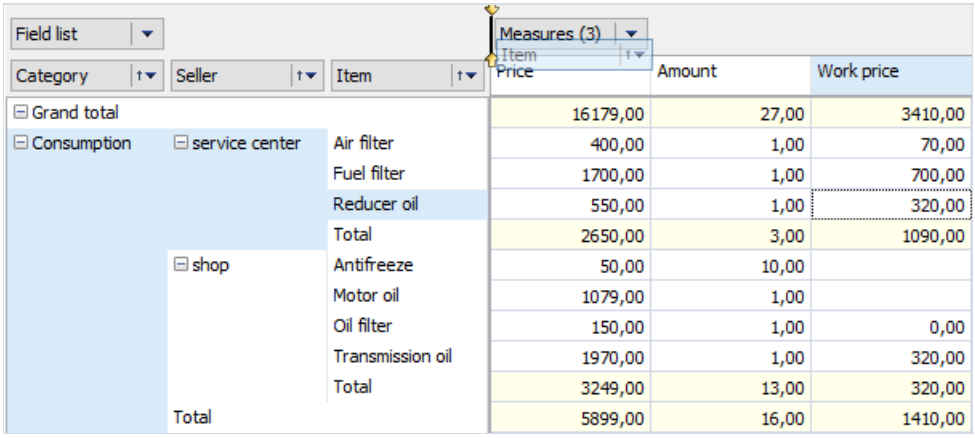
- changing the visibility and size of the columns
- selecting data cells for copying to the clipboard
- exporting data to the same format as the main grid

The status bar shows the current row and total row count.

Rotation

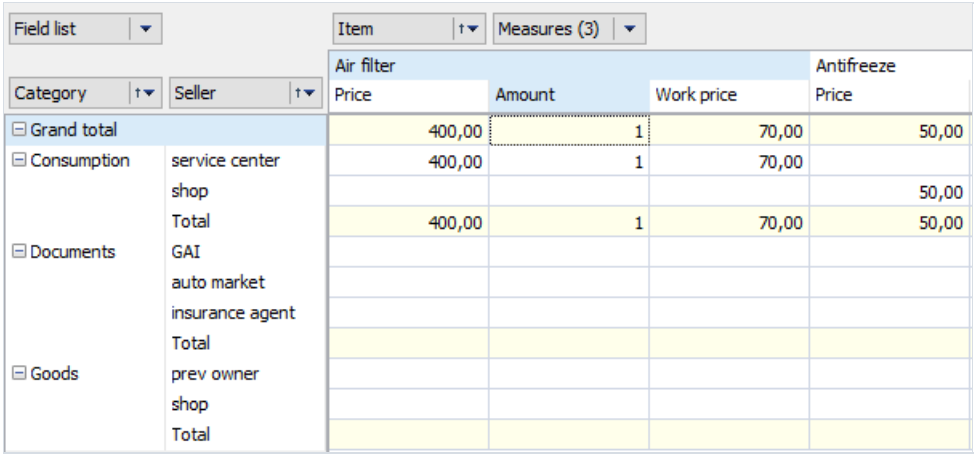
The grid layout can easily be changed by moving the dimensions between the three regions: filter region, row dimensions region and column dimensions region. In OLAP terms, this operation is called rotation, because it corresponds to rotating a multi-dimensional data array. Data rotation enables the same information to be analysed from different perspectives.

To rotate data in the grid the various dimensions must be dragged with the mouse from one region to another. As the mouse is dragged over the grid a special pointer shows where the dimension will go if the button mouse is released at that moment (see grid below):



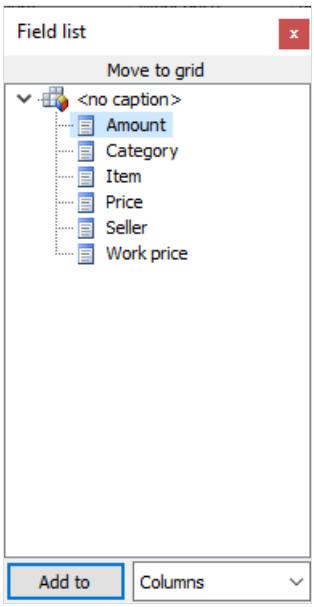
Field list			Measures (3)		
Category	Seller	Item	Price	Amount	Work price
Grand total			16179,00	27,00	3410,00
Consumption	service center	Air filter	400,00	1,00	70,00
		Fuel filter	1700,00	1,00	700,00
		Reducer oil	550,00	1,00	320,00
		Total	2650,00	3,00	1090,00
	shop	Antifreeze	50,00	10,00	
		Motor oil	1079,00	1,00	
		Oil filter	150,00	1,00	0,00
		Transmission oil	1970,00	1,00	320,00
		Total	3249,00	13,00	320,00
	Total		5899,00	16,00	1410,00

This is the resultant grid after the "Item" dimension has been moved from the row dimensions region to the column dimensions region:



Field list		Item	Measures (3)		
		Air filter			Antifreeze
Category	Seller	Price	Amount	Work price	Price
Grand total		400,00	1	70,00	50,00
Consumption	service center	400,00	1	70,00	
	shop				50,00
	Total	400,00	1	70,00	50,00
Documents	GAI				
	auto market				
	insurance agent				
	Total				
Goods	prev owner				
	shop				
	Total				

The move can also be made using the field list dialog of the grid:



As well as partial rotation it is also possible to perform total rotation (grid transposition). This operation moves all the row dimensions to the column region and all the column dimensions to the row region. Unlike partial rotation, grid transposition does not require the recalculation of data cells and as a result is instantaneous. Transpose a grid using the 'Transpose' button on the toolbar. Our transposed grid looks like this:

Move filter fields here

Field list / Measures (3)	Category: Grand total	Seller: Consumption / service center: Air filter	Item: Fuel filter	Item: Reducer oil	Item: Total	Seller: shop: Antifreeze
Price	16179,00	400,00	1700,00	550,00	2650,00	50,00
Amount	27,00	1,00	1,00	1,00	3,00	10,00
Work price	3410,00	70,00	700,00	320,00	1090,00	

Filtering

Data filtering is performed by excluding certain dimension values from the analysis. Please note that it does not matter in which region of the grid this dimension is located (filter region, row dimensions region or column dimensions region). Filtered values are not only hidden in the grid header but are also excluded from measures and Totals calculations.

Values of dimensions can be filtered using the drop-down list which is opened with a mouse-click on the dimension filtering button ('1' below):



Key to the grid above:

1. dimension filtering button
2. list of dimension values
3. drop-down list toolbar

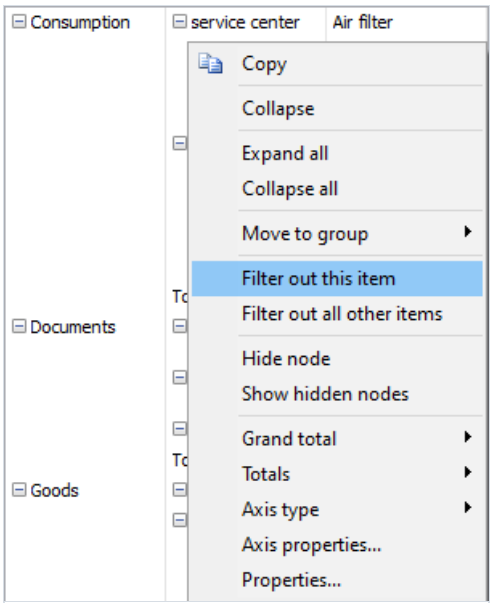
Checked values in the drop-down list are visible in the grid. If un-checked, all the corresponding rows of the source data are excluded from grid calculations. Values are toggled either by clicking on its checkbox or by selecting the value and pressing the spacebar on the keyboard. Holding the Ctrl button down while clicking a checkbox turns off all the values except the clicked one.

Depending on the grid settings, changes may be applied automatically or by the Apply button.

The toolbar may contain the following commands:

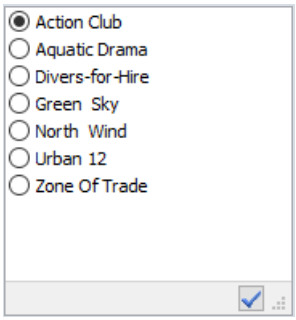
Icon	Command
	Mark all values as visible
	Mark all values as hidden
	Toggle visible / hidden
	Show only the values that are not filtered by other dimensions
	Apply changes and close list
	Cancel changes and close list

Filters can also be applied through the grid context menu:

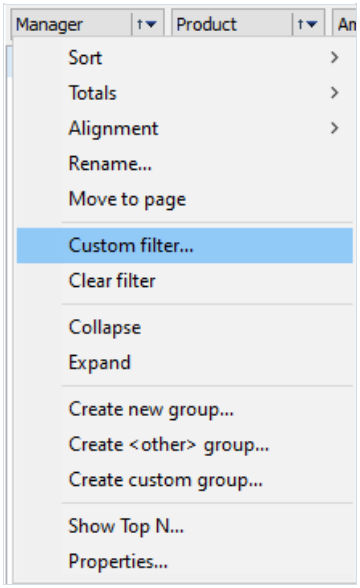


The Filter drop-down list can also be configured as radio buttons instead of checkboxes.

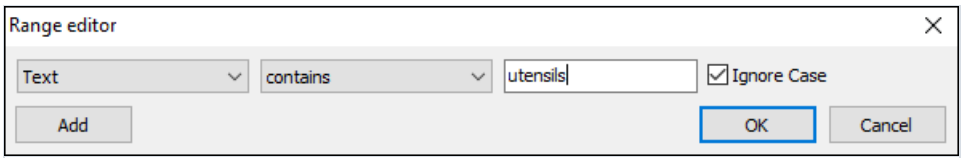
Radio buttons allow the selection of only one value at a time as visible:



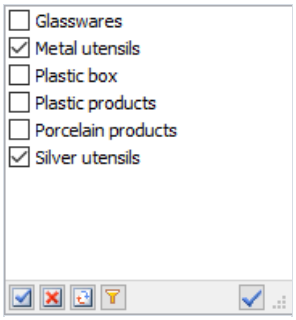
The Dimension values Range Editor offers conditional filtering of values. It is opened from the dimension context menu item "Custom filter...":



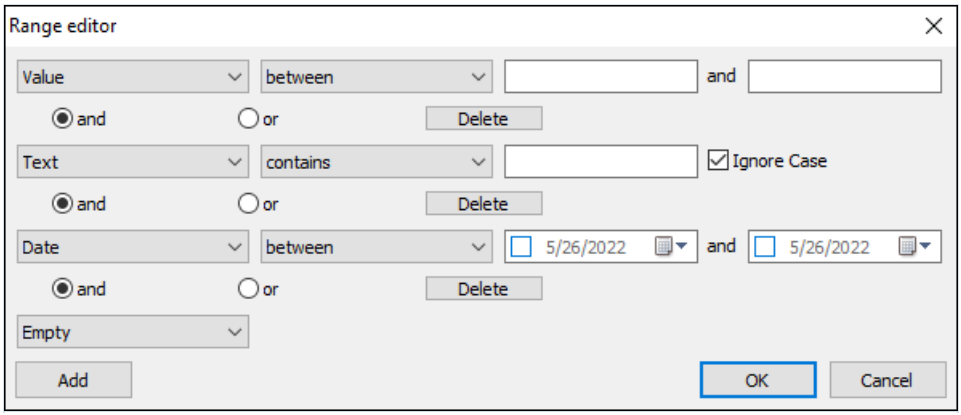
For example, the "Product" dimension can be set to filter out all values except where "Text contains 'utensils' substring":



This is equivalent to:



The Range Editor can create multiple, simultaneous conditions of varying type:



Filtering affects measures and the calculation of Totals, but the grid can also be set to simply hide particular nodes in the headers. In this case the hidden nodes do not affect measures and the calculation of Totals. Node hiding can only be set using the context menu.

Sorting

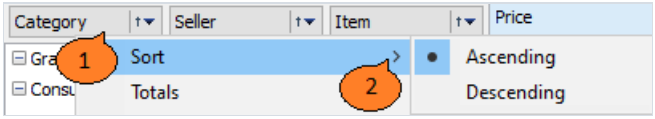
All the data in the grid is displayed in sorted order, irrespective of the original order in the source table. The sort order is either ascending or descending. Numeric values and date/times are sorted in their natural order, ascending or descending. Strings are sorted alphabetically, ascending or descending.

Every dimension in the grid can have its own sort order. For example, rows for the "Category" dimension can have ascending order and those for the "Seller" dimension descending order. The default sort order is ascending for all dimensions.

The sort order for a selected dimension is changed either by:

1. double-clicking on the dimension title
2. or selecting the required order from the dimension context menu (numbered '2' in diagram below)

The current dimension sort order is marked with a special symbol (numbered '1' in diagram below):



Apart from the sort order, it is also possible to set the sort type separately for lines and columns.There are three types of sorting:

1. by axes values : the default sort type
2. by measure Totals : data is sorted by the Total values for a measure - if there are several measures then only the values of the measure associated with the selected cell are included in the sort
3. by active row/column : sorting is similar to the previous type, but instead of the Total values of the row/column the values of the current row/column (selected in the grid) are used

The sort type is selected through the toolbar for both row dimensions and column dimensions .

Grouping

FastCube can collect several dimension values into a group. Grouping is a two stage process - group creation followed by population with values.

Grouping features:

- unlimited number of dimension groups
- empty groups allowed
- empty groups not shown on axes
- groups must have unique names inside a dimension
- a dimension value cannot belong to more than one group
- inclusion of a dimension value in one group automatically excludes it from all other groups
- a dimension value can be excluded from grouping
- a system group "Others" can be created, which includes all values not belonging to any other group
- group deletion automatically excludes all the values it contains
- group creation, deletion, renaming, value inclusion and exclusion can be performed while the cube is active
- a filter window shows all groups and their members : group filtering state depends on member states
- moving a dimension from one region to another does not reset groups
- dimension groups are saved in cube files together with the dimension members

The axis shows dimensions having groups at two levels: group level and member level. The group level contains dimension group names and also dimension values that do not belong to any group. The member level contains the values which belong to the groups. A group can be in a collapsed state, in which case the group members are not shown. If all groups are collapsed then the member level is not shown.

Beware! any operation involving a group, including collapse/expansion, causes measure recalculation because of the change to the axis.

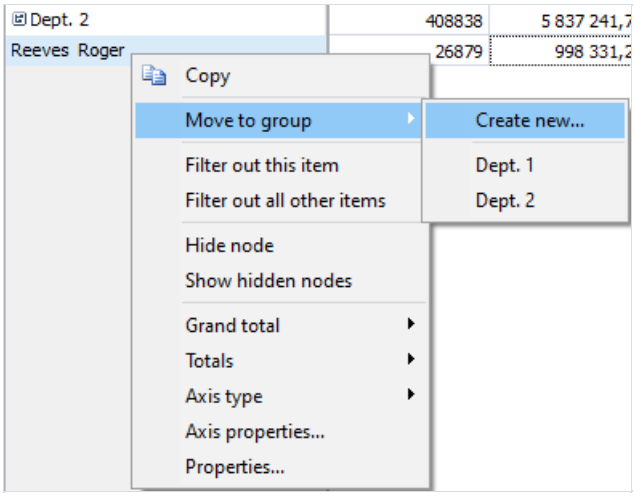
An example of a grid where the "Manager" dimension has no grouping:

Product	Customer			
Field list	Measures (4)			
Manager	Amount	Cost	Paid	Balance
⊟ Grand total	601230	11 706 187,56	5 219 178,52	6 487 009,04
Baldwin Janet	4951	247 319,65	8 354,20	238 965,45
Burbank Kevin	20628	1 421 975,07	536 548,98	885 426,09
Glon Jacques	322977	3 422 430,60	71 131,63	3 351 298,97
Guckenheimer Scott Jr	59936	724 337,69	21 410,20	702 927,49
Montgomery Jennifer	6957	364 227,36	1 592,20	362 635,16
Nordstrom Carol	314	15 550,20	15 503,20	47,00
Parker Robert	10490	218 139,22	188 610,63	29 528,59
Phong Leslie	68100	1 816 894,39	1 574 939,34	241 955,05
Reeves Roger	26879	998 331,20	843 409,90	154 921,30
Steadman Walter	77085	2 293 621,88	1 903 866,24	389 755,64
Sutherland Claudia	2913	183 360,30	53 812,00	129 548,30

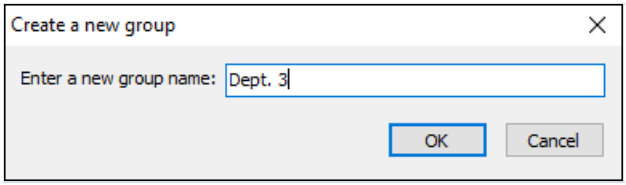
The same grid with the dimension "Manager" having two groups, "Dept.1" and "Dept.2", each manager having been moved to the correct Department group:

Manager		Amount	Cost	Paid	Balance
Grand total		601230	11 706 187,56	5 219 178,52	6 487 009,04
Dept. 1	Total	165513	4 870 614,59	2 523 991,62	2 346 622,97
	Baldwin Janet	4951	247 319,65	8 354,20	238 965,45
	Burbank Kevin	20628	1 421 975,07	536 548,98	885 426,09
	Guckenheimer Scott Jr	59936	724 337,69	21 410,20	702 927,49
	Steadman Walter	77085	2 293 621,88	1 903 866,24	389 755,64
	Sutherland Claudia	2913	183 360,30	53 812,00	129 548,30
Dept. 2		408838	5 837 241,77	1 851 777,00	3 985 464,77
Reeves Roger		26879	998 331,20	843 409,90	154 921,30

Note that group "Dept.1" is expanded and group "Dept.2" is collapsed. Also, manager "Reeves Roger" does not belong to any group since she works in "Dept.3". Let's create a group for her. First open the context menu by right-clicking on value "Reeves Roger". The "Move to group" options are : add to a previously created group, or start a new group. Let's choose "Create new...":



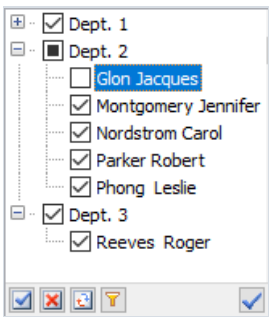
And enter the department name:



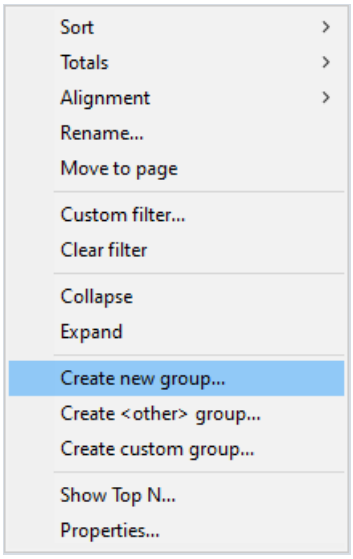
The new group is created and "Reeves Roger" is moved to it:

Manager	Amount	Cost	Paid	Balance
Grand total	601230	11 706 187,56	5 219 178,52	6 487 009,04
Dept. 1	165513	4 870 614,59	2 523 991,62	2 346 622,97
Dept. 2	408838	5 837 241,77	1 851 777,00	3 985 464,77
Dept. 3	26879	998 331,20	843 409,90	154 921,30

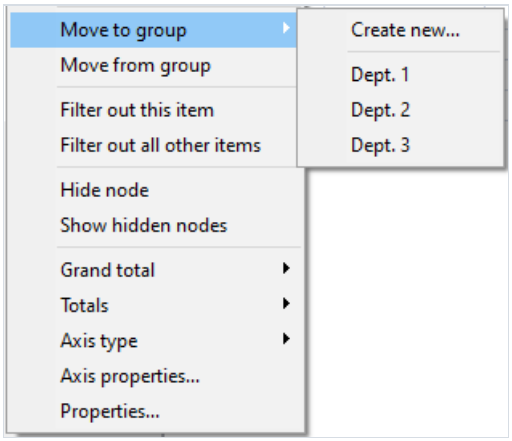
The filter drop-down list for the "Manager" dimension now looks like this:



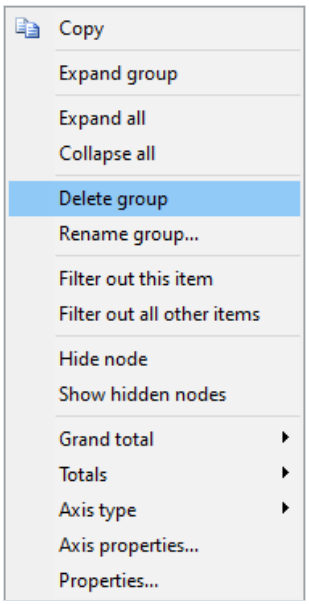
And the context menu for the dimension is:



While the context menu for a dimension member is:



And the context menu for the dimension group is:

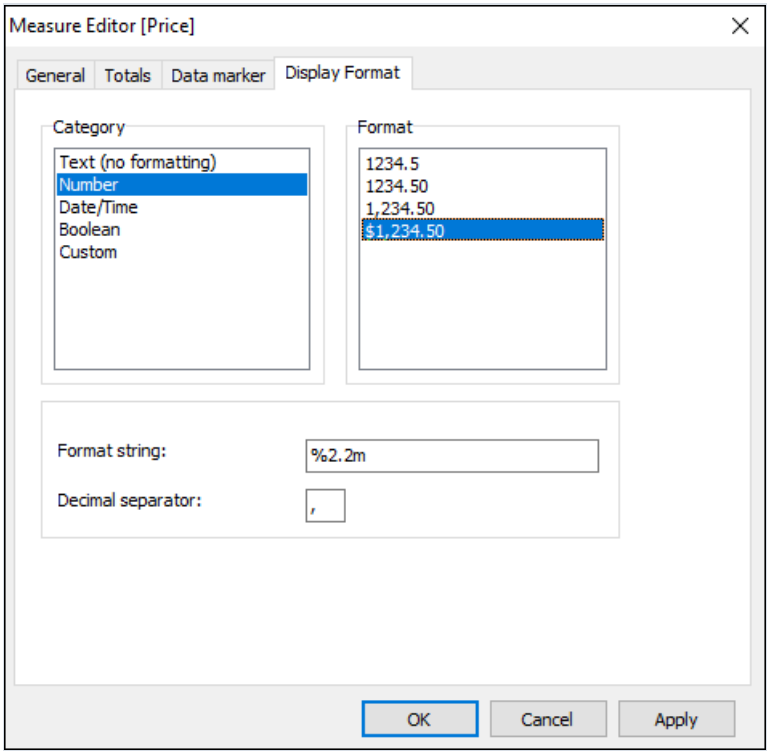


Data Formatting

By default a grid formats data according to the field type and the precision. For numeric measures the grid uses a number format having the same precision as the database field. But sometimes one or more measures need to be presented differently. For example, let's change the format of the "Price" measure to currency:

Category	Seller	Item	Price	Amount	Work price
Grand total			16179,00	27	3410,00
Consumption	service center	Air filter	400,00	1	70,00
		Fuel filter	1700,00	1	700,00
		Reducer oil	550,00	1	320,00
		Total	2650,00	3	1090,00

The Format Editor is opened either from the cell's context menu or through the toolbar:



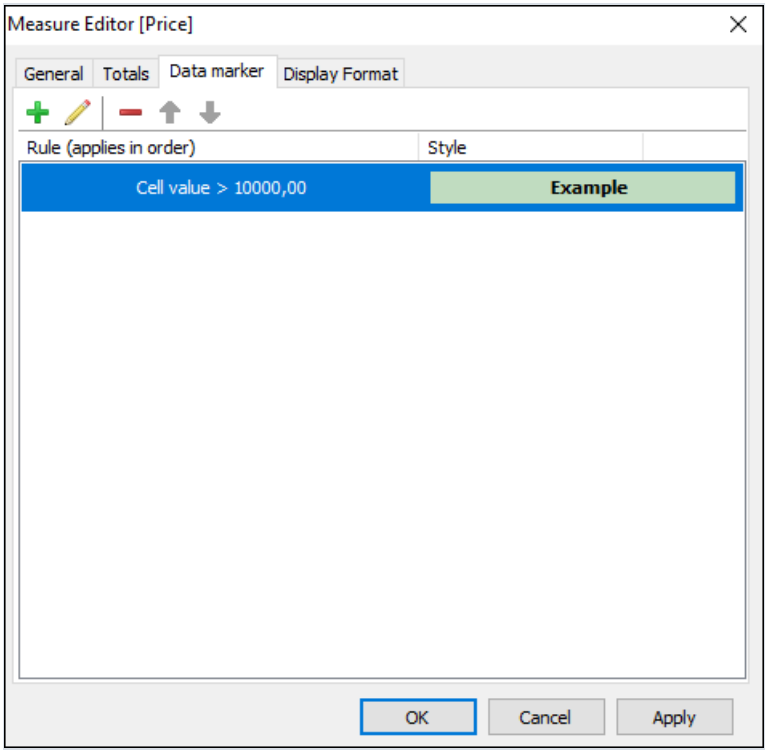
This editor lists the format categories on the left with the corresponding formats on the right. The format string and decimal separator for the selected category and format are at the bottom. Let's select the "Number" category and "$1,234.50" format. The format string is an argument for the Delphi "Format" function, which is used by FastCube to implement number formatting. The format string and decimal separator can be altered if necessary. If the decimal separator is left blank then the value from the current regional setup is used:

Grand total			$16,179,00	27	3410,00
Consumption	service center	Air filter	$400,00	1	70,00
		Fuel filter	$1,700,00	1	700,00
		Reducer oil	$550,00	1	320,00
		Total	$2,650,00	3	1090,00

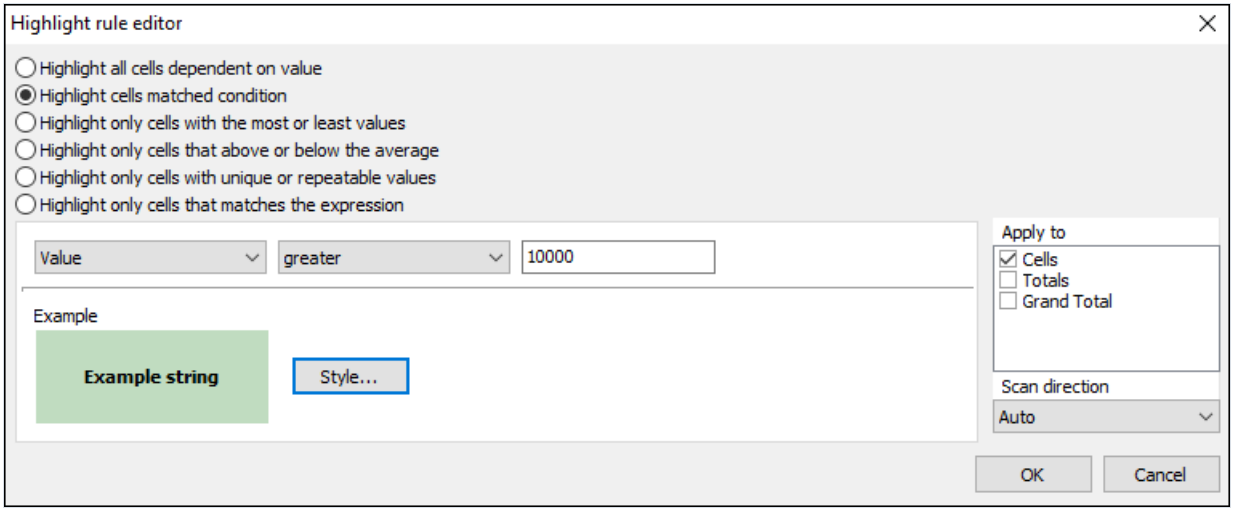
Conditional highlighting

The grid allows custom drawing styles to be set up for particular measure values. For example, numbers inside or outside a particular range may need to be highlighted.

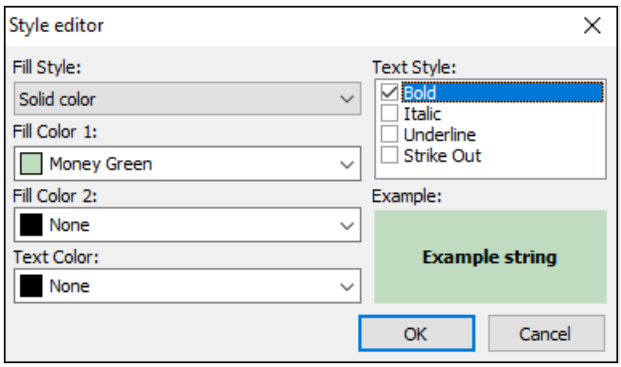
Before highlighting measure values a new highlight rule must be created in the measure editor dialog:



Highlight rules can process measure data as value, text or data, or as NULL. Allowable conditions depend on the type of the measure's data. So, for example, value processing allows "greater" and "lower" conditions while text processing allows "contains" and "not contains" conditions, which look for a substring within a text string:

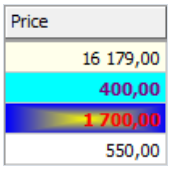


The "Style..." button opens the drawing style editor:



The style editor has options for the background fill style and color, and text style and color. Solid fill style only requires a "Fill Color 1" value, but gradient fills also require a "Fill Color 2" value.

Here is an example after applying different conditional styles to some measure cells:

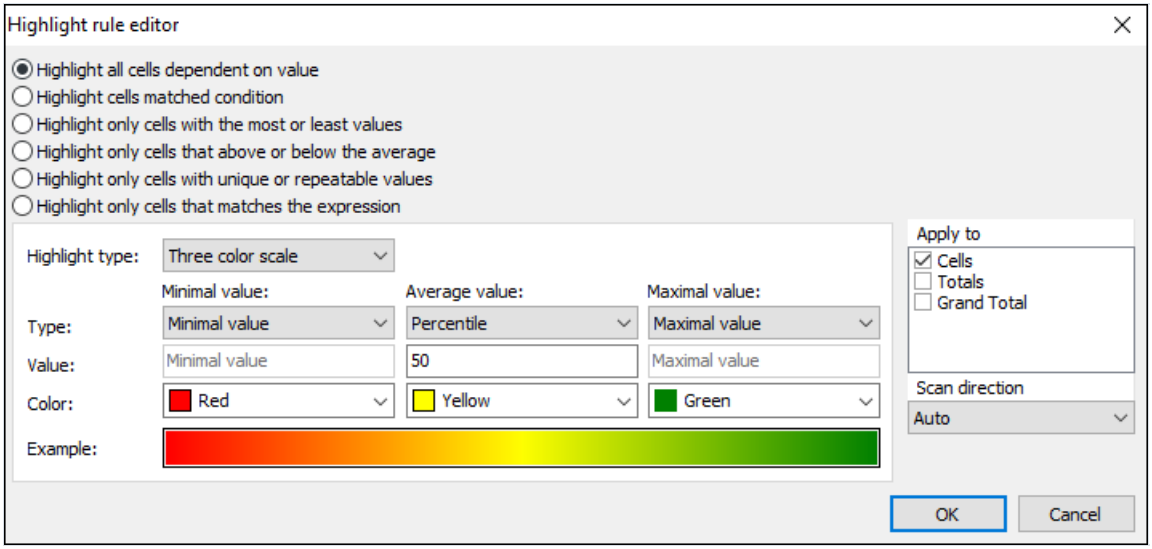


Continuous highlighting

As well as the conditional cell highlighting described above, FastCube can also apply four different types of highlight known as continuous highlighting, which applies highlighting to all cells dependent on their values. Each type of continuous highlighting is described below.

1. Two color scale and

2. Three color scale



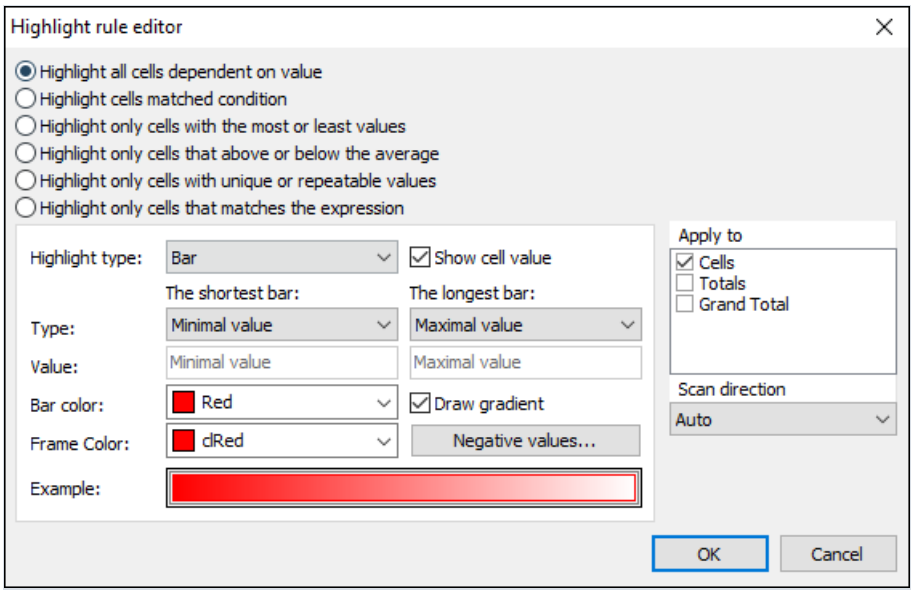
Color scale highlighting fills each cell's background with a color calculated from the given color gradient scale. The editor dialog sets the values for the extreme points and their corresponding colors, and also for one intermediate point for the three color scale. The values for the points can be set either as absolute numbers ("Number" option) or as one of the following relative options:

- Minimum by row/column - minimal measure value in row/column
- Percent by row/column - value field sets the percent relative to minimal and maximal measure values in row/column
- Percentile by row/column - value field sets the percentile relative to measure values in row/column

An example of two color and three color scale highlighting:

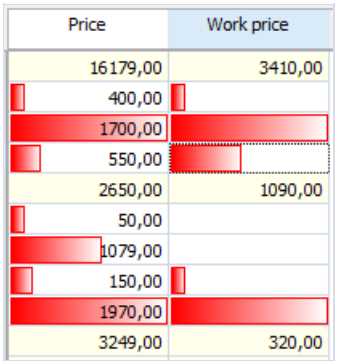
Price	Work price
16179,00	3410,00
400,00	70,00
1700,00	700,00
550,00	320,00
2650,00	1090,00
50,00	
1079,00	
150,00	0,00
1970,00	320,00

3. Bar

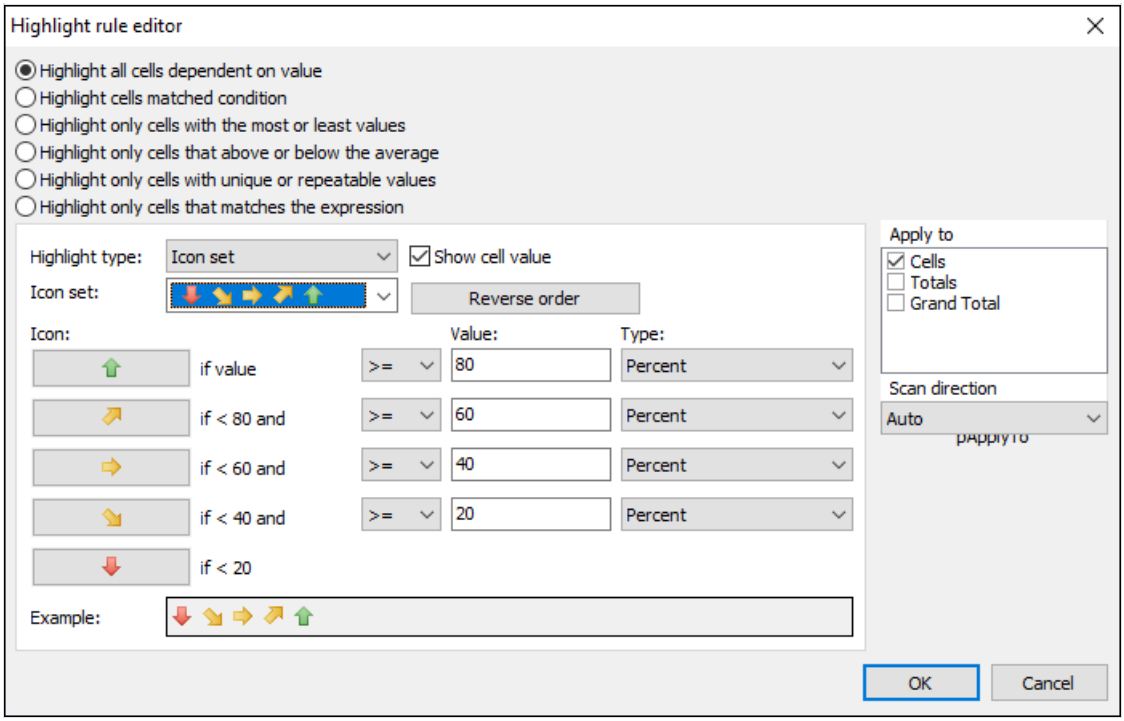


This type of highlighting draws a colored bar within the measure cell. The length of the bar depends on the values set under the Shortest bar and the Longest bar properties. The property values, as previously, can be set with either the absolute or the relative option. The bar can be drawn either in a solid color or with a gradient color to white. The frame color for the bar is set independently. The cell value can also be hidden for this type of highlighting, leaving only the bar visible.

An example of bar highlighting with and without visible cell values:

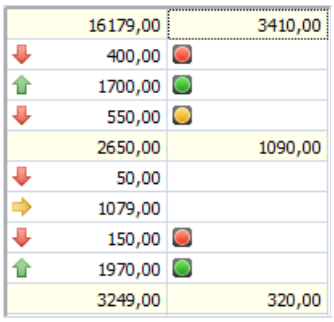


4. Icon set



In Icon set highlighting an icon related to the value inside the measure cell is drawn to the left of the cell's value. The editor dialog has options for sets of 3, 4 or 5 icons. Each icon used requires a value range to be set. The cell value can also be hidden for this type of highlighting, leaving only the icon visible.

An example of icon set highlighting with and without visible cell values:



FastScript use

FastCube can use FastScipt to perform non-standard calculations.

Scripts can be used for both measure and filter calculations.

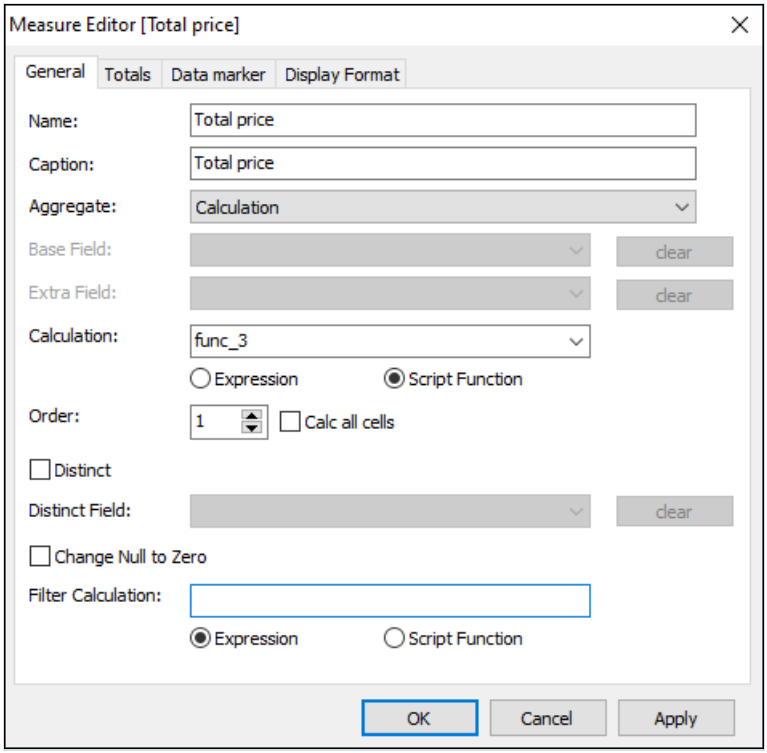
Calculated measures

To create a calculated measure we need to set the aggregate type as "Calculation" or as "Calculation (detail)" in the measure editor.

The two types of calculated measure are:

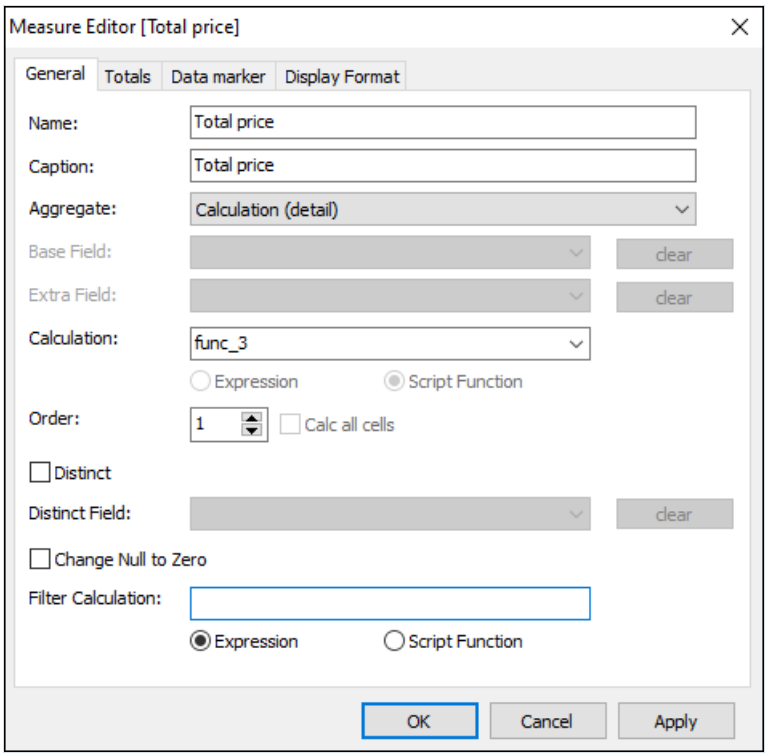
- Calculation based on other measure values ("Calculation")

Measures which are calculated after other measures have been built using the standard measure processing. The calculation function has access to other measure values, dimension values and previous measure values. The "Calc all cells" checkbox turns on calculation for all cells, including "empty" cells (cells which do not have corresponding rows in the source data).



- Calculation based on source data ("Calculation (detail)")

Measures which are calculated on source data and built together with standard measures. The calculation function has access to the current source data row and to dimension values. The "Calc all cells" checkbox is disabled for this type of calculation.



The use of either type of calculated measure requires either a new calculation function to be written or an existing function to be chosen in the "Function" field - double-clicking on the "Function" field opens the measure editor. If the "Function" field is empty then a new function is automatically created.

The "Order" field sets the calculation order.

The measure editor window has three areas:

1. toolbar
2. text editor with syntax highlighting
3. treeview with available variables, functions and classes

The toolbar contains the actions for checking syntax, saving changes, cancelling changes and choice of programming language.

The text editor is used to compose and edit the function code (a 'script'). The function code is a program written in the chosen language that is interpreted by FastScript at calculation time. The function code must return a value through the "Result" variable.

The script has access to the following grid elements:

- dimension list - Dimensions : TDimensions;
- measure list - Measures : TMeasures;
- field list - SliceFields : TfcxSliceFields;
- service variable - CustomObject: TfcxCustomObject.

The service variable enables the reading/writing of user data for a particular measure cell. For example, a pointer for future use can be assigned to any object created in the calculation function. Each cell has its own service variable. The CustomObject can only be used in measures built on source data. If the CustomObject variable contains a pointer to an allocated object or memory area then it must be deallocated after use. For this purpose a 'Final: boolean' argument is passed to the function body. If 'Final' = 'True' then the finalization procedure is activated, which includes memory deallocation.

The following classes are available for use in calculated measure functions:

// Measure list

```
TMeasures = class
published
  property Items[AIndex: Integer]: TMeasure;
  property ItemByCaption[AIndex: String]: TMeasure;
  property ItemByName[AIndex: String]: TMeasure;
  property Count: Integer;
  property RecordCount: integer;
  property DetailValue[ARecordIndex: Integer; AFieldName: String]: Variant;
  property XLevel: integer;
  property YLevel: integer;
end;
```

// Dimension list

```
TDimensions = class
published
  property Items[AIndex: Integer]: TDimension;
  property ItemByCaption[AIndex: String]: TDimension;
  property ItemByName[AIndex: String]: TDimension;
  property XAxisItems[AIndex: Integer]: TDimension;
  property YAxisItems[AIndex: Integer]: TDimension;
  property IsTotalByCol: boolean
  property IsTotalByRow: boolean
  property XLevel: integer;
  property YLevel: integer;
  property XAxisLevelsCount: integer;
  property YAxisLevelsCount: integer;
end;
```

// Field list

```
TfcxSliceFields = class
published
    property Items[Index: Integer]: TfcxSliceField;
    property ItemByCaption[AIndex: String]: TfcxSliceField;
    property ItemByName[AIndex: String]: TfcxSliceField;
    property Count;
end;
```

// Measure

```
TMeasure = class
published
  property Caption: String;
  property CurrentValue: Variant;
  property CurrentCaption: String;
  property FieldName: String;
  property ColOffsetValue[Offset: integer]: Variant;
  property RowOffsetValue[Offset: integer]: Variant;
  property ColOffsetValueWithDimValue[ADimValue: Variant]: Variant;
  property RowOffsetValueWithDimValue[ADimValue: Variant]: Variant;
  property TotalValueForDims[ADimNames: String]: Variant;
  property ColRowOffsetValue[ColOffset, RowOffset: integer]: Variant;
  property ColRowOffsetValueWLevel[ColOffset, RowOffset, ColLevelOffset, RowLevelOffset: integer]:
Variant
end;
```

// Dimension

```
TDimension = class
published
  property FieldName: String;
  property Caption: String;
  property CurrentValue: Variant;
  property CurrentCaption: String;
  property SubGroup: TDimension;
end;
```

// Field

```
TfcxSliceField = class(TPersistent)
published
  property FieldName: String;
  property Caption: String;
  property CurrentValue: Variant;
  property CurrentCaption: String;
  property FilterCount: Integer;
  property IsFiltered: Boolean;
end;
```

// Service object

```
TfcxCustomObject = class(TPersistent)
published
  property Value: Pointer;
  property ValueIsNil: Boolean;
end;
```

Here are detailed descriptions of the methods and properties of the above listed classes:

TMeasure

- Caption - measure caption
- FieldName - measure field name
- CurrentValue - current value
- CurrentCaption - current value caption
- ColOffsetValue[Offset: integer] - measure value from cell with column offset from the current value
- RowOffsetValue[Offset: integer] - measure value from cell with row offset from the current value
- ColOffsetValueWithDimValue[ADimValue: Variant] - measure value corresponding to current dimension ADimValue value of column dimensions
- RowOffsetValueWithDimValue[ADimValue: Variant] - measure value corresponding to current dimension ADimValue value of row dimensions
- TotalValueForDims[ADimNames: String] - value of total over requested dimensions (DimNames separated by ',')
- ColRowOffsetValue[ColOffset, RowOffset: integer] - measure value from cell with column and row offset from the current value
- ColRowOffsetValueWLevel[ColOffset, RowOffset, ColLevelOffset, RowLevelOffset: integer] - measure value from cell with column and row offset from the value on requested level

TDimension

- Caption - dimension caption
- FieldName - dimension field name
- CurrentValue - current value
- CurrentCaption - current value caption
- SubGroup - dimension subgroup

TfcxSliceField

- Caption - field caption
- FieldName - field name
- CurrentValue - current value (available only for filter script)
- CurrentCaption - current caption (available only for filter script)
- FilterCount - number of filtered values
- IsFiltered - true when field values are filtered, otherwise false

TMeasures

- Items[AIndex: Integer] - measure array with access by index
- ItemByCaption[AIndex: String] - measure array with access by caption
- ItemByName[AIndex: String] - measure array with access by name (default property)
- Count - number of measures
- RecordCount - number of detail rows
- DetailValue[ARecordIndex: Integer; AFieldName: String] - field value in the detail row
- XLevel - Level of X axis to which the current cell belongs
- YLevel - Level of Y axis to which the current cell belongs

TDimensions

- Items[AIndex: Integer] - dimension array with access by index
- ItemByCaption[AIndex: String] - dimension array with access by caption
- ItemByName[AIndex: String] - dimension array with access by name (default property)
- Count - number of dimensions
- XAxisItems[AIndex: Integer] - dimension in X axis
- YAxisItems[AIndex: Integer] - dimension in Y axis
- IsTotalByCol - true when cell is a total cell by X axis, otherwise false
- IsTotalByRow - true when cell is a total cell by Y axis, otherwise false
- XAxisLevelsCount - number of dimensions in X axis
- YAxisLevelsCount - number of dimensions in Y axis
- XLevel - Level of X axis to which the current cell belongs
- YLevel - Level of Y axis to which the current cell belongs

TfcxSliceFields

- Items[Index: Integer] - field array with access by index
- ItemByCaption[AIndex: String] - field array with access by caption
- ItemByName[AIndex: String] - field array with access by name (default property)
- Count - number of fields

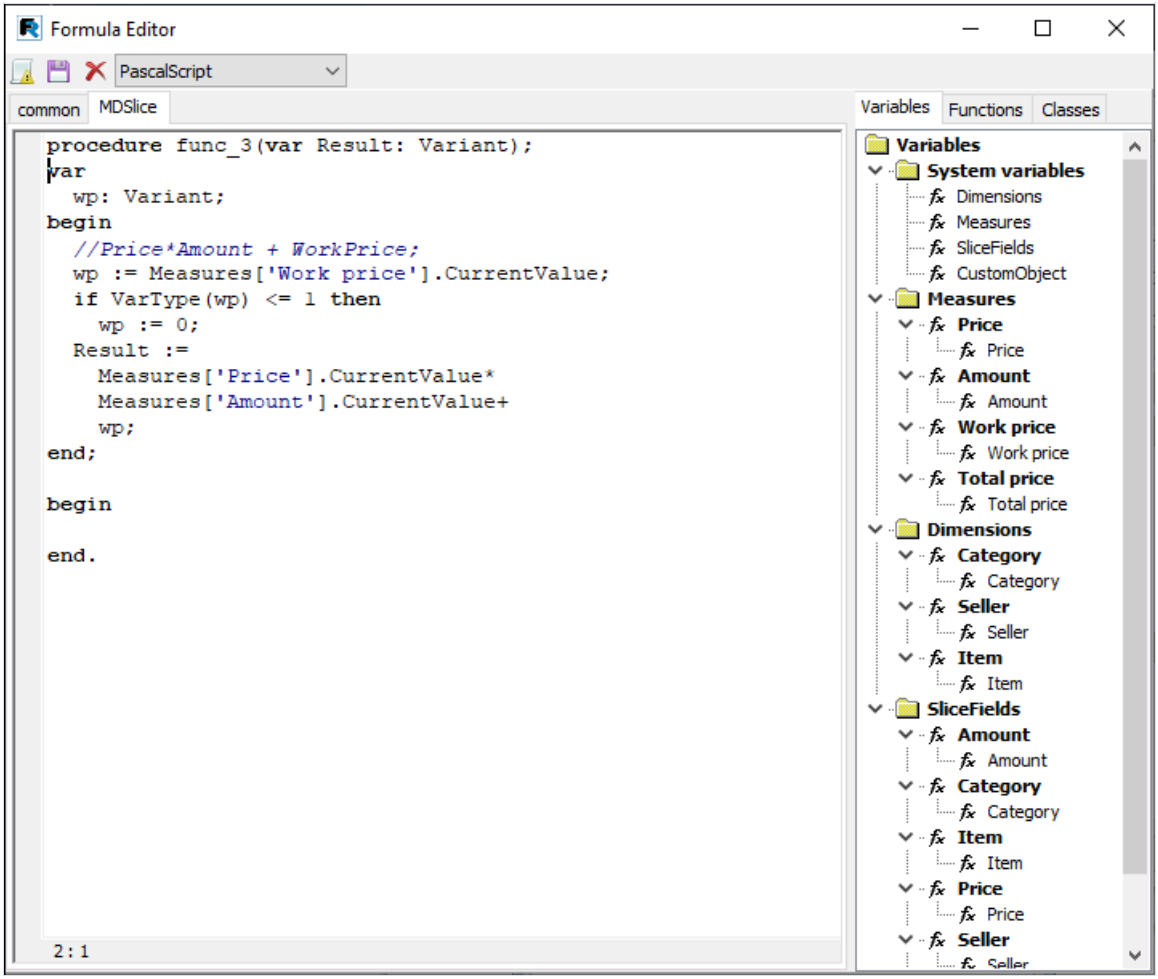
TfcxCustomObject

- Value - pointer to a user object or memory area
- ValueIsNil - checks whether value is empty : used for checking before initial object creation or memory allocation

Here is an example of a calculated measure value on other measures:

```
Sum = Price * Amount + WorkPrice
```

	procedure func_3(var Result: Variant);	procedure header
	var wp: Variant;	declare variable wp
	begin	
	`wp := Measures['Work price'].CurrentValue;`	assign current measure "Work price" value to wp variable
	`if VarType(wp) <= 1 then wp := 0;`	if wp does not exists (Null or UnAssigned), then treat wp value as 0 (we use this test to prevent addition using Null and UnAssigned values)
	`Result :=`` Measures['Price'].CurrentValue*`` Measures['Amount'].CurrentValue+`` wp;`	define Result as current measure "Price" value multiplied by current value of measure "Amount" plus wp value.
	end;	



A function which calculates the measure value on other measures is called once for each cell (dimension intersection). If "Calc all cells" is not set then the function is not called for empty cells.

A function which calculates the measure value on source data is called for each cell as many times as rows exist in the source data. The function has two arguments: Result and Final:

Result - current measure value

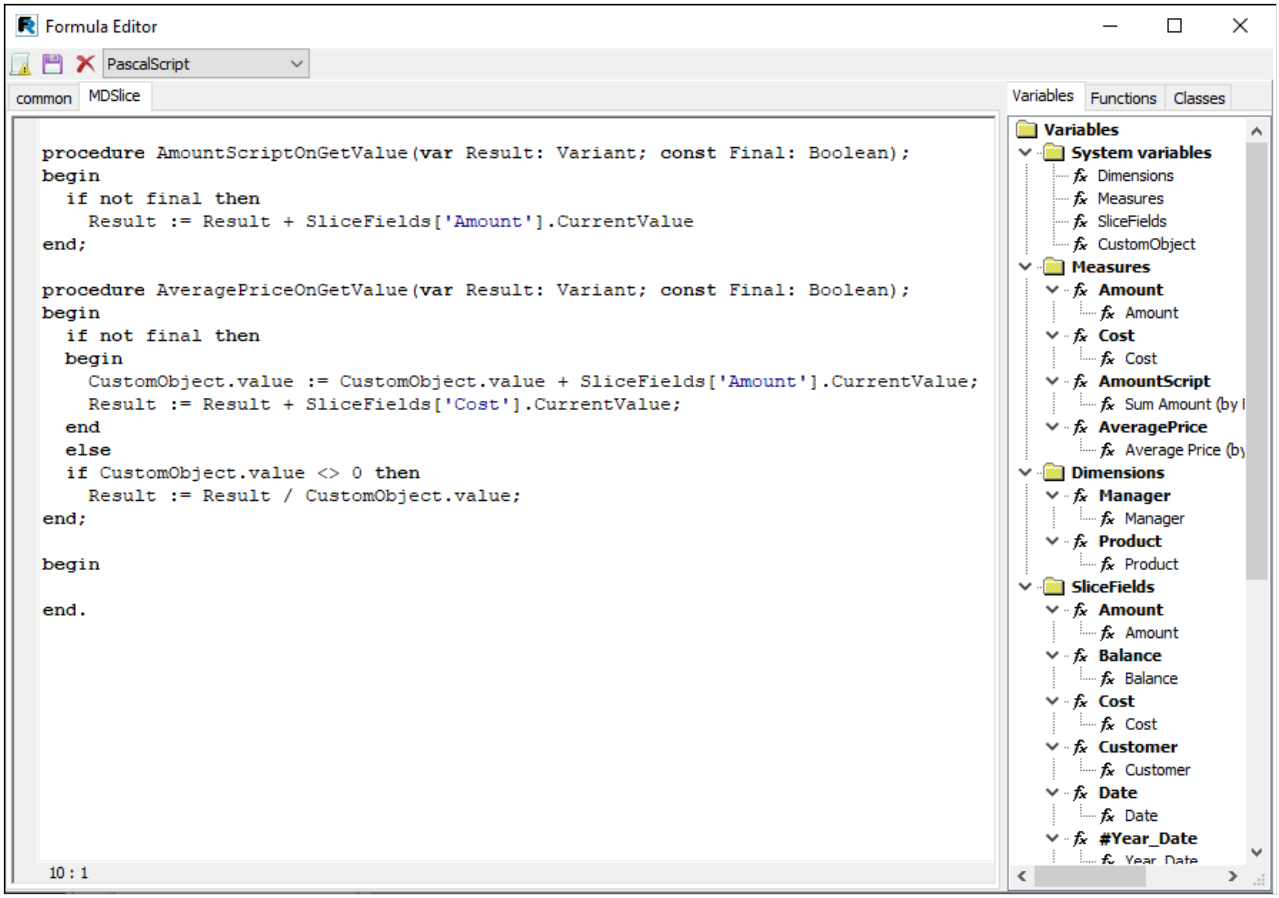
Final - true when this is a final call, otherwise false

The final call occurs for each non-empty cell when the source data traverse has finished. In this case the function Final argument has the value "True". The final call is needed for performance supplementary calculations and for memory deallocation.

In the following example there are two functions:

AmountScriptOnGetValue function - calculates sum of "Amount" field : note that summation occurs only when Final = False

AveragePriceOnGetValue function - calculates average price : requires the summation of values of "Amount" and "Cost" measures and calculation of the average in the final call : note that the CustomObject variable is used to store the sum of the "Amount" measure : memory deallocation is not needed as memory has not previously been allocated



Measures with calculated filters

Measures can have a filter defined by a script. A filter is used if a measure is to be calculated only for those rows of source data that meet some criteria. The script is run for each row of the source data before the calculation of the measure occurs. The script checks whether source rows meet one or more criteria. The script return value (Boolean type) controls whether or not the rows are used for the measure calculation. A 'False' return value means skip that particular row, while 'True' means accept it.

A filter script has access to the following variable:

- field list - SliceFields: TfcSliceFields

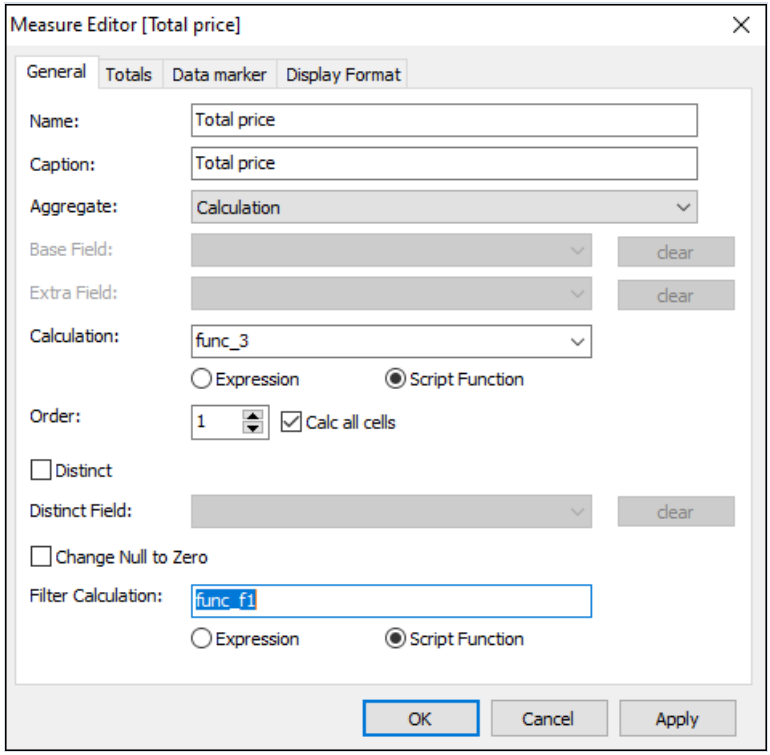
Let's look at an example:

We need to calculate a customer's prepayments. Prepayment is modelled as a negative balance, so we need to sum only negative balances, that is, to create a calculated measure on the "Balance" field with aggregate function "Sum". We also need to create a filter for this measure to accept only rows with Balance < 0:

```
begin
  Result := SliceFields['Balance'].CurrentValue < 0; // test field with "Balance" name
end
```

The calculated filter can be applied both to regular and to calculated measures.

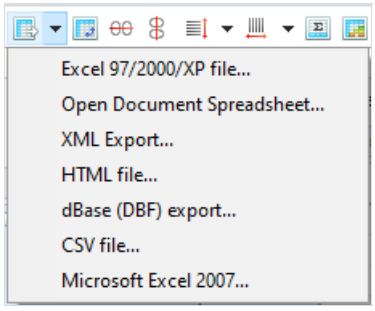
The filter is created on the "Filtering" page of the measure editor:



Data export

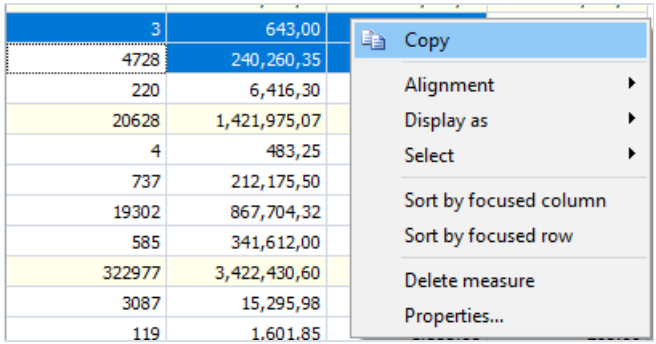
FastCube can export grid data to a number of formats for further editing, archiving or sending by email, etc.

Presently export is supported in five formats: Excel, ODS, XML, HTML and DBF. Export in each format is initiated from buttons on the toolbar:



Exporting uses either the grid data or the source cube data, depending on the export format chosen. So, for example, Excel export uses the dimensions and measures in the grid, while DBF export is from the source cube data.

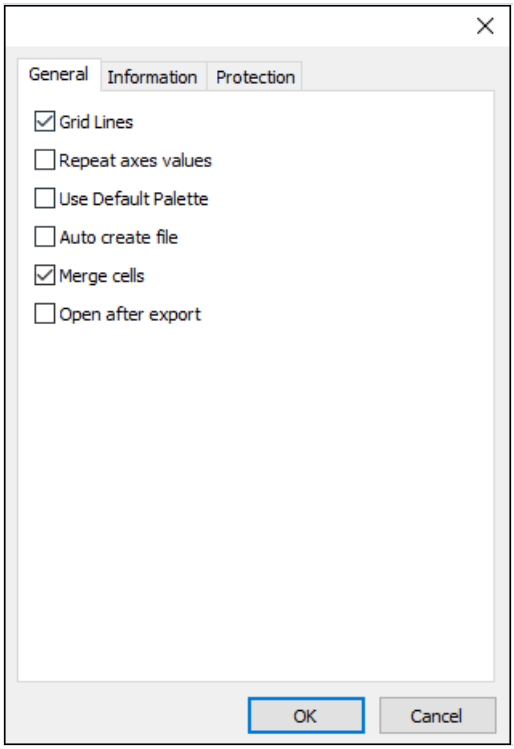
In addition to exporting data, FastCube can copy data from a selected region to the clipboard. A region in the grid can be selected either using the mouse or by pressing the Shift+arrow keys on the keyboard. Copying to the clipboard is done using the usual 'Ctrl+C' or 'Ctrl+Insert' combinations or using the context menu:



Export to XLS

Excel is a Microsoft application for working with electronic spreadsheets. It is included in Microsoft Office. XLS files can also be opened with OpenOffice and LibreOffice.

When exporting to Excel format the output file settings are made in the export dialog:



Export settings:

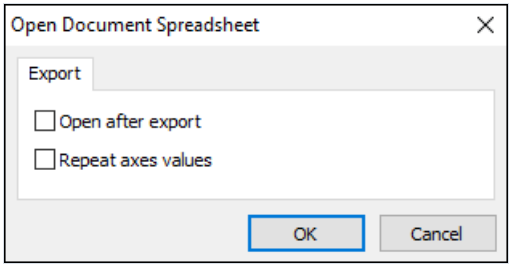
- Repeat axes values - grouped values in grid headers are repeated for each row/column
- Grid lines - whether or not to draw grid lines
- Open after export - open file immediately after export (requires Excel, OpenOffice or LibreOffice)
- Auto create file - create the file name without showing the file save dialog

Export to ODS

OpenDocument Format (ODF : OASIS Open Document Format for Office Application) was designed by OASIS and based on the XML format used in OpenOffice.

FastCube supports export to the table type OpenDocument format (.ods). These files can be opened in OpenOffice or LibreOffice.

When exporting to ODF format the output file settings are made in the export dialog:



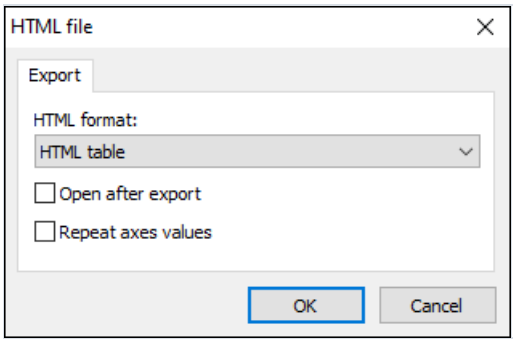
Export settings:

- Repeat axes values - grouped values in grid headers are repeated for each row/column
- Open after export - open file immediately after export (requires OpenOffice or LibreOffice)

Export to HTML

HTML (Hypertext Markup Language) is regarded as the standard language for documents on the Internet. It is used for creating relatively simple but well designed documents. HTML supports hypertext linking as well as simple document layout.

When exporting to HTML format the output file settings are made in the export dialog:



Export settings:

- HTML format

HTML table - export as a normal HTML table (<table> tag)

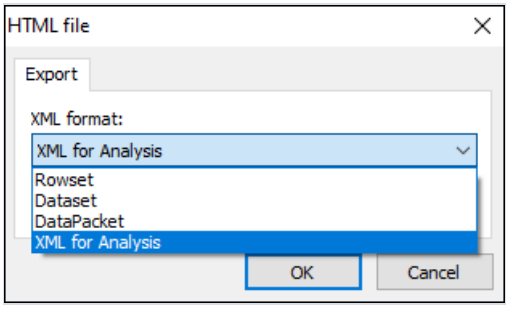
Excel Worksheet HTML - export in HTML format containing Excel extensions

- Repeat axes values - grouped values in grid headers are repeated for each row/column
- Open after export - open file immediately after export

Export to HTML

XML (Extensible Markup Language) is an extendable markup language. XML is intended for structured data storage and also for information exchange between different programs.

When exporting to XML format the output file settings are made in the export dialog:



Export parameters:

- XML format:

Rowset - export cube data in a format compatible with ADO Rowset

Dataset - export cube data in a format compatible with .Net Dataset

DataPacket - export cube data in a format compatible with Embarcadero ClientDataset

XML for Analysis - export grid data in XML for Analysis format

- Open after export - open file immediately after export

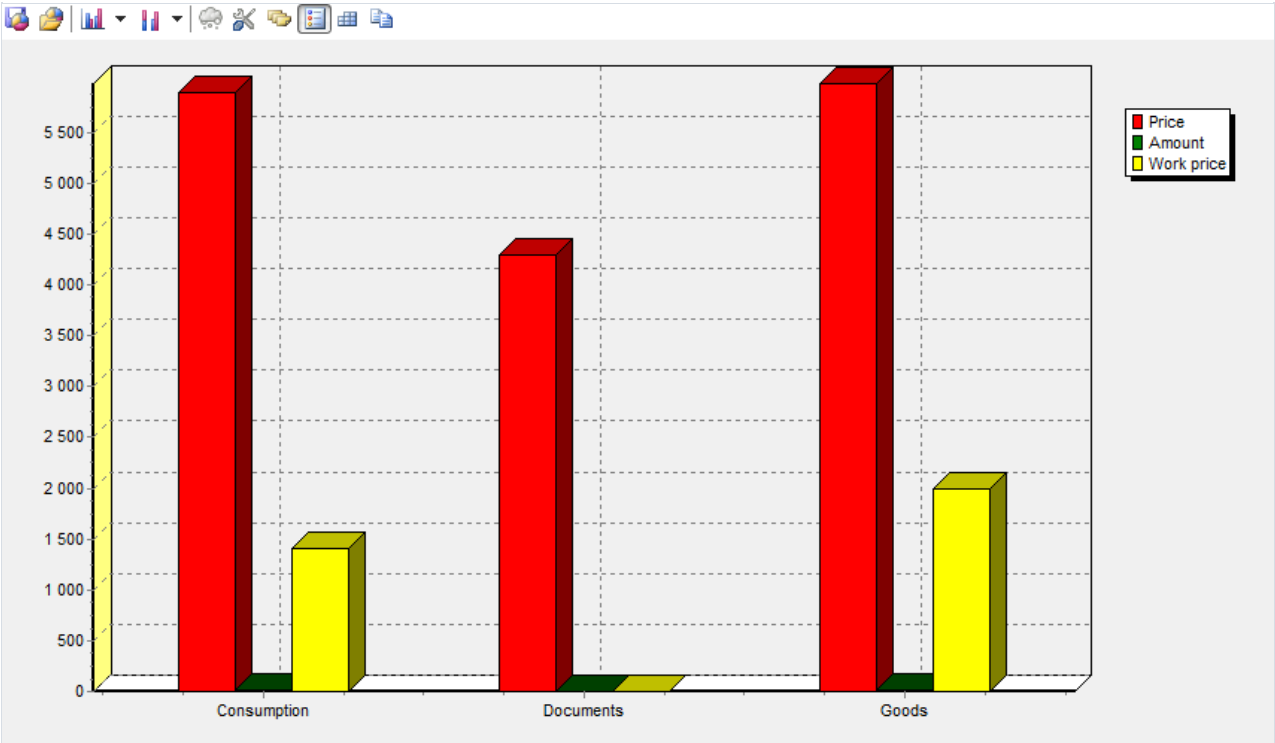
Toolbar

The grid toolbar contains the following set of buttons:

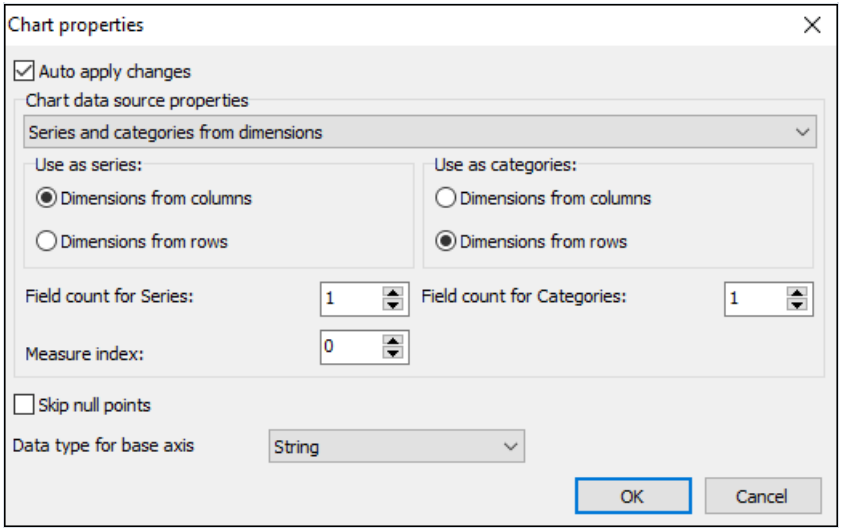
Icon	Command
	Save cube or scheme to file cube files have an mdc extension - they contain source data array, filters and formatting settings scheme files have an mds extension - as well as formatting settings they can also contain filters
	Load cube or scheme file also allows loading of an additional data array to the current cube (e.g. data from another period)
	Clear formatting scheme
	Export data to external formats drop-down menu shows list of available formats
	Transpose grid
	Hide dimension values having only zero or empty data in rows
	Hide dimension values having only zero or empty data in columns
	Select sort mode by rows
	Select sort mode by columns
	Open measure editor dialog
	Open measure highlight options
	Open measure format options
	Open field list window
	Show information about grid structure, data loading elapsed time, creation of headers and calculation of measures

The Chart

The data in a grid may be presented as a cumulative 2D chart. The TeeChart component is used for display of the chart. Depending on the settings, one of the major dimensions of the rows/columns is used for the chart series and the opposing dimension (i.e. the major dimension in the columns/rows) is used to supply the values in each of the series.



If the source grid does not contain row or column dimensions, an internal representation of a grid with row and column dimensions will be built inside the chart. The chart properties dialog is opened from the toolbar:



The main property is the source of data for the series and categories. Possible sources are: the dimensions and the measures in the grid. The dimension region, number of dimensions for the names of categories/series and the measure index need to be selected appropriate to the selected source.

The Chart Toolbar

The chart toolbar contains the following set of buttons:

Icon	Command
	Save chart layout settings to file
	Load chart layout settings from file
	Select type of chart series
	"Freeze" chart chart will not update automatically
	Chart properties editor
	Display labels on values of series
	Display chart legend
	Display data editor
	Copy to clipboard
	Multiple-bar type

Working with FastReport Report Generator

Building a report with the help of a FastCube grid is very similar to creating standard cross-tab reports in the FastReport environment. To avoid repetition, refer to the parts of the FastReport manual dealing with cross-tab reports. However, the practicalities of creating a grid-based report are described here.

All grids use a multi-dimensional data structure (the cube) as their data source. A cube must contain data in a format that is suitable for grid construction. This minimizes the time required for building and setting up the grid. One cube may serve as a data source for an unlimited number of grids having different structures. Two elements are required for building a report:

1. a cube (TfcxpCube) - the data source for the grid
2. a cross-view (TfcxpCrossView) - the object representing the data in the grid

To build a grid-based report place a TfcxpCube  object on the data sheet from the FastCube 2 palette. Then using the object inspector connect a dataset to this object, or enter the name of a previously created cube file.

The TfcxpCube object's main properties are:

- Dataset - a FastReport dataset : at the moment only TfrxDBDataset is supported
- FileName - a file name of a saved cube
- Active - shows whether or not the cube is active

if Active = True then multi-dimensional data structures are created inside the cube : these structures are filled either with the values from the Dataset or with the data from the cube file

if Active = False then the cube becomes inactive and its internal structures are empty

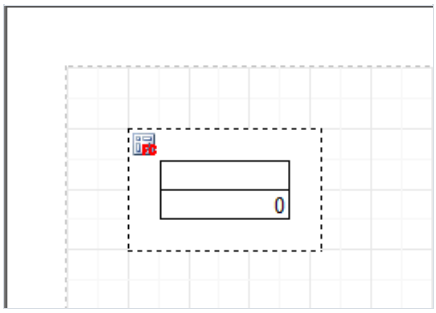
To build a cross-view the cube report should be active. To activate the cube report set its Active property to True.

The TfcxpSliceGridProvider object can be used as an alternative data source for TfcxpCrossView. TfcxpSliceGridProvider enables linkage of a TfcxSliceGrid component on the application's form with the TfcxpCrossView object.

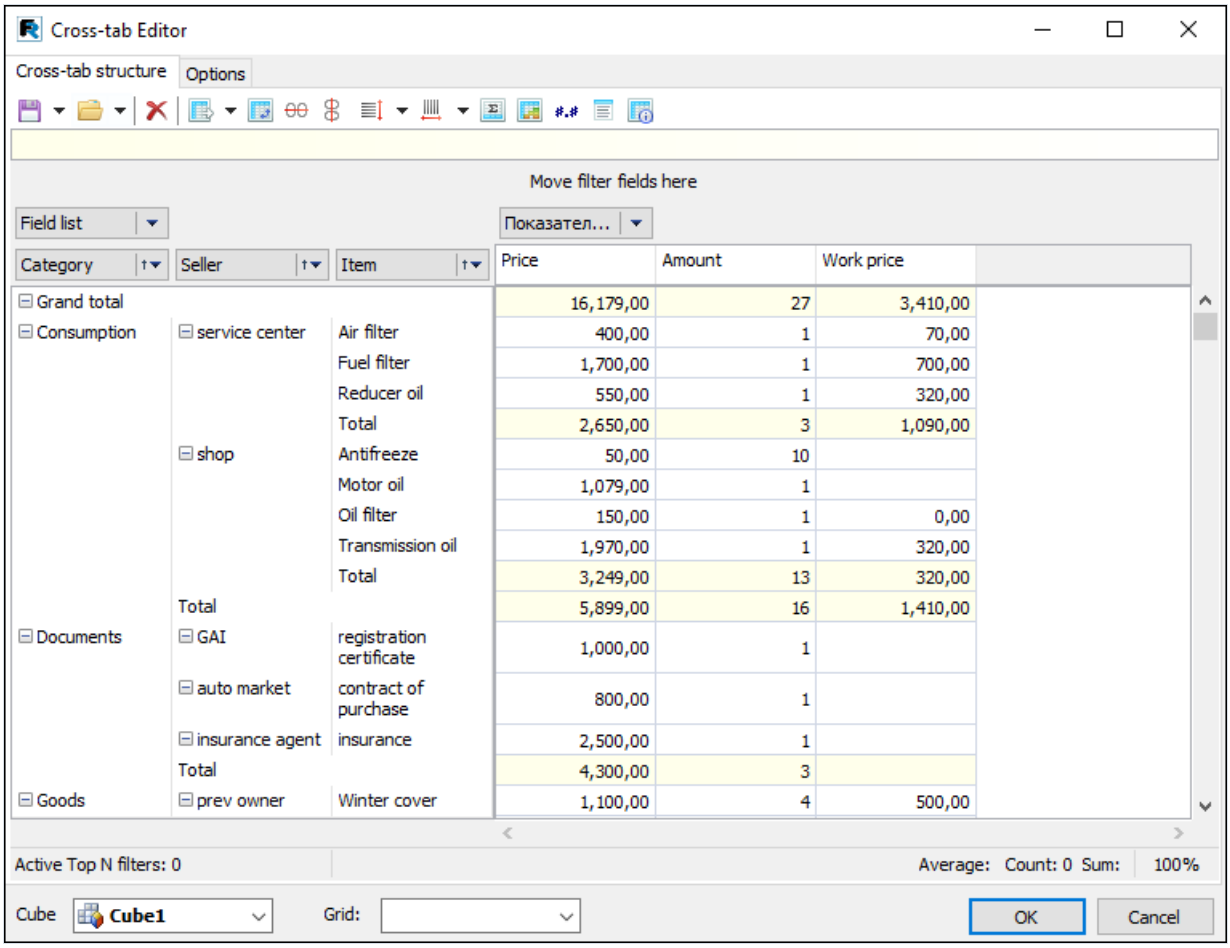
The TfcxpChartView component enables inclusion of a chart in report. The TfcxpChartProvider object enables linkage of a chart with a component on the application's form.

The cross-view object

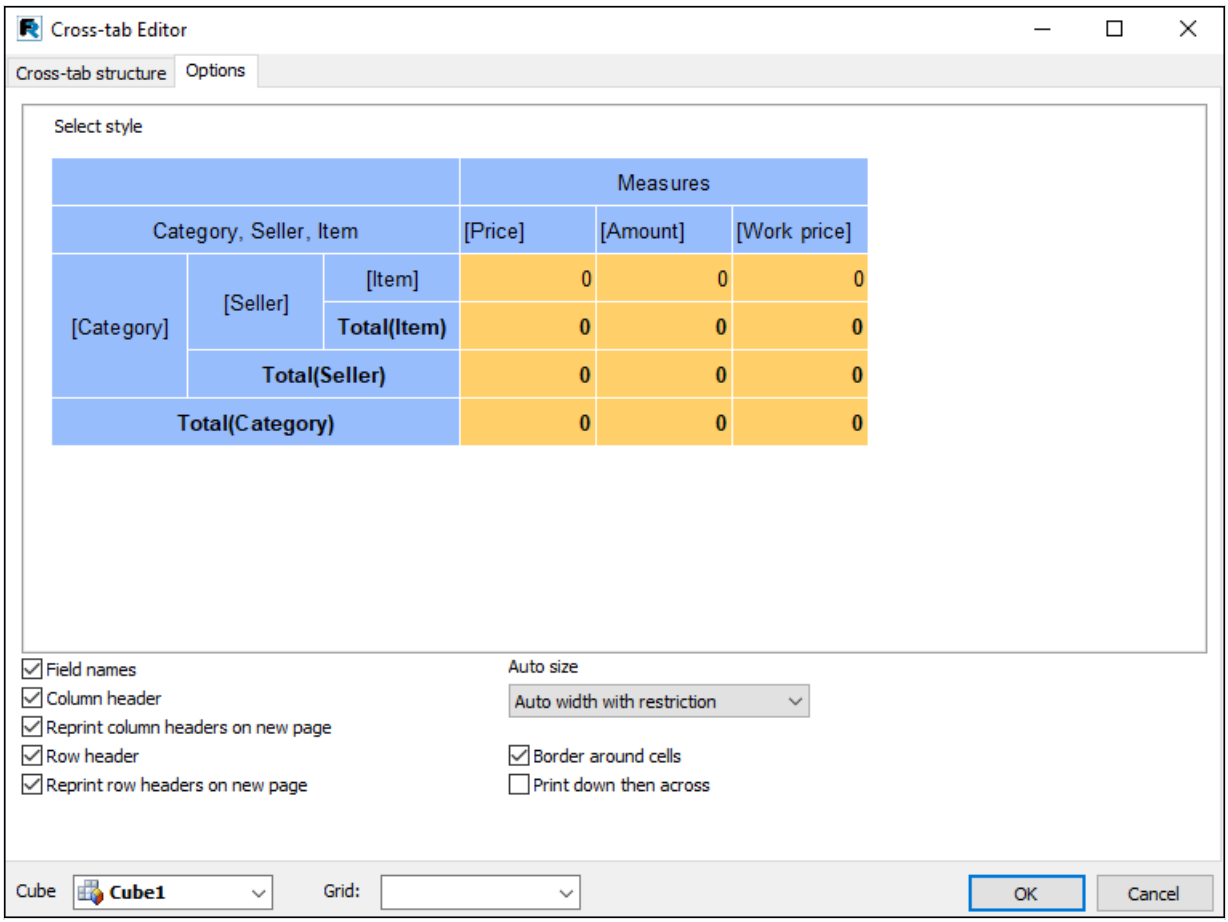
The cube data is output using a cross-view object, TfcxpCrossView. Place the "FastCube 2 Cross-tab" object on the report design page:



The main settings for the object are listed in its editor, which is opened by double-clicking on the object. The cross-view editor has two pages of settings and, at the bottom, a drop-down list of the available cubes. After a cube is selected from the list, the first page of settings is displayed (see below). There is a detailed description of the cross-view (grid) toolbar and the grid itself earlier in this manual.



The second page of the editor contains a number of settings related to the cross-view layout:



The diagram is a representation of the cross-view object as it currently appears in the report, using the currently selected style. The drop-down "Select style" menu gives a choice of one of six preset color schemes for the cross-view display layout. At the bottom of the page there are options for customizing the appearance of the cross-view:

- Field names - show field (dimension) names
- Column header - show columns headers
- Reprint column header on new page - repeat column headers on new pages
- Row header - show row headers
- Reprint row header on new page - repeat row headers on new pages
- Border around cells - show border around cells
- Print down then across - if the cross-view is bigger than a page, it is printed first down and then across
- Auto size - defines how cell width and height are calculated (see below)

The Cross-view object properties (TfcxpCrossView) are:

Property	Type	Description
data related settings		
Cube	TfrcxpCube	data source - cube
representation related settings		
ShowNames	Boolean	show dimension names
ShowColumnHeader	Boolean	show column header
ShowRowHeader	Boolean	show row header

Property	Type	Description
RepeatColumnHeaders	Boolean	repeat column headers on new pages
RepeatRowHeaders	Boolean	repeat row headers on new pages
cell size settings		
PaintSizes	class	(see table below)
other settings		
DownThenAcross	Boolean	if cross-view is larger than the page, it is printed first down and then across
GapX, GapY	Extended	horizontal and vertical spacing
NextCross	TfrcCrossView	next cross-view object
NextCrossGap	Extended	spacing from the next cross-view object

The PaintSizes members are:

Property	Type	Description
AutoSizeStyle	enumeration	style of automatic size calculationmay have the following values: - ssDefault : DefaultColWidth, DefaultRowHeight are used - ssBySlice : by the size of "actual" grid in editor - ssAutoColWidth : automatic width calculation - ssAutoColWidthRestrict : automatic width calculated with value limitation - ssAutoRowHeight : automatic height calculation - ssByMemoSize : by the size of memo objects nested in cross-view
MaxColWidth	Extended	width limit when AutoSizeStyle = ssAutoColWidthRestrict
DefaultRowHeight, DefaultColWidth	Extended	default values for cross-view drawing

There are several styles for setting the cell width and height while creating the report:

ssDefault - the width and height of all the cells are set to DefaultColWidth and DefaultRowHeight, respectively

ssBySlice - the sizes of the external grid or the editor grid are used

ssAutoColWidth - automatic column width calculation : column width is set according to the values and the fonts in use : the width is large enough to fit all the data in the column

ssAutoColWidthRestrict - also automatic column width calculation (as before) but limited to a MaxColWidth value : if width exceeds MaxColWIdth then the height of the affected row increases

ssAutoRowHeight - automatic row height calculation : row height is set according to the values and fonts in use : the height is large enough to fit all the data in the row, providing that the column width is set to DefaultColWidth.

ssByMemoSize - by the size of the memo-objects, which are embedded in the TfrcCrossView object in the FastReport designer : memo-objects are able to change their size only when AutoSizeStyle = ssByMemoSize

If required the print layout of the cross-view may be further customized. This includes the selection of font (size, style, color etc.), background fill, alignment etc. separately for each memo-object embedded into the TfrcCrossView in the FastReport designer. All these changes are made in the object inspector : the properties are similar to the properties of the TfrxCrossObject component in FastReport.

Example of FastCube Use

This chapter describes examples of using FastCube.

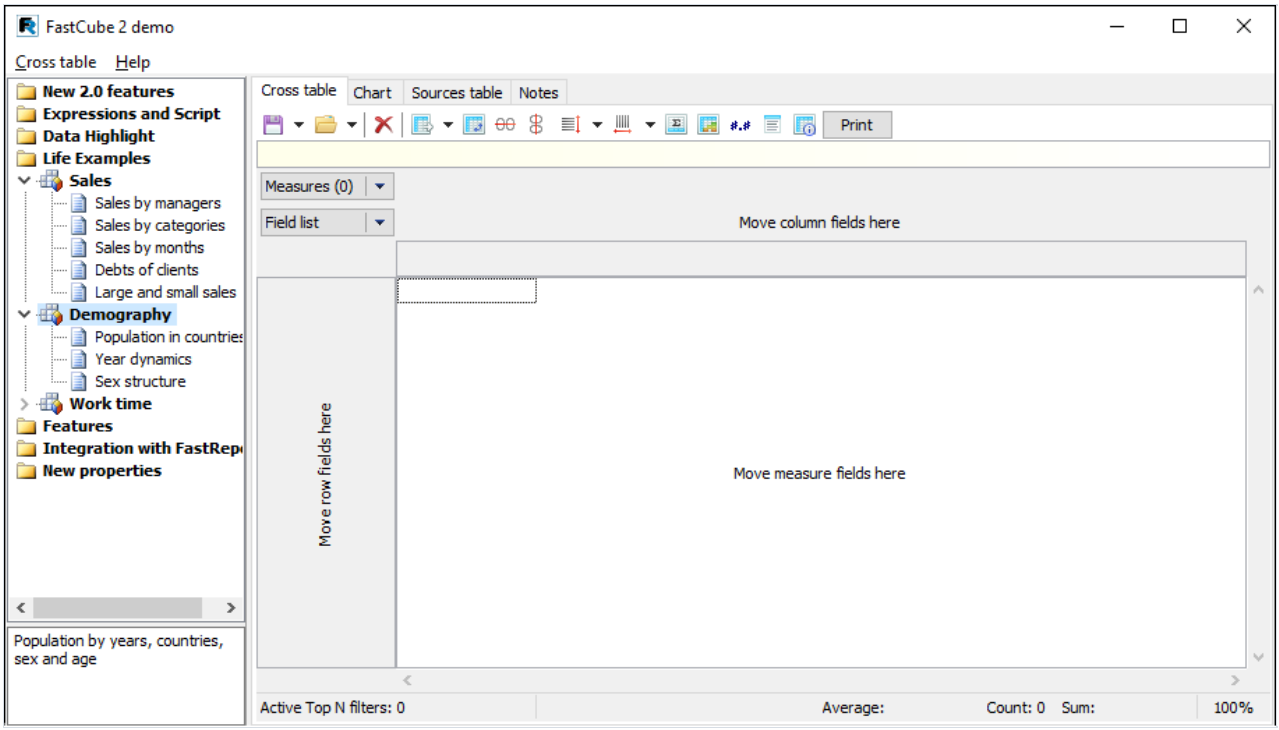
The examples can be seen in the demo program 'Advanced'.

Example "Demography"

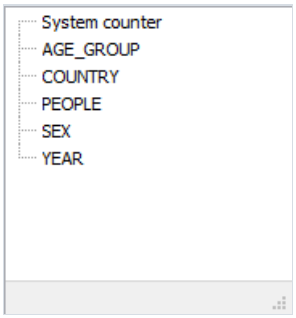
Let's show an example of using a multi-dimensional table holding demographic data for a number of countries. The source data was taken from the free service of the U.S. Census Bureau (http://www.census.gov/ipc/www/idbprint.html).

After some insignificant processing the data was loaded in the cube, so allowing the data to be displayed in a convenient way.

After loading the data into FastCube, start with the empty grid below (see 'Advanced' demo, 'Life Examples', 'Demography'):



Clicking on the "Field list" element, show this drop-down list:

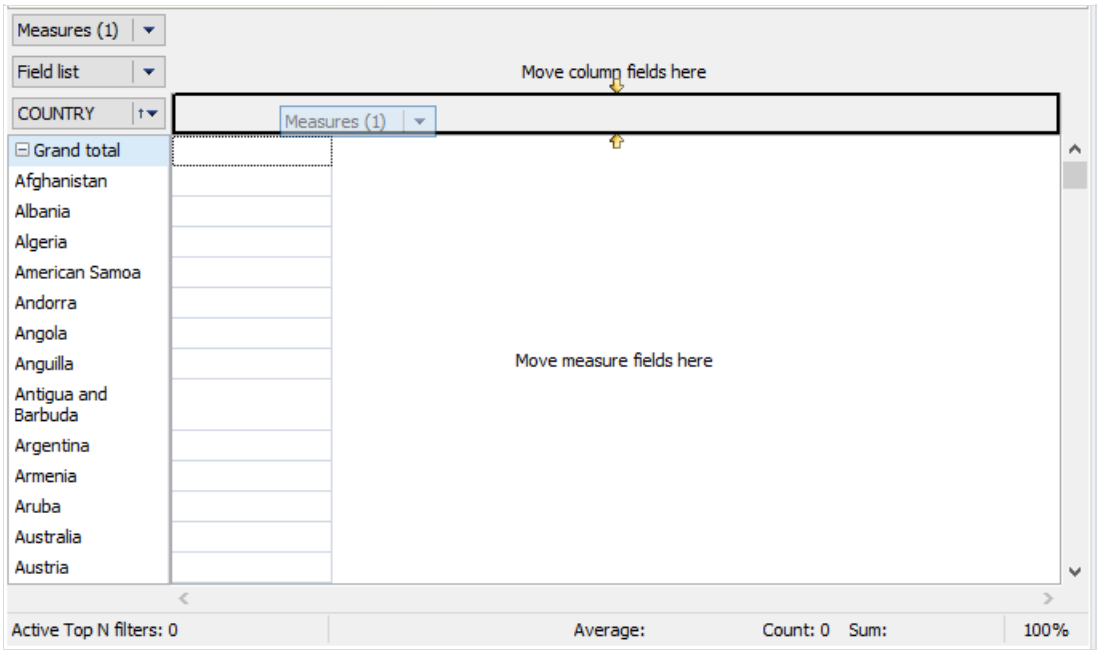


What can be done with this data? Let's create several examples.

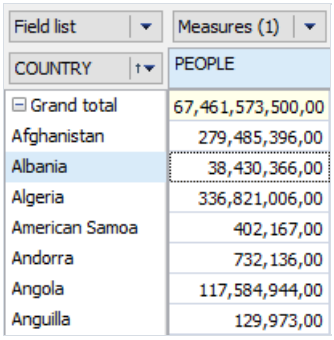
If you have not been able to reproduce these examples, you can open the ready-made examples from the Advanced demo, "Life Examples", "Demography".

Example 1: Population by country

Drag the "Country" field to the row region and the "People" field to the data region, then move the measures field that is already located in the filters region into the columns region:



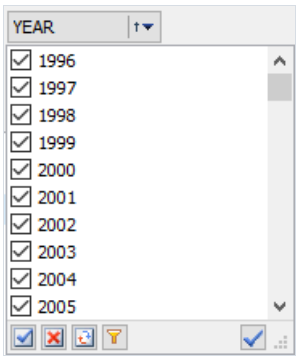
These steps give the grid below:



It is easy to see that the data in the grid is incorrect. The reason for this is that here the "People" measure sums the values for all years. So we need to specify the required year. To do this, we must drag the "Year" field into the filters region.

When a field is added to the filters region of the grid, an element appears to enable data filtering through a drop-down list.

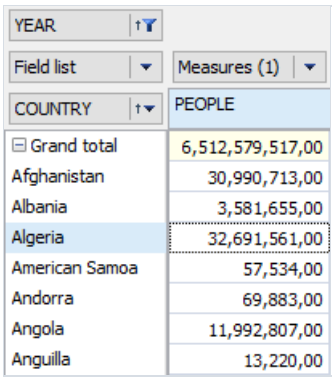
Let's click on the drop-down list of the "Year" field:



As shown, the grid contains data for 11 years. But we only need the data for 2006 - this can be achieved in several ways:

1. by un-checking every year not needed, one by one (i.e. from 1996 to 2005)
2. by un-checking just 2006 and then clicking the "Inverse" button
3. by clicking the "Clear all" button and then clicking on 2006
4. by clicking on 2006 while holding down the 'Ctrl' key on the keyboard

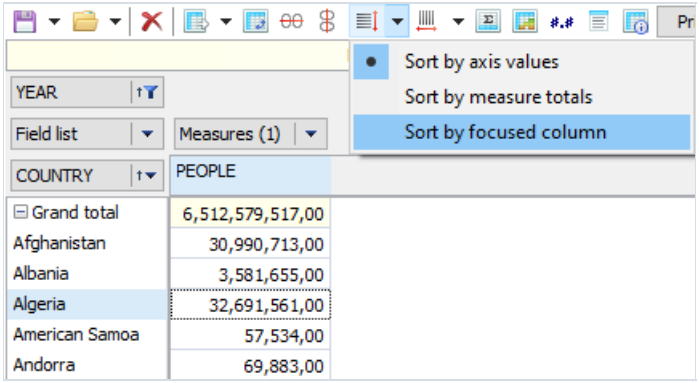
After these changes to the "Year" dimension the grid looks like this:



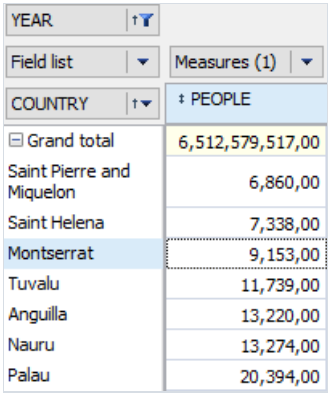
Note that the title 'Year' is displayed in italic, which means that a filter is applied to it.

Also note that the values of the "Country" dimension are listed in the alphabetical order. For our analysis it would be more useful to list the countries in descending order of population. To do this, we need to sort the data by the value of the measure.

Click on any data cell in the grid, in the column which will be used for sorting the data, and select the option "Sort by focused column" in the toolbar:



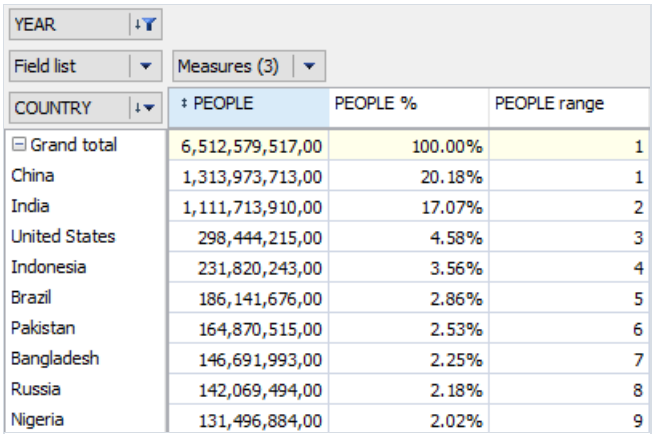
This results in the grid displaying the data sorted by the value of the "People" measure (population), but in ascending order. Note that the column used for the sort is marked with a special symbol (diamond):



To change the sort order to descending, double-click on the "Country" dimension (or do the same on the drop-

down dimension menu).

Adding Percentage and Rank columns to the measure values, shows that China takes the first place by population (20% of the total population) and the United States only takes third place (4% of the total population).

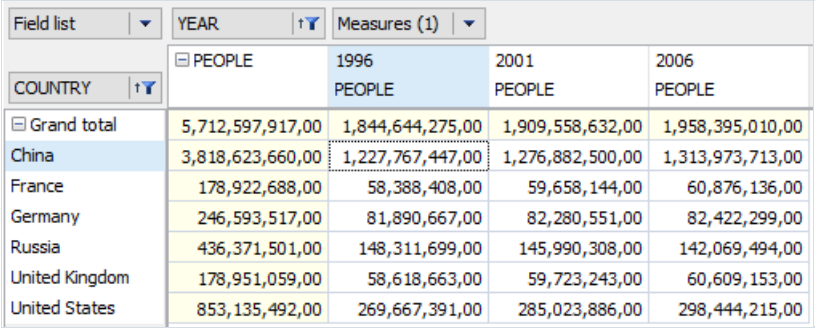


YEAR			
Field list	Measures (3)		
COUNTRY	PEOPLE	PEOPLE %	PEOPLE range
Grand total	6,512,579,517,00	100.00%	1
China	1,313,973,713,00	20.18%	1
India	1,111,713,910,00	17.07%	2
United States	298,444,215,00	4.58%	3
Indonesia	231,820,243,00	3.56%	4
Brazil	186,141,676,00	2.86%	5
Pakistan	164,870,515,00	2.53%	6
Bangladesh	146,691,993,00	2.25%	7
Russia	142,069,494,00	2.18%	8
Nigeria	131,496,884,00	2.02%	9

Example 2: The dynamics of population growth

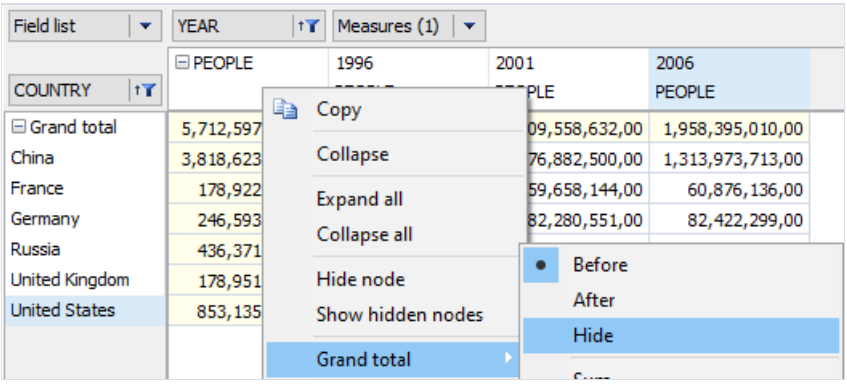
This example is based on the grid created above.

To achieve our objective we need to move the "Year" dimension from the filters region to the columns region, before the measures field, and add the values '1996' and '2001' to the years filter. As a result, the grid will show the population size in 5-year intervals between 1996 and 2006. Further, we can filter the country dimension to show just six countries: China, France, Germany, Russia, United Kingdom and the USA:

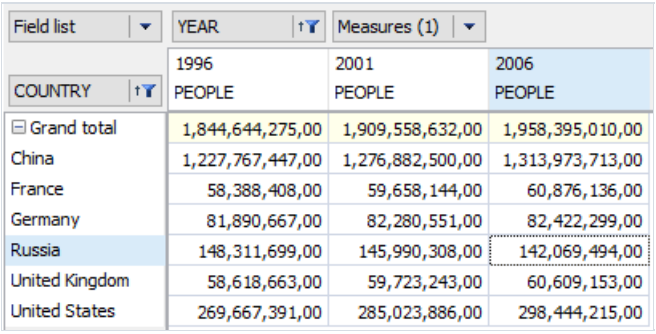


Field list	YEAR	Measures (1)		
COUNTRY	PEOPLE	1996 PEOPLE	2001 PEOPLE	2006 PEOPLE
Grand total	5,712,597,917,00	1,844,644,275,00	1,909,558,632,00	1,958,395,010,00
China	3,818,623,660,00	1,227,767,447,00	1,276,882,500,00	1,313,973,713,00
France	178,922,688,00	58,388,408,00	59,658,144,00	60,876,136,00
Germany	246,593,517,00	81,890,667,00	82,280,551,00	82,422,299,00
Russia	436,371,501,00	148,311,699,00	145,990,308,00	142,069,494,00
United Kingdom	178,951,059,00	58,618,663,00	59,723,243,00	60,609,153,00
United States	853,135,492,00	269,667,391,00	285,023,886,00	298,444,215,00

Now the first column shows the sum of the other columns and does not make any sense in this context, so we can hide the column. This is done through the context menu for the People item - 'Grand total->Hide':

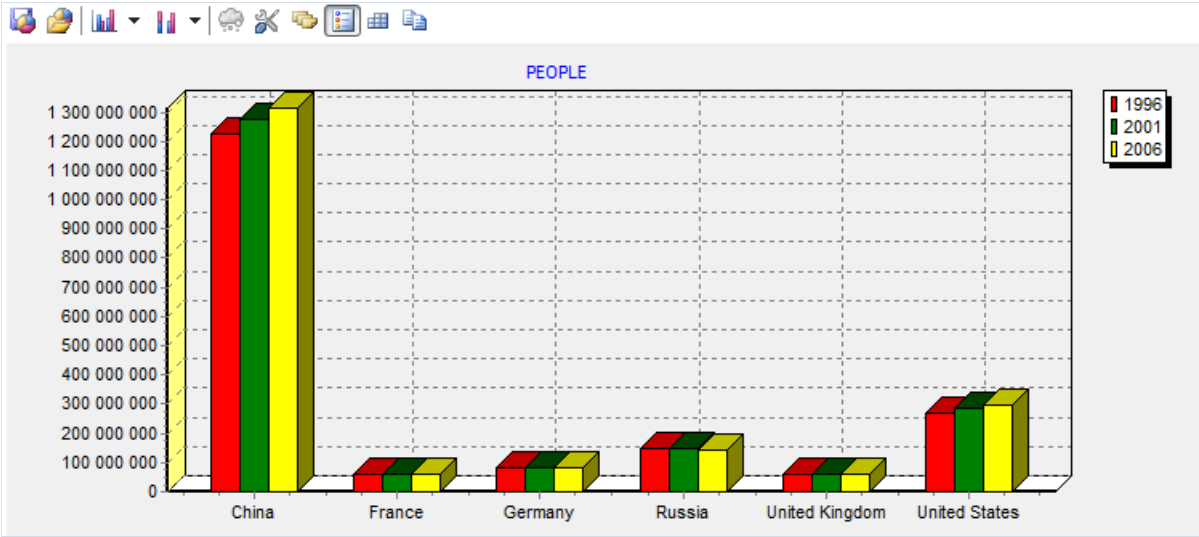


The final grid looks like this:



Field list	YEAR	Measures (1)	
COUNTRY	1996 PEOPLE	2001 PEOPLE	2006 PEOPLE
Grand total	1,844,644,275,00	1,909,558,632,00	1,958,395,010,00
China	1,227,767,447,00	1,276,882,500,00	1,313,973,713,00
France	58,388,408,00	59,658,144,00	60,876,136,00
Germany	81,890,667,00	82,280,551,00	82,422,299,00
Russia	148,311,699,00	145,990,308,00	142,069,494,00
United Kingdom	58,618,663,00	59,723,243,00	60,609,153,00
United States	269,667,391,00	285,023,886,00	298,444,215,00

This grid is useful for presenting the raw demographic data, but a far greater impact can be made when presented as a chart:



The chart clearly shows the relative sizes of the populations.

But to reveal which populations are increasing in size and which are falling, the grid data can be emphasized using the highlight tools:

Field list	YEAR	Measures (1)	
COUNTRY	1996 PEOPLE	2001 PEOPLE	2006 PEOPLE
Grand total	1,844,644,275,00	1,909,558,632,00	1,958,395,010,00
China	1,227,767,447,00	1,276,882,500,00	1,313,973,713,00
France	58,388,408,00	59,658,144,00	60,876,136,00
Germany	81,890,667,00	82,280,551,00	82,422,299,00
Russia	148,311,699,00	145,990,308,00	142,069,494,00
United Kingdom	58,618,663,00	59,723,243,00	60,609,153,00
United States	269,667,391,00	285,023,886,00	298,444,215,00

Example 3: Population by sex and age

In this example we will try to answer the following questions:

1. children of which gender are born more frequently?
2. does this gender ratio change with age?

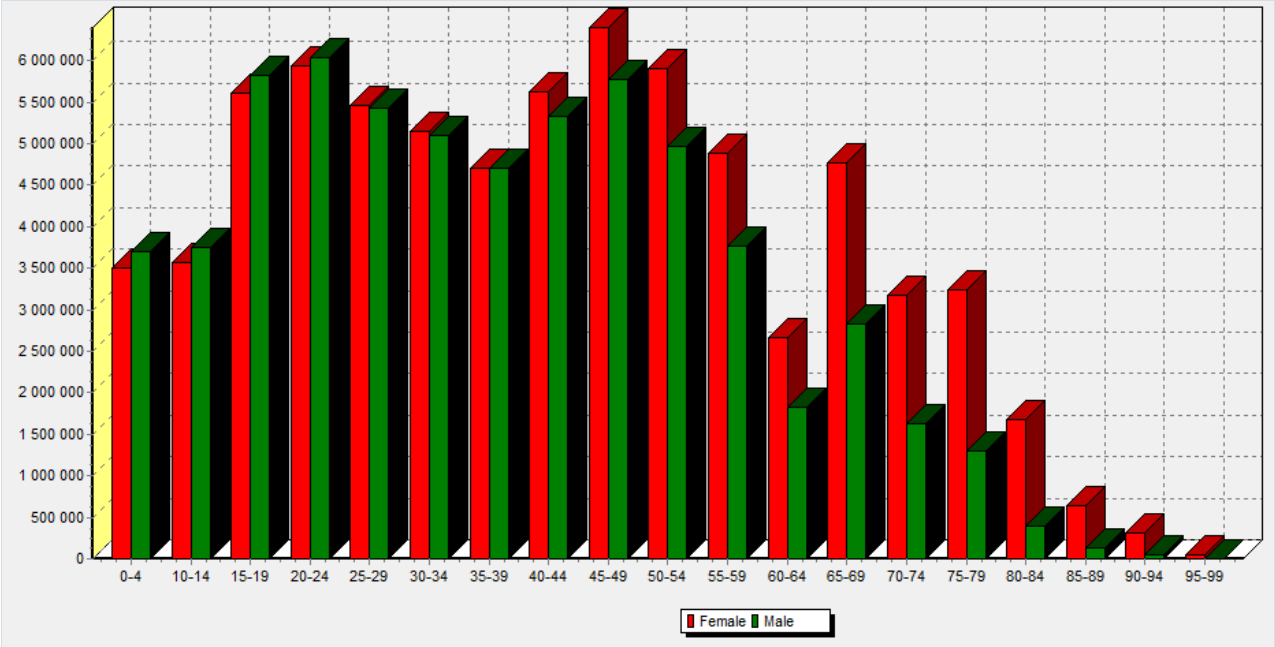
To answer these questions we will make the following changes to the grid:

1. drag the "Age_group" field from the Field list into the row region
2. drag the "Year" and "Country" fields into the filters region
3. drag the "Sex" field from the Field list and the measures field to the measure fields region
4. exclude the values of '100+' and '5-9' from the values of the "Age_group" dimension
5. leave only 2006 in the "Year" dimension
6. leave only Russia in the "Country" dimension

These steps produce the following grid:

YEAR	COUNTRY		
Field list	Measures (1)	SEX	
	PEOPLE		
AGE_GROUP	Female	Male	
Grand total	73271679	62549510	
0-4	3509867	3706851	
10-14	3568550	3753774	
15-19	5601275	5824867	
20-24	5938204	6036309	
25-29	5454448	5426420	
30-34	5151138	5093157	
35-39	4700485	4701412	
40-44	5621646	5331305	
45-49	6398180	5774332	
50-54	5910562	4975127	
55-59	4892784	3773697	
60-64	2660493	1824590	
65-69	4764813	2830270	
70-74	3175555	1621404	
75-79	3236414	1294706	
80-84	1674391	399073	
85-89	639783	130747	
90-94	319176	45401	
95-99	53915	6068	

Just as previously, the raw data is less informative than a chart (though the figures may be important for printed reports), so let's switch to the chart page:



A chart makes it easy to answer our initial questions:

1. more boys are born than girls
2. however, from the age group 40-44 more women survive than men