

Unleashing the Crosstab in Crystal Reports: The Grid Value Functions



Applies to:

Crystal Reports for Enterprise (SAP BusinessObjects 4.0), Crystal Reports 2011, Crystal Reports 2008.

For more information, visit the [Business Objects homepage](#).

Summary

Crosstabs are integral reporting tools for any BI needs. The Grid Value functions in Crystal Reports are very powerful in customizing the content or format of a crosstab. However, they often remain an unexplored terrain by the developers thanks to a dearth of comprehensive explanatory information and their usage complexity.

This article aims to bring forth a detailed study of these grid functions in tapping the full potential of a crosstab in Crystal Reports and illustrate some of their powerful usage.

Crystal Reports for Enterprise version has been used in this article for illustrative purposes, although the same fundamentals of grid functions also apply to Crystal Reports 2008 and 2011.

Author: Anil Kumar, Kiran Jayakumar

Company: Deloitte Consulting LLP

Created on: 9 October 2011

Author Bio



Anil Kumar has around six years of experience in BI and Data warehousing technologies and is currently working for Deloitte Consulting LLP. His research work includes various International publications in the field of semantic analysis of textual data, semantic tagging of text and concept based navigation in ontology.



Kiran Jayakumar is a Specialist Master at Deloitte Consulting LLP and has over seven years of SAP implementation experience. He has led and implemented data warehouse solutions across Logistics, Asset Management and Human Resource functional areas. He possesses strong capabilities in designing and implementing complex data warehouse solutions.

Table of Contents

Introduction	3
Importance of Grid Value Functions	3
Grid Value Functions – Quick Syntax.....	3
Understanding the Grid Value Functions.....	4
Cell Index Functions.....	4
Statistical Functions	5
Row/Column Group Functions.....	5
Row/Column Path Index Functions.....	5
Row/Column Label Functions	6
Cell Value Functions	6
Using the Grid Value Functions	7
Troubleshooting	13
Related Content.....	14
Disclaimer and Liability Notice.....	15

Introduction

Crosstabs are integral reporting tools for any BI needs. The Grid Value functions in Crystal Reports are very powerful in customizing the content or format of a crosstab. However, they often remain an unexplored terrain by the developers thanks to a dearth of comprehensive explanatory information and their usage complexity.

This article aims to bring forth a detailed study of these grid functions in tapping the full potential of a crosstab in Crystal Reports and illustrate some of their powerful usage.

Crystal Reports for Enterprise version has been used in this article for illustrative purposes, although the same fundamentals of grid functions also apply to Crystal Reports 2008 and 2011.

Importance of Grid Value Functions

A crosstab in Crystal Reports, per se in its basic form, is tightly coupled to the data items specified in the crosstab definition. However, usually this does not suffice and some sort of customization is necessitated to achieve the desired output. To extend the customization and functionality of crosstab, Crystal report provides an array of Grid Value functions. These Grid functions equip a report developer to implement any sophisticated data as well as formatting logic in a crosstab.

Some of the scenarios where Grid functions are indispensable are below.

- Specifying calculations in a calculated member (row/column) – percentages, margins, ratios, any mathematical functions.
- Location as well as frequency of calculated members as well as blank rows /columns.
- Conditional formatting.
- Dynamic Row/column header of added rows and columns.
- Addition of new specific member summary rows/columns.
- Addition of new measures as separate rows/columns instead of the displaying in all the cells by default.

Grid Value Functions – Quick Syntax

The table below gives a quick summary and syntax of the Grid Value functions. These functions are grouped into appropriate categories for ease of understanding. The next sections explain these functions in detail.

	Grid Value Function	Return Type	Arguments
Cell Index	CurrentRowIndex	Number(RowIndex)	Nil
	CurrentColumnIndex	Number(ColumnIndex)	Nil
	CurrentSummaryIndex	Number(SummaryIndex)	Nil
Statistical	GetNumColumns	Number	Nil
	GetNumColumnGroups	Number	Nil
	GetNumRows	Number	Nil
	GetNumRowGroups	Number	Nil
	GetNumSummaries	Number	Nil
Row/Column Group Index	GetRowGroupIndexOf()	Number(RowGroupIndex)	Number(RowIndex)
	GetColumnGroupIndexOf()	Number(ColumnGroupIndex)	Number(ColumnIndex)
Row/Column Path Index	GetRowPathIndexOf()	Number(RowIndex) [-1 if not found]	String(RowLabel_Level1), String(RowLabel_Level2), ...
	GetColumnPathIndexOf()	Number(ColumnIndex) [-1 if not found]	String(ColumnLabel_Level1), String(ColumnLabel_Level2), ...
	GetIndexOf()	Number(RowIndex/ColumnIndex)	String(RowName/ColumnName),

) [-1 if not found]	String(RowLabel/ColumnLabel)
Row/Column Label	GridLabelAt()	String(RowLabel/ColumnLabel)	String(RowName/ColumnName), Number(RowIndex/ColumnIndex)
	GridRowColumnValue()	String(RowLabel/ColumnLabel)	String(RowName/ColumnName), Number(RowIndex/ColumnIndex)
Cell Value	GridValueAt()	Number	Number(RowIndex), Number(ColumnIndex), Number(SummaryIndex)
	GetTotalValueFor()	Number	String(RowName/ColumnName), Number(RowIndex/ColumnIndex)

Understanding the Grid Value Functions

For ease of understanding, let's consider a sample crosstab as shown below. The crosstab displays yearly quarterly sales figures for various countries within different regions.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Total	2000				2001					
2				Total	Q1	Q2	Q3	Q4	Total	Q1	Q2	Q3	Q4
3	Total		145560	73470	19900	17330	18045	18195	72090	18840	18410	17450	17390
4	Asia	Total	75150	37890	10000	8650	9690	9550	37260	9790	9370	9300	8800
5		Japan	19050	9900	3400	2300	2000	2200	9150	2400	2300	2350	2100
6		India	10690	5090	1200	1250	1290	1350	5600	1450	1400	1450	1300
7		China	45410	22900	5400	5100	6400	6000	22510	5940	5670	5500	5400
8	Europe	Total	70410	35580	9900	8680	8355	8645	34830	9050	9040	8150	8590
9		UK	20980	9680	2500	2380	2455	2345	11300	2900	2750	2650	3000
10		Germany	18900	9900	3400	2000	2000	2500	9000	2400	2300	2200	2100
11		France	30530	16000	4000	4300	3900	3800	14530	3750	3990	3300	3490

The sections below explain the Grid functions in detail.

Cell Index Functions

These functions are used to return the index of a cell. Each cell in the crosstab is associated with a row index, column index and summary index. Row and column index identifies the cell location while the summary index identifies the measure. The index values are whole numbers starting with 0. The Each cell in the crosstab can contain multiple measures. Each measure (e.g. sales amount, sales quantity, profit etc.) is associated with a summary index starting with 0 for the first measure.

By default, a crosstab in Crystal Reports has cells for row totals and column totals as shown in the sample crosstab as above. These also include row grand total as well as column grand total which usually have the index 0.

The table below gives the index of different cells in the sample crosstab above.

Cell	CurrentRowIndex	CurrentColumnIndex	CurrentSummaryIndex
C3	0	0	0
D3	0	1	0
E3	0	2	0
E4	1	2	0
E8	5	2	0
G6	3	4	0

The above index remains the same even if the row/column total values are hidden. However, the values change when grand total rows are placed at the beginning or at the end.

Note: These functions are often used in conjunction with other formulas to get more sophisticated logic and provide dynamic reference to other rows and columns in calculated members

Statistical Functions

These functions are used to get the total number of rows (columns) and row groups (column groups) in the entire crosstab.

The number of rows/columns include the totals' rows even if they are hidden. Any additional calculated members or blank rows (columns) are not included.

The row groups (column groups) indicate the number of levels viz. number of fields in the rows (columns). For example, in the sample crosstab there are two row groups: first representing the region and second representing the country.

The table below gives the values that these statistical functions return for the sample crosstab above.

GetNumColumns	11
GetNumColumnGroups	2
GetNumRows	9
GetNumRowGroups	2
GetNumSummaries	1

Row/Column Group Functions

These functions return the Row Group (Column Group) Index of the row (column) associated with a cell. The argument should be a row index (column index).

Note: These are important function that are often used in the Insertion formula for a calculated member which determines the location where the calculated member appears in the crosstab and if the calculated member repeats for each member of a row level or not.

These functions are explained further in the later section [Using the Grid Value Functions](#).

Row/Column Path Index Functions

These functions return the row index (column index) of a row (column) referred by that specific row (column) member label. If there are more than one row/column groups, all the hierarchical row/column member labels should be specified in that order (separated by comma) to completely identify that specific row.

Example: In the sample crosstab above, `GetRowPathIndexOf("Asia", "Japan")` would identify the fifth row for Asia and Japan and would return a row index of 2.

If, however, the member labels of the all row level are not specified, the function returns the Totals row for the specified level.

Example: GetRowPathIndexOf("Asia") would return the row index of 1 referring the Totals row for "Asia".

In case that specific member label is not found in the row tree, a value of -1 is returned.

Example: if GetRowPathIndexOf("USA") is specified, it would search for USA in the region level, and would return -1 as this value does not exist for Region level.

Similarly, GetRowPathIndexOf("Japan") would return -1 as there is no Region member with the label "Japan".

Note: The return value of -1 can be used in a condition to check if a member exists in the crosstab or not; especially when the members are likely to change for a crosstab rows/columns.

Row/Column Label Functions

These functions are used to return the member label for a specified row/column level at a particular row/column index.

Example: In the sample crosstab above, GridRowColumnValue("Region", 2) returns "Asia" – the row member label for Region for row with index 2 (in this case fifth row).

These can be used in specifying the row/column heading in the header formula for calculated column or blank column. They can also be used in a conditional formatting formula to conditionally format a crosstab.

Note: These functions can be used in specifying the row/column heading in the header formula for calculated column or blank column. They can also be used in a conditional formatting formula to conditionally format a crosstab

Cell Value Functions

The GridValueAt() function returns the value of a cell referenced by the cell's row, column and summary index. This function when used in conjunction with other functions to get dynamic reference of other cell values can fulfill any sophisticated logic.

Example: In the sample crosstab above, GridValueAt(2,2,0) returns the sales value 3400 for Q1,2000 for Japan, Asia.

The function GetTotalValueFor() returns the total value for row/column level.

Example: GetTotalValueFor("Region") returns 145560.

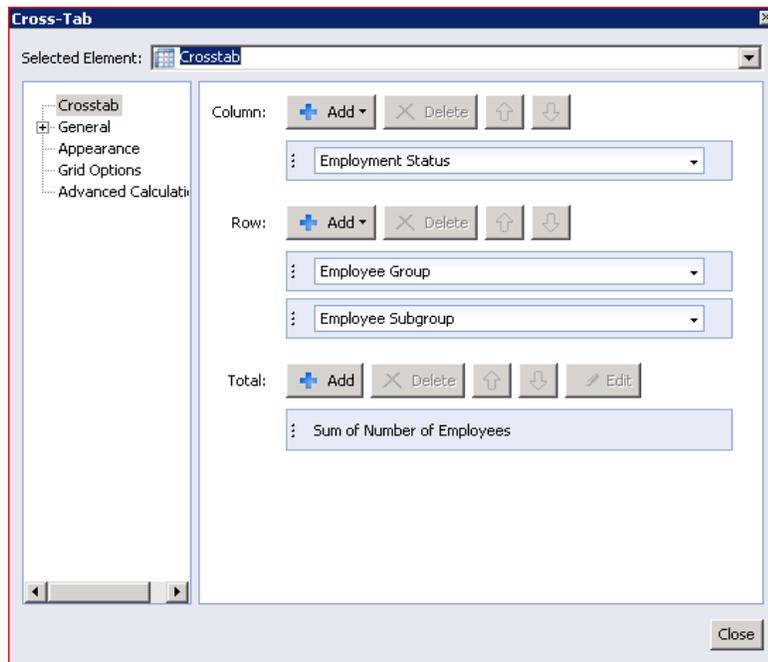
Using the Grid Value Functions

Grid Value functions are used in conjunction with calculated members in a crosstab. These can be used in the formula for Header, Insertion, Group Value or Calculation of the calculated member. To demonstrate the power of these functions, let's consider a sample crosstab and work on some enhancement requirements to it.

The crosstab below displays number of employees in a sample organization by 'Employee Group' and 'Employee Subgroup' as rows and 'Employment Status' as columns.

		Total	Active	Inactive	Retiree	Withdrawn
Total		1,250	1,130	1	15	104
Contract	Total	1	1	0	0	0
	Contract >= 1 Year	1	1	0	0	0
Non-Employee	Total	212	205	0	0	7
	Applicant	178	178	0	0	0
	Contractor	24	19	0	0	5
	Executive	10	8	0	0	2
Regular	Total	1,000	893	1	10	96
	Casual Hourly	1	1	0	0	0
	Executive	2	2	0	0	0
	Hourly FT	122	98	1	0	23

The crosstab definition is below.



This crosstab is now customized per the requirements below.

Requirement #1: % of Active Employees over 'Active' and 'Inactive' Employees as a separate Column.

Step1: Create a calculated column by choosing any pre-defined calculation – for example, sum of 'Active' and 'Inactive' column members. A new calculated column "Sum" is added adjacent to 'Inactive' member as shown in the second snapshot below.

Active	Inactive	Retired	Withdrawn
1,130			104
1			0
1			0
205			7
178			0
19			5
8			
893	1	10	
1	0	0	
2	0	0	

Active	Inactive	Sum
1,130	1	1,131
1	0	1
1	0	1
205	0	205
178	0	178
19	0	19
8	0	8
893	1	894
1	0	1
2	0	2
98	1	99
39	0	39
1	0	1

Step 2: In the calculation formula for this new member, use this formula:

```

if
    GridValueAt(CurrentRowIndex, GetColumnPathIndexOf("Active"),
    CurrentSummaryIndex)
    + GridValueAt(CurrentRowIndex, GetColumnPathIndexOf("Inactive"),
    CurrentSummaryIndex)=0
then
    0
else
    GridValueAt(CurrentRowIndex, GetColumnPathIndexOf("Active"),
    CurrentSummaryIndex)
    /
    (
    GridValueAt(CurrentRowIndex, GetColumnPathIndexOf("Active"),
    CurrentSummaryIndex)
    + GridValueAt(CurrentRowIndex, GetColumnPathIndexOf("Inactive"),
    CurrentSummaryIndex)
    ) * 100

```

Note: a. For each row (using currentrowindex), the GetColumnPathIndexOf() function gets the column index of the "Active" and "Inactive" column members. The GridValueAt() functions uses these cell index references and gets the corresponding cell values and applies the % formula on them.

b. Any complex mathematical formula can be used in calculation. As a specific example, any new measure can be specified here as a formula. In that case, a new measure appears a column. On a contrast, if this measure was added as a second measure to crosstab definition, it would appear as second measure in all the cells (summary index =1).

Step3: In the header formula, specify “% of Active”. The crosstab now appears as shown.

		Total	Active	Inactive	% of Active	Retiree
Total		1,250	1,130	1	100	15
Contract	Total	1	1	0	100	0
	Contract >= 1 Year	1	1	0	100	0
Non-Employee	Total	212	205	0	100	0
	Applicant	178	178	0	100	0
	Contractor	24	19	0	100	0
	Executive	10	8	0	100	0
Regular	Total	1,000	893	1	100	10
	Casual Hourly	1	1	0	100	0
	Executive	2	2	0	100	0
	Hourly FT	122	98	1	99	0
	Hourly PT	42	39	0	100	0
	Hourly PT (<20 Hr)	1	1	0	100	0
	Hourly PT (>= 20 Hr)	5	5	0	100	0

Requirement #2: Highlight the ‘Inactive’ column values in Red.

Step1: Selecting the cell values, click on format element on right click menu and use the following formula in the cell font:

```

if GridRowColumnValue("Z0EMPLOYEE_Q002.Employment Status")="Inactive"
then crRed
else crBlack
    
```

Note: a. The GridRowColumnValue() returns the labels for ‘Employment Status’ column group for each cell. The cells are marked in Red when this value is equal to “Inactive”.

b. In any Grid formula whenever a Row/Column Name is referenced, do not include the default curly parenthesis “{ }” in the row/column name.

c. GridLabelAt() function can also be used in place of GridRowColumnValue()

This conditionally formats the cell values in the “Inactive” member column as shown below. Without this formula all the cells would be formatted in red.

		Total	Active	Inactive	% of Active	Retiree
Total		1,250	1,130	1	100	15
Contract	Total	1	1	0	100	0
	Contract >= 1 Year	1	1	0	100	0
Non-Employee	Total	212	205	0	100	0
	Applicant	178	178	0	100	0
	Contractor	24	19	0	100	0
	Executive	10	8	0	100	0
Regular	Total	1,000	893	1	100	10
	Casual Hourly	1	1	0	100	0
	Executive	2	2	0	100	0
	Hourly FT	122	98	1	99	0
	Hourly PT	42	39	0	100	0
	Hourly PT (<20 Hr)	1	1	0	100	0
	Hourly PT (>= 20 Hr)	5	5	0	100	0

Requirement #3: For each “Employee Subgroup”, there should be a Summary row that displays the % of total employees in “Employee Subgroup” to the overall total employees.

Step 1: As explained in requirement #1, add a new row below “Executive”. This time add a blank row.

Step 2: Accessing the “Advanced Options” from ‘Edit Crosstab’ right-click menu option, use the following formula as the calculation formula for the blank row:

```
GridValueAt (GetRowPathIndexOf (GridRowColumnValue ("Z0EMPLOYEE_Q002.Employee Group" ,
currentrowindex)), CurrentColumnIndex, CurrentSummaryIndex)
/
GetTotalValueFor ("Z0EMPLOYEE_Q002.Employee Group") *100
```

Note: a. For each cell in the specified calculated row (referenced by currentrowindex), the GetTotalValueFor() function returns the Grand Total value which becomes the denominator. This gets the total value for all the values across Employee Group field.

b. GridRowColumnValue() function returns the ‘Employee Group’ row label (namely “Non-Employee”) for the current row. Subsequently, it becomes GetRowPathIndexOf(“Non-Employee”) which gives the rowindex of the Total’s row for ‘Non-Employee’. The GridValueAt() function then uses rowindex to get the corresponding total value for the Employee Subgroup, which becomes the numerator.

c. We could have hardcoded the value “Non-Employee” instead of using GridRowColumnValue() function. But, interestingly, the latter makes the function dynamic and, as shall follow soon, it will allow us to automatically fetch corresponding “Employee Group” label for different “Employee Subgroup” members.

Step 3: Specify Header Formula as:

```
"% of "+ GridRowColumnValue ("Z0EMPLOYEE_Q002.Employee Group" , currentrowindex)
```

Note: This demonstrates how grid functions can be used for dynamic header.

Now the crosstab shows as follows. Note that the row appears only for ‘Non-Employee’ member.

		Total	Active	Inactive	Retiree	Withdrawn
Total		1,250.00	1,130.00	1.00	15.00	104.00
Contract	Total	1.00	1.00	0.00	0.00	0.00
	Contract >= 1 Year	1.00	1.00	0.00	0.00	0.00
Non-Employee	Total	212.00	205.00	0.00	0.00	7.00
	Applicant	178.00	178.00	0.00	0.00	0.00
	Contractor	24.00	19.00	0.00	0.00	5.00
	Executive	10.00	8.00	0.00	0.00	2.00
	% of Non-Employee	16.96	18.14	0.00	0.00	6.73
Regular	Total	1,000.00	893.00	1.00	10.00	96.00
	Casual Hourly	1.00	1.00	0.00	0.00	0.00
	Executive	2.00	2.00	0.00	0.00	0.00
	Hourly FT	122.00	98.00	1.00	0.00	23.00

Step 4: Alter the Insertion formula for this calculated column as:

GetRowGroupIndexOf(CurrentRowIndex) = 2

Note: a. The Insertion Formula evaluates for each row in the crosstab. The calculated member row is inserted after each row that satisfies this insertion formula for the calculated member. GetRowGroupIndexOf() function returns the row group index for each of the rows. Hence the calculated row appears after each level 2 row member because of the value 2 in the condition.

b. To restrict the condition that this row shows up only for Employee Group: Non-Employee, and Employee Subgroup: Executive, add this extra condition to the insertion formula, so that then the formula evaluates true only for the specific combination of level members

and GridRowColumnValue("Z0EMPLOYEE_Q002.Employee Group") = "Non-Employee"
 and GridRowColumnValue("Z0EMPLOYEE_Q002.Employee Subgroup") = "Executive"

As shown below, this calculated row appears after each leaf member in the row level 2 group.

		Total	Active	Inactive	Retiree	Withdrawn
Total		1,250.00	1,130.00	1.00	15.00	104.00
Contract	Total	1.00	1.00	0.00	0.00	0.00
	Contract >= 1 Year	1.00	1.00	0.00	0.00	0.00
	% of Contract	0.08	0.09	0.00	0.00	0.00
Non-Employee	Total	212.00	205.00	0.00	0.00	7.00
	Applicant	178.00	178.00	0.00	0.00	0.00
	% of Non-Employee	16.96	18.14	0.00	0.00	6.73
	Contractor	24.00	19.00	0.00	0.00	5.00
	% of Non-Employee	16.96	18.14	0.00	0.00	6.73
	Executive	10.00	8.00	0.00	0.00	2.00
	% of Non-Employee	16.96	18.14	0.00	0.00	6.73
Regular	Total	1,000.00	893.00	1.00	10.00	96.00
	Casual Hourly	1.00	1.00	0.00	0.00	0.00
	% of Regular	80.00	79.03	100.00	66.67	92.31
	Executive	2.00	2.00	0.00	0.00	0.00
	% of Regular	80.00	79.03	100.00	66.67	92.31

Step 5: Change the Insertion formula as

`GetRowGroupIndexOf(CurrentRowIndex) = 1`

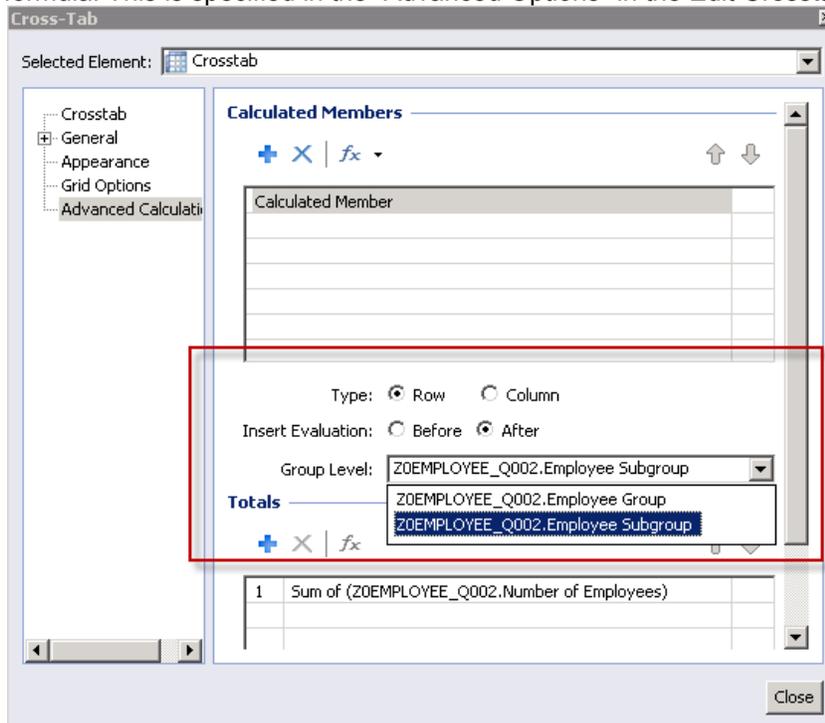
Now the calculated field appears as once per each “Employee Group” member.

		Total	Active	Inactive	Retiree	Withdrawn
Total		1,250.00	1,130.00	1.00	15.00	104.00
Contract	Total	1.00	1.00	0.00	0.00	0.00
	Contract >= 1 Year	1.00	1.00	0.00	0.00	0.00
	% of Contract	0.08	0.09	0.00	0.00	0.00
Non-Employee	Total	212.00	205.00	0.00	0.00	7.00
	Applicant	178.00	178.00	0.00	0.00	0.00
	Contractor	24.00	19.00	0.00	0.00	5.00
	Executive	10.00	8.00	0.00	0.00	2.00
	% of Non-Employee	16.96	18.14	0.00	0.00	6.73
Regular	Total	1,000.00	893.00	1.00	10.00	96.00
	Casual Hourly	1.00	1.00	0.00	0.00	0.00
	Executive	2.00	2.00	0.00	0.00	0.00
	Hourly FT	122.00	98.00	1.00	0.00	23.00

Note: As a rule of thumb:

- a. When a calculated member is repeated for each member for level 'n', the insertion formula should be: `GetRowGroupIndexOf(CurrentRowIndex) = n`
- b. When a calculated member is repeated once for all members for level 'n', the insertion formula should be: `GetRowGroupIndexOf(CurrentRowIndex) = n-1`

The location of the inserted member also depends on when at what level is the insertion formula evaluated and whether the new calculated row(s) is(are) inserted Before or After the row(s) that satisfies insertion formula. This is specified in the “Advanced Options” in the Edit Crosstab window as shown below.



Troubleshooting

Below are some of the common syntax errors while using the grid functions.

- **<row/column name> is not a valid row or column name**

This usually occurs when the expected argument is a row/column name. Re-check the row/column name to ensure that the spelling is correct. Do not enclose the row/column name in curly parenthesis { }. Make sure to enclose the row/column name in double quotes “ ”.

- **The remaining text does not appear to be part of formula**

This usually appears if extra arguments are provided. Verify the arguments provided.

- **The function call does not match any overloads of <function name>**

This error appears when arguments provided do not match in type or number to the arguments expected.

Related Content

[Crystal Reports: Crosstabs in Crystal 2008](#)

[Crystal Reports: Calculate Inside a Crosstab](#)

[Crystal Report for Enterprise - User Guide](#)

For more information, visit the [Business Objects homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.