Hyperion System 9 BI+ Financial Reporting:

Tips and Tricks for Designing Advanced Reports

Emily Stinaff
March 1, 2008

Quality financial planning, budgeting and consolidation applications call for quality report generation to complete the picture of a comprehensive and valuable enterprise performance management solution. Hyperion Financial Reporting, a module of the Oracle Business Intelligence Suite Enterprise Edition Plus, is a tool that transforms enterprise data into operational insights with highly-formatted, book-quality report creation ability. The task of creating outputs ranging from internal management reports to external SEC packages is facilitated though an intuitive, object-oriented interface that even the most non-tech savvy users can employ.

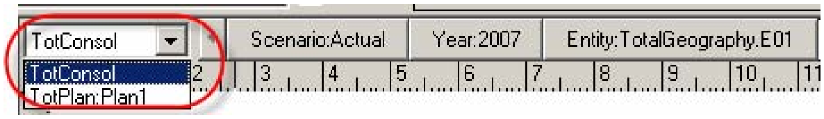
Often, a basic financial report can take no more than a matter of minutes to build in Hyperion Financial Reporting (FR). At times, however, the creation of a seemingly standard financial report can be a deceptively challenging task. With the wide array of features and functions available in FR, it can simply be a matter of becoming familiar with knowing how to best use the tools at your disposal to effectively create the desired end result. This paper is designed to give experienced report designers real-world examples of a few reporting design topics, certain challenges that can arise, and suggested workarounds to resolve them.

REPORT 1: Multiple Grids, Multiple Data Connections, Cross-Grid Formulas, Conditional Formatting

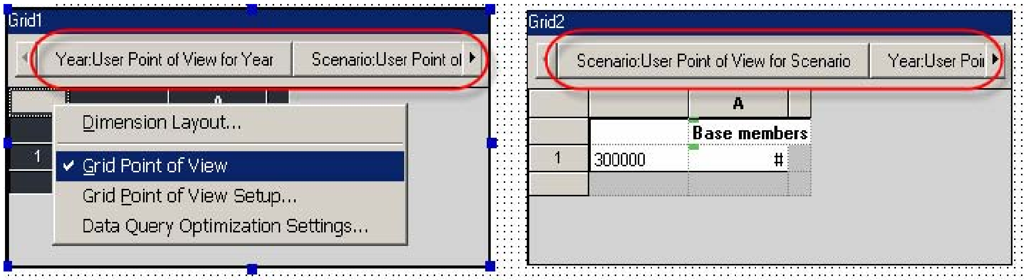
The following report is an example of an internal variance report that clearly displays any data discrepancies; in this case, between Hyperion Financial Management and Hyperion Planning. The application difference here need not apply – the grids could all be pulling from the same product or same database connection.

Multiple Grids & Data Connections (Review & Considerations)

When creating a report with multiple grids that have different database connections, multiple user Points of View (POVs) will result. The drop-down that displays at the top left allows users to toggle among the connections and change the POV selections as needed, but it is best to reduce/eliminate this need as much as possible.



Adding a grid POV with a locked selection (where feasible) to each grid in the report alleviates the burden on the end user to make the appropriate member selections for each POV. Any dimensions that need to be dynamic will have 'User Point of View for X' as the member selection. Dimensions with this dynamic selection will remain in the top user POV bar, while selected dimensions will be removed.


Grid1
Year:User Point of View for Year
Scenario:User Point of
Dimension Layout...
✓ Grid Point of View
Grid Point of View Setup...
Data Query Optimization Settings...
1
Grid2
Scenario:User Point of View for Scenario
Year:User Poi
A
Base members
1
300000
#

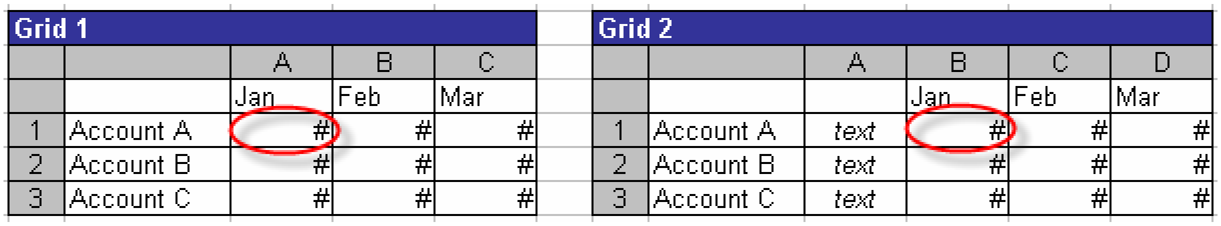
In this example report, the only dimensions required to remain dynamic are Year, Scenario and Entity. For both grids, these dimensions have 'User Point of View for X' selected. Since this is an application variance report, all three dimensions must match between the two connections in order for the report to make sense. This responsibility falls on the user. While alternatives such as placing the dynamic dimensions in the page dimension of the report and utilizing prompts exist, such alternatives can prove just as, if not more, cumbersome.

Note: To ensure proper use of such a report, it is vital that users are trained in such a way that they realize the presence of, and understand to the need to, verify the multiple POVs. A text box with instructions may prove beneficial as well.

Cross-Grid Formulae (Review and Considerations)

A formula referencing multiple grids is similar to a single-grid formula with added specificity (as one might expect) for a particular grid. For example, whereas the formula "Row[1] – Row[2]" would be a functional formula in a single grid, "Grid1.Row[1] – Grid2.Row[2]" would be the necessary syntax when applying this formula across grids (if the formula is actually contained in Grid 1 or 2, the corresponding argument can be removed). An important restriction to keep in mind: when utilizing constant row references (as in the previous example), the columns must correspond; when utilizing constant column references, the rows must correspond. Inserting text or formula rows/columns can change this alignment.

Given the present example, this would mean that the data represented by intersection [A,1] in Grid 1 must mirror that of the data represented by intersection [A,1] in Grid 2 (and, if placing the formula in a third grid, the columns in Grid 3 must also match). If, for example, Grid 1 and Grid 2 are the same except Grid 2 has a text column inserted after the account dimension, the simple version of the formula would not hold:



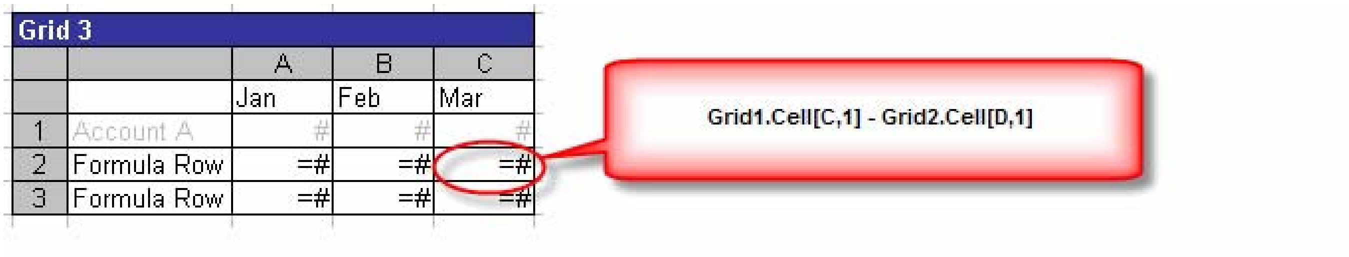
Grid 1

		A	B	C
		Jan	Feb	Mar
1	Account A	#	#	#
2	Account B	#	#	#
3	Account C	#	#	#

Grid 2

		A	B	C	D
			Jan	Feb	Mar
1	Account A	text	#	#	#
2	Account B	text	#	#	#
3	Account C	text	#	#	#

When this is the case, an individual formula needs to be developed for each cell. This depicts a grid used for cross-grid formula purposes only (so row 1, the requisite data row, is hidden):

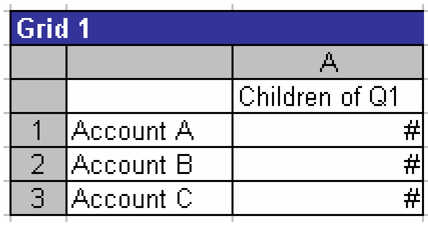


Grid 3

		A	B	C
		Jan	Feb	Mar
1	Account A	#	#	#
2	Formula Row	=#	=#	=#
3	Formula Row	=#	=#	=#

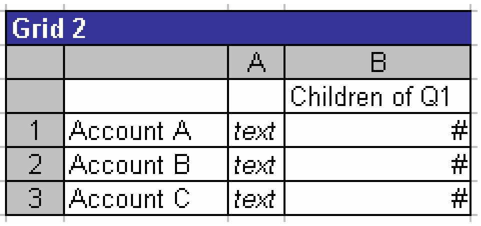
Grid1.Cell[C,1] - Grid2.Cell[D,1]

If a "fat" column selection (such as children of Q1) is used in Grid 1 and/or Grid 2 (still assuming the misalignment), Grid 3's columns would need to remain a "thin" selection (Jan, Feb, Mar – separately) and the formula would need to include sub-cell arguments:



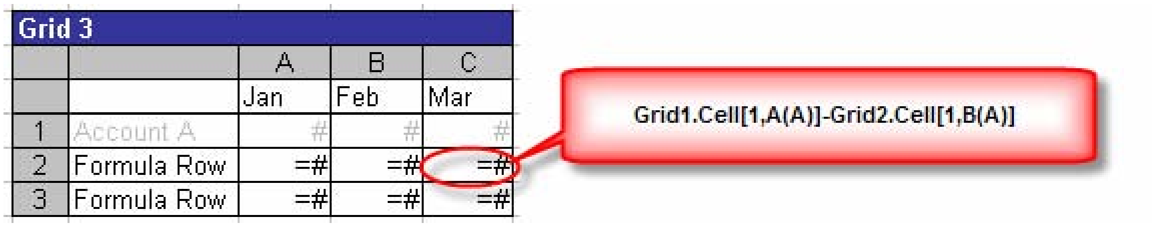
Grid 1

		A
		Children of Q1
1	Account A	#
2	Account B	#
3	Account C	#



Grid 2

		A	B
			Children of Q1
1	Account A	*text*	#
2	Account B	*text*	#
3	Account C	*text*	#



Grid 3

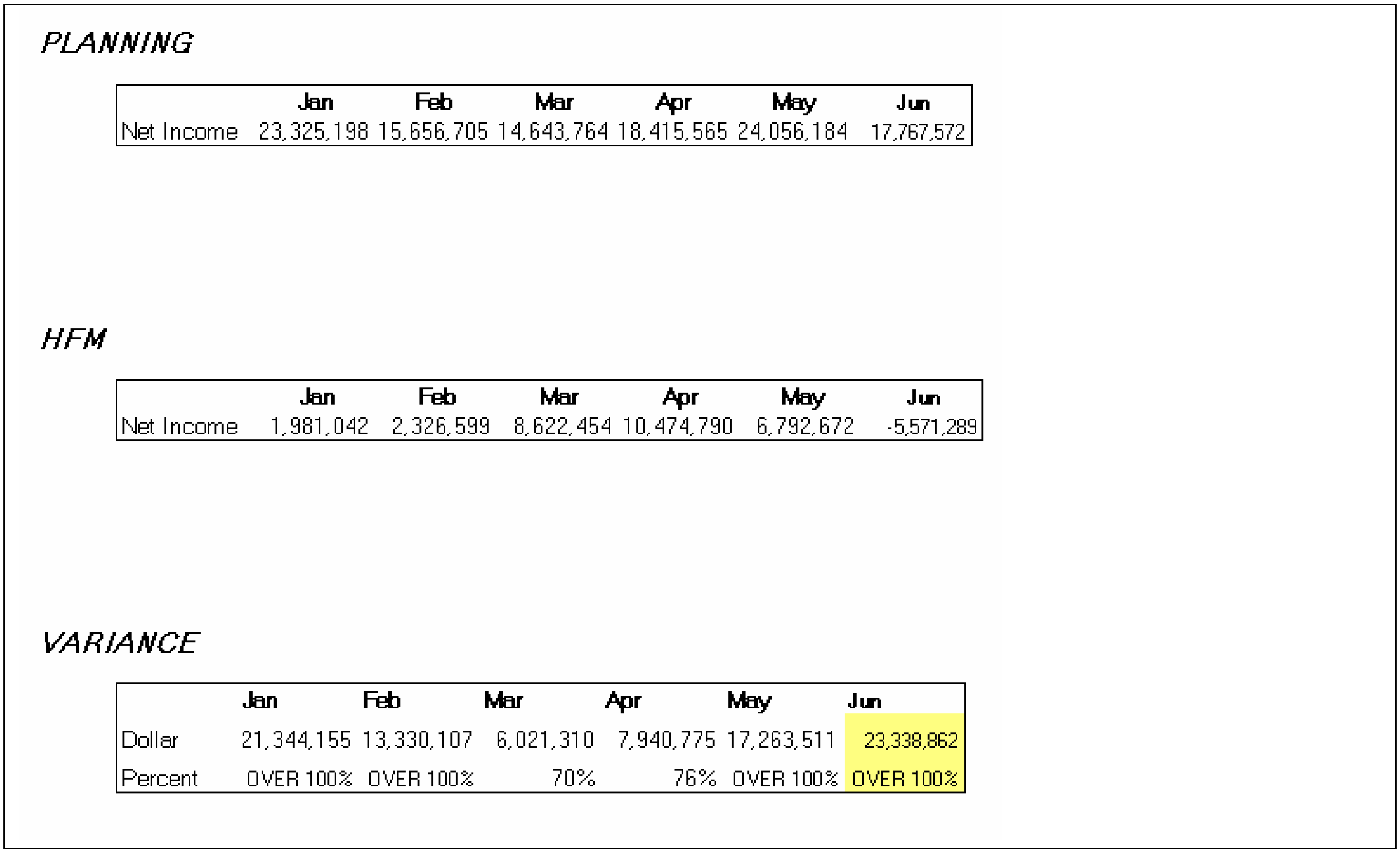
		A	B	C
		Jan	Feb	Mar
1	Account A	#	#	#
2	Formula Row	=#	=#	=#
3	Formula Row	=#	=#	=#

Grid1.Cell[1,A(A)]-Grid2.Cell[1,B(A)]

Such a setup would also be necessary if Grid 1 and Grid 2 have the same layout, but Grid 3 has an inserted column that throws off the alignment.

Conditional Formatting (Tips and Considerations)

Assume that, incorporating the multiple grid design items as reviewed, the final result of the report you are working on is to look like this:



PLANNING

	Jan	Feb	Mar	Apr	May	Jun
Net Income	23,325,198	15,656,705	14,643,764	18,415,565	24,056,184	17,767,572

HFM

	Jan	Feb	Mar	Apr	May	Jun
Net Income	1,981,042	2,326,599	8,622,454	10,474,790	6,792,672	-5,571,289

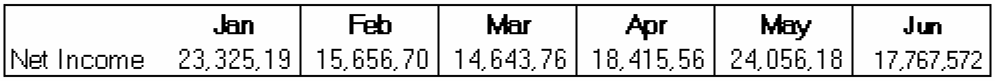
VARIANCE

	Jan	Feb	Mar	Apr	May	Jun
Dollar	21,344,155	13,330,107	6,021,310	7,940,775	17,263,511	23,338,862
Percent	OVER 100%	OVER 100%	70%	76%	OVER 100%	OVER 100%

- Three grids: One connected to the Planning database, one to HFM and a third solely for the purpose of creating cross-grid formulae (the db connection is essentially irrelevant).

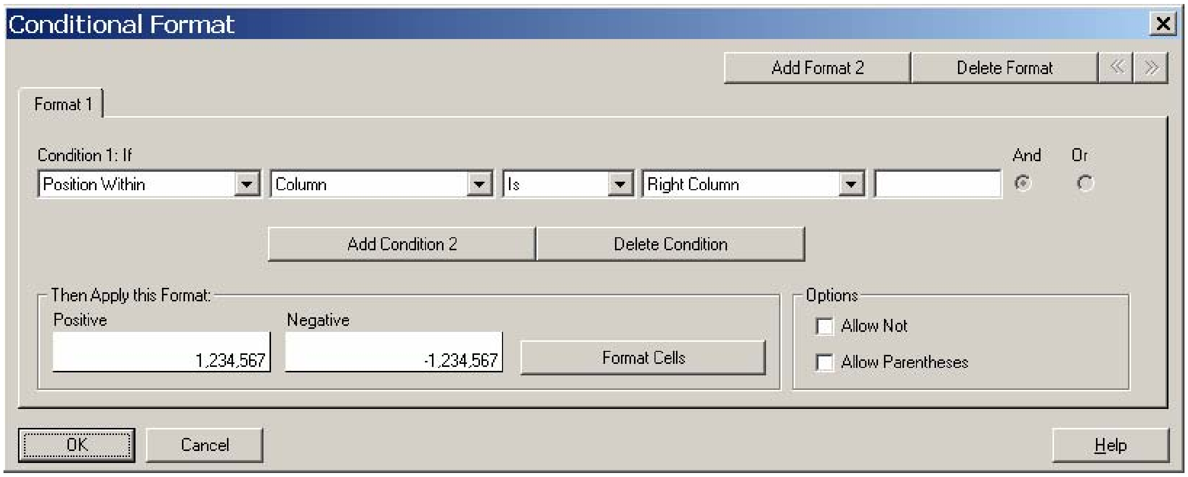
- The period columns are based upon the base members of the Period dimensions, allowing for the potential display of all 12 months ("fat" selection, meaning a single column in the studio expands to two or more columns upon view). However, a grid suppression setting is used (to hide all future, unpopulated periods); alternatively, a range function (through the current period) can be put in place as the member selection.

After creating the first two grids and making the appropriate member selections, a relatively small detail becomes an issue: the border. Since the columns are dynamically displayed via a "fat" selection, a hard-coded right border on the period column would result in:



	Jan	Feb	Mar	Apr	May	Jun
Net Income	23,325,19	15,656,70	14,643,76	18,415,56	24,056,18	17,767,572

If the individual column borders are not desired, conditional formatting may be applied. Using the "position within" condition, place a right-border on the right column.



Conditional Format

Format 1

Condition 1: If

Position Within | Column | Is | Right Column | And ● Or ○

Then Apply this Format:

Positive: 1,234,567

Negative: -1,234,567

Options

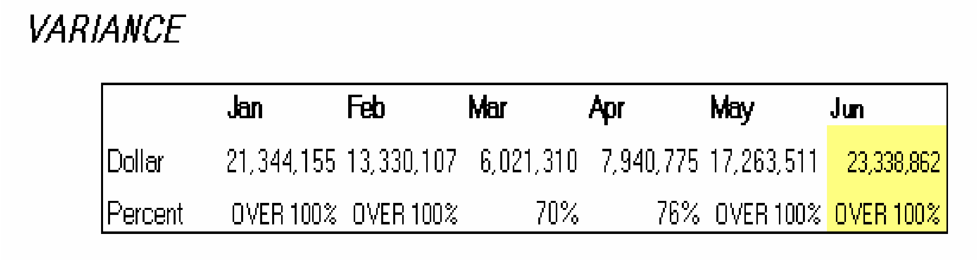
☐ Allow Not

☐ Allow Parentheses

*Keep in mind that any previously-applied formatting will carry through to the conditional formatting dialog, and is relevant when applying a conditional format against two or more cells with different underlying formats.

Another option: Insert a text column to the right of the existing data column, simply to hold a left border.

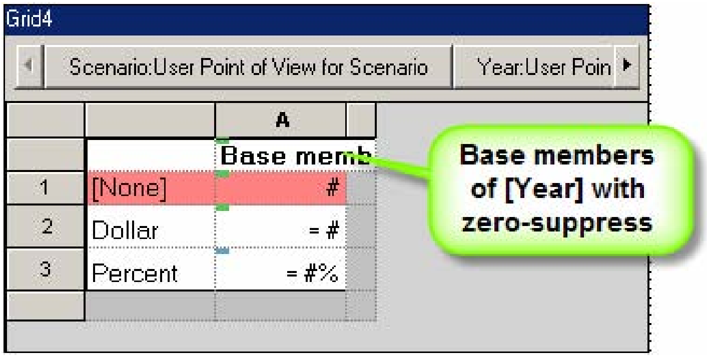
Now, looking at the Variance section of the report, design items include:



VARIANCE

	Jan	Feb	Mar	Apr	May	Jun
Dollar	21,344,155	13,330,107	6,021,310	7,940,775	17,263,511	23,338,862
Percent	OVER 100%	OVER 100%	70%	76%	OVER 100%	OVER 100%

- Two formula rows (as described in previous section); data row is hidden.
- Any percent variance exceeding 100% should not display the actual value, rather “OVER 100%” (note: mathematical conditional statements do not support text arguments).
- The last period on the report (the current period) is always highlighted in yellow.



Grid4

Scenario:User Point of View for Scenario | Year:User Poin

		A
		Base memb
1	[None]	#
2	Dollar	= #
3	Percent	= #%

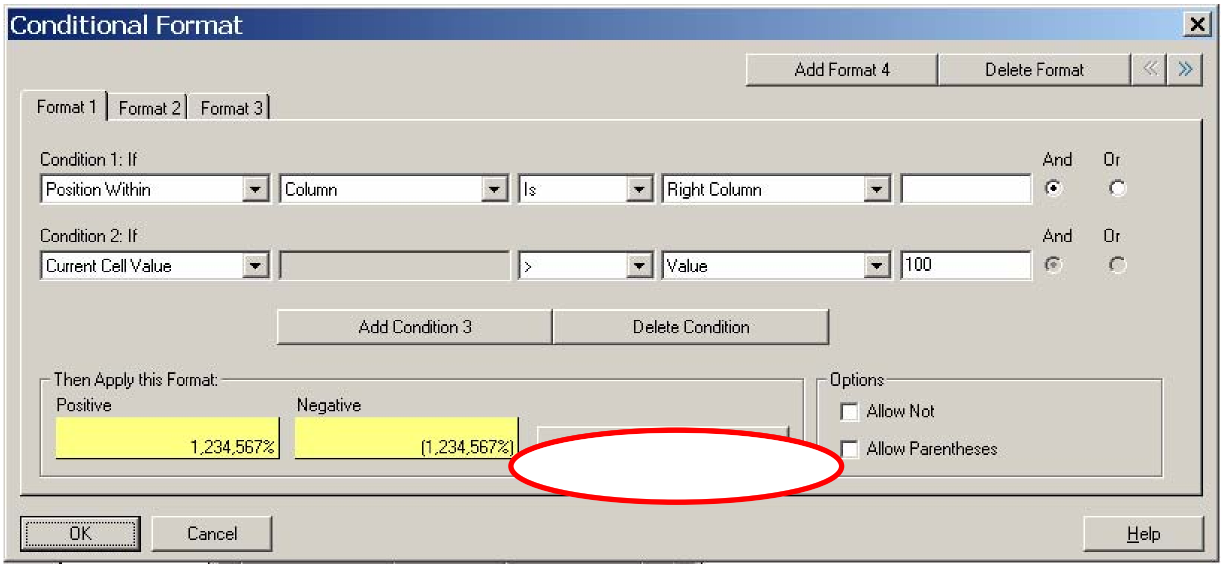
Base members of [Year] with zero-suppress

To create the right border, a conditional format can be applied as stated above, or a new (blank) text column can be inserted to the right of column A and formatted with a left border.

Cells A,2 and A,3 need additional conditional formatting. Although they share in common the need to be highlighted yellow and given a right border if the position within the column is the right column, do NOT apply this format to these two cells at the same time. The underlying numerical format is different (dollars vs. percent), and the conditional formatting dialog will be forced to pick one.

Instead, cell A,2 can be given the singular condition using position within resulting in a yellow highlight and a right border. Cell A,3, however, needs a compound statement. Not only does it need the condition of the yellow highlight and right border for the last column, it also needs any value exceeding 100 to read as "OVER 100%". With these two separate conditions, you must also account for the possibility of both conditions occurring. Conditional formatting is not incremental; if a condition occurs and a conditional format is found, the process ends. It does not continue to search through the other conditional formats to see if they may apply as well.

Therefore, 3 separate format tabs are needed, with the most specific (the combination of "OVER 100%" and the yellow highlight/right border) named first:



Conditional Format

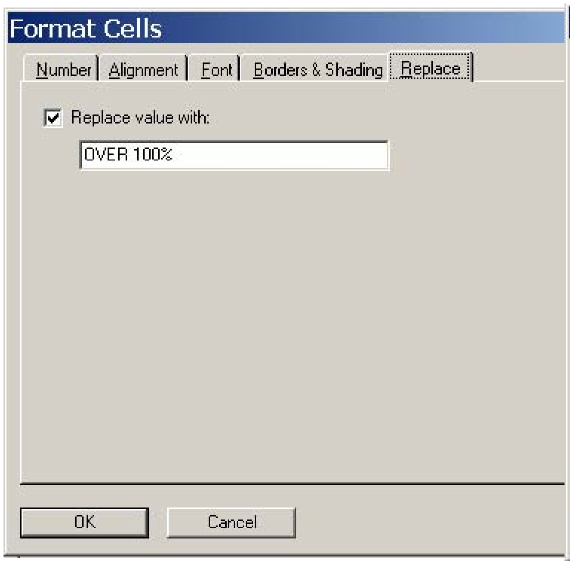
Format 1 | Format 2 | Format 3

Condition 1: If Position Within | Column | Is | Right Column | And ● Or ○

Condition 2: If Current Cell Value | > | Value | 100 | And ● Or ○

Then Apply this Format:
Positive: 1,234,567%
Negative: (1,234,567%)

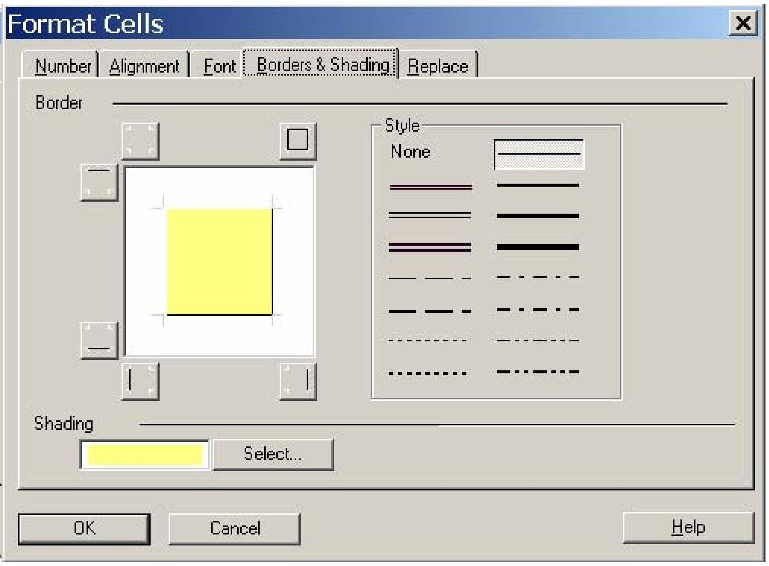
Options
☐ Allow Not
☐ Allow Parentheses



Format Cells

Number | Alignment | Font | Borders & Shading | Replace

☑ Replace value with: OVER 100%



Format Cells

Number | Alignment | Font | Borders & Shading | Replace

Border

Style
None

Shading

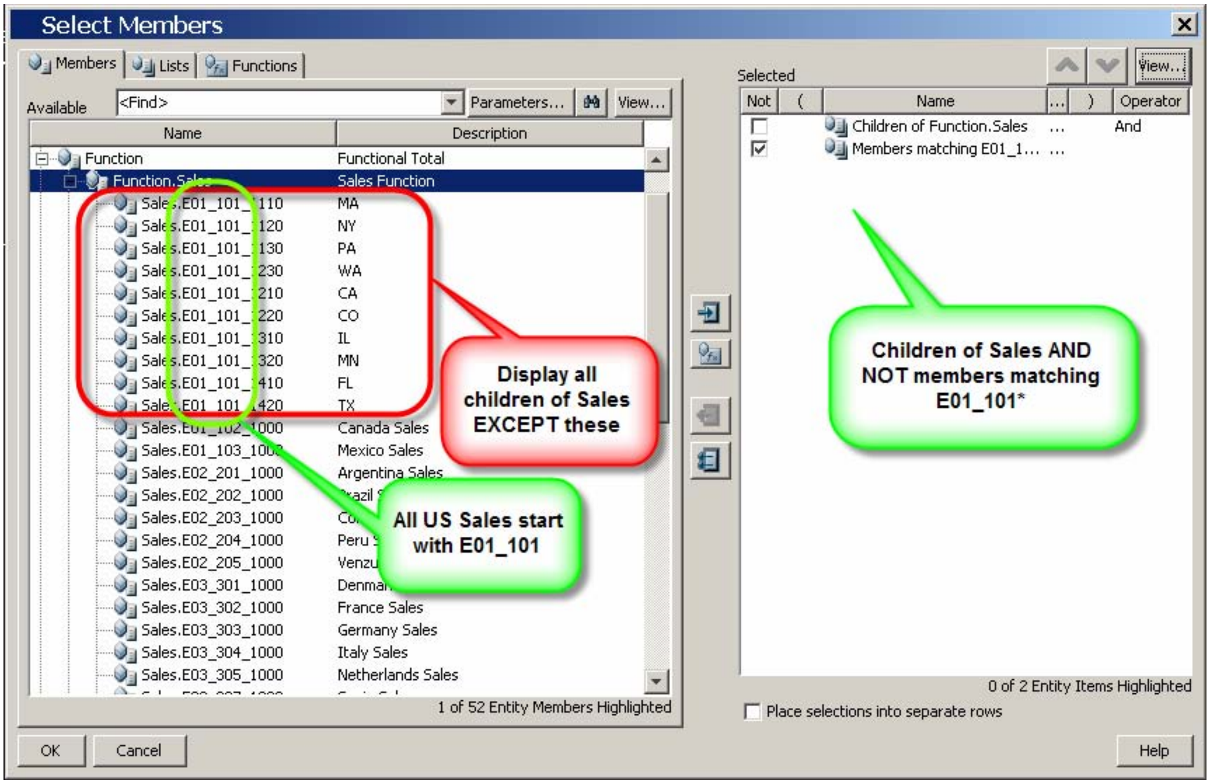
REPORT 2: Conditional Suppression to Overcome Range Limitations, Conditional Formatting to Overcome Suppression Limitations and Advanced Member Selection

The next example is a report that needed a few workarounds to achieve the desired functionality. A common challenge encountered with the design of a report is creating dynamic selections based on structures or hierarchies that don't exist. If the application metadata structure is not to be changed, another solution is needed. Another frequent issue that comes up is the apparent limitations of the range function when trying to populate the columns of a report with a dynamic period selection. And finally, this report will give an example of how to display whether or not the current unit's calculation status (HFM) is "acceptable" (and only display when it is not).

Advanced Member Selection (Tips and Considerations)

The rows of this report contain the entity dimension. The selection of the members is a bit convoluted due to the existing structure of the dimension. This report is to be an international sales and other report, with sales from each entity outside the US is listed first, followed by all other functions.

In this application, the Functional rollup does not separate sales by country and the Legal rollup does not separate by function. The member selection for the first item, international sales, can be assigned in this manner:



Name	Description
Function	Functional Total
Function.Sales	Sales Function
Sales.E01_101_1110	MA
Sales.E01_101_1120	NY
Sales.E01_101_1130	PA
Sales.E01_101_1230	WA
Sales.E01_101_1210	CA
Sales.E01_101_1220	CO
Sales.E01_101_1310	IL
Sales.E01_101_1320	MN
Sales.E01_101_1410	FL
Sales.E01_101_1420	TX
Sales.E01_102_1000	Canada Sales
Sales.E01_103_1000	Mexico Sales
Sales.E02_201_1000	Argentina Sales
Sales.E02_202_1000	razil
Sales.E02_203_1000	Co
Sales.E02_204_1000	Peru
Sales.E02_205_1000	Venzu
Sales.E03_301_1000	Denma
Sales.E03_302_1000	France Sales
Sales.E03_303_1000	Germany Sales
Sales.E03_304_1000	Italy Sales
Sales.E03_305_1000	Netherlands Sales

1 of 52 Entity Members Highlighted

Not	(	Name	...	)	Operator
☐		Children of Function.Sales	...		And
☑		Members matching E01_1...	...		

0 of 2 Entity Items Highlighted

☐ Place selections into separate rows

Display all children of Sales EXCEPT these

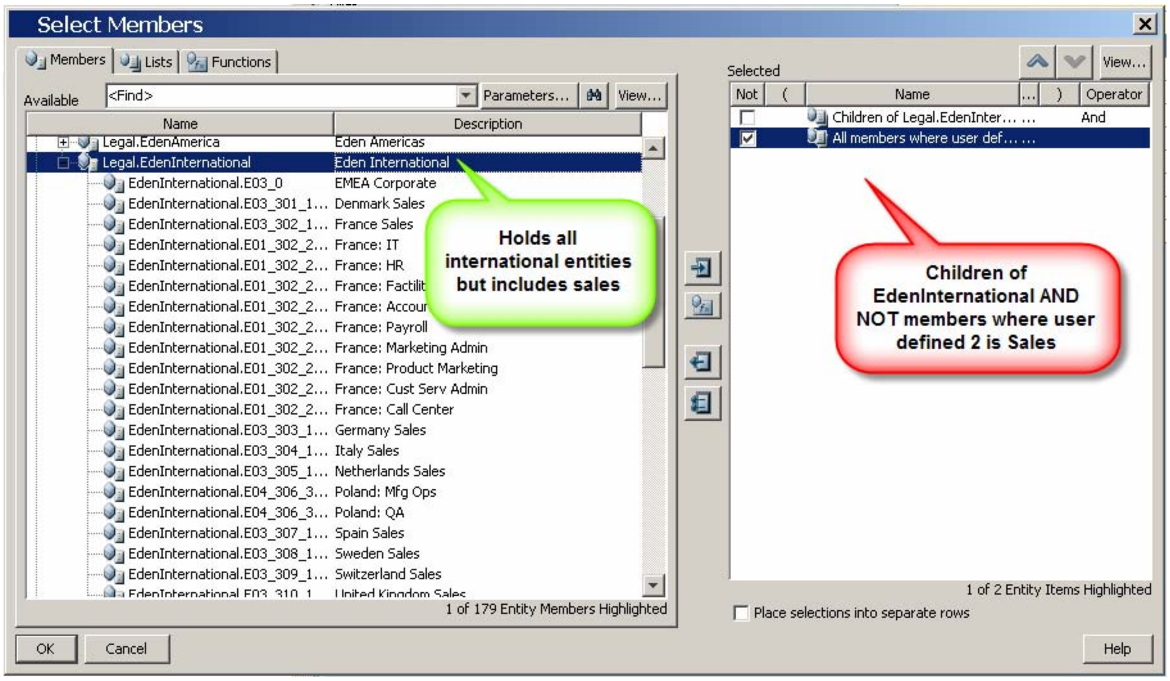
All US Sales start with E01_101

Children of Sales AND NOT members matching E01_101*

This is an advanced member selecting using a familial relationship and excluding a match function. Although the US sales entities are contained under a common parent under a different hierarchy, cross-hierarchical exclusions are not interpreted. Using the match function to specify all entities starting with E01_101*, all current and future US sales entities can be dynamically excluded, assuming the current naming convention is consistently applied.

The next selection needs to include all other functions (besides sales) for all international entities. The Legal consolidation structure gives us a breakdown between domestic and international, but includes sales. Sales are not under a common parent and the naming convention here does not allow for the efficient use of a match function. If a structural metadata change truly cannot be made here, the use of a user defined field (in HFM) or a UDA (in Planning) should be considered. If the sales accounts can be uniquely tagged, that tag can then be used in the exclusion criteria for this member selection.

In this example, the user defined field 2 in HFM has been utilized to tag each sales account with "Sales".



Name	Description
Legal.EdenAmerica	Eden Americas
Legal.EdenInternational	Eden International
EdenInternational.E03_0	EMEA Corporate
EdenInternational.E03_301_1...	Denmark Sales
EdenInternational.E03_302_1...	France Sales
EdenInternational.E01_302_2...	France: IT
EdenInternational.E01_302_2...	France: HR
EdenInternational.E01_302_2...	France: Factilit
EdenInternational.E01_302_2...	France: Accour
EdenInternational.E01_302_2...	France: Payroll
EdenInternational.E01_302_2...	France: Marketing Admin
EdenInternational.E01_302_2...	France: Product Marketing
EdenInternational.E01_302_2...	France: Cust Serv Admin
EdenInternational.E01_302_2...	France: Call Center
EdenInternational.E03_303_1...	Germany Sales
EdenInternational.E03_304_1...	Italy Sales
EdenInternational.E03_305_1...	Netherlands Sales
EdenInternational.E04_306_3...	Poland: Mfg Ops
EdenInternational.E04_306_3...	Poland: QA
EdenInternational.E03_307_1...	Spain Sales
EdenInternational.E03_308_1...	Sweden Sales
EdenInternational.E03_309_1...	Switzerland Sales
EdenInternational.E03_310_1	United Kingdom Sales

1 of 179 Entity Members Highlighted

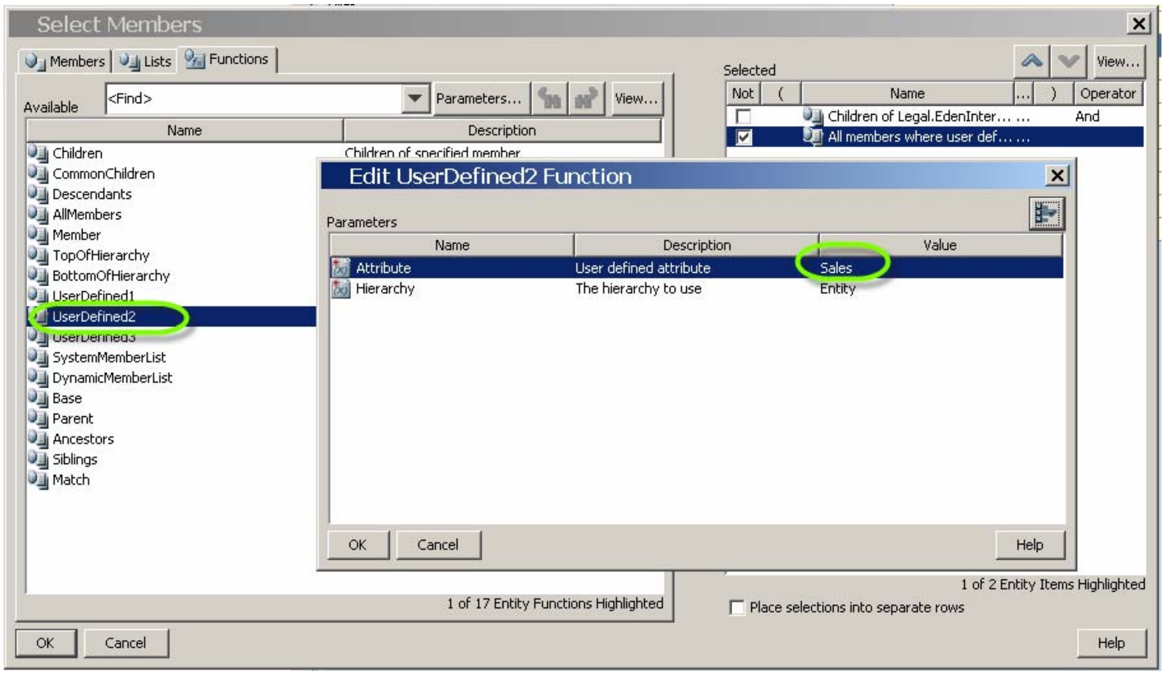
Not	(	Name	...	)	Operator
□		Children of Legal.EdenInter...	...		And
☑		All members where user def...	...		

1 of 2 Entity Items Highlighted

□ Place selections into separate rows

Holds all international entities but includes sales

Children of EdenInternational AND NOT members where user defined 2 is Sales



Name	Description
Children	Children of specified member
CommonChildren	
Descendants	
AllMembers	
Member	
TopOfHierarchy	
BottomOfHierarchy	
UserDefined1	
UserDefined2	
UserDefined3	
SystemMemberList	
DynamicMemberList	
Base	
Parent	
Ancestors	
Siblings	
Match	

1 of 17 Entity Functions Highlighted

Not	(	Name	...	)	Operator
□		Children of Legal.EdenInter...	...		And
☑		All members where user def...	...		

1 of 2 Entity Items Highlighted

□ Place selections into separate rows

Name	Description	Value
Attribute	User defined attribute	Sales
Hierarchy	The hierarchy to use	Entity

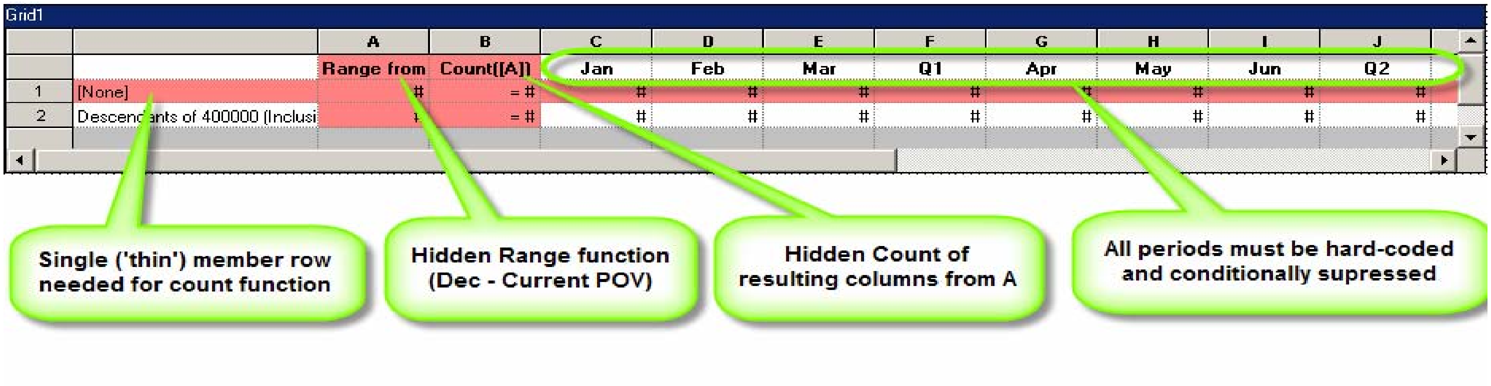
Range Function Limitations (Workaround)

The range function available within HR allows for the dynamic display of time periods, and can be particularly useful when needing to show actual data through the current period and plan or budget data beyond the current period. However, there are a couple of important limitations to be aware of when working with the range function:

- If an 'MMM Q' display through the current period is desired, the range function will not suffice. A range can only work with one level of the time period hierarchy.
- If the ability to select the last period in the year (using a first period – current POV and current POV + 1 – last period setup) without having the range reiterate is needed, a workaround is necessary.

Both of these issues can be solved using a hidden 'count' column that will essentially depict the current period and allow for conditional suppression based upon this information.

Given the first limitation with the MMM Q display, all periods must be "hard-coded" in the columns since the range function will not work. Assume that this application is based on a calendar year. A range function column (Jan – current POV) is added to the report and hidden. A second hidden column holds the count function based upon the range (which will return 4 for April, 5 for May, etc.).



Grid1

		A	B	C	D	E	F	G	H	I	J
		Range from	Count([A])	Jan	Feb	Mar	Q1	Apr	May	Jun	Q2
1	[None]	#	= #	#	#	#	#	#	#	#	#
2	Descendants of 400000 (Inclusi	#	= #	#	#	#	#	#	#	#	#

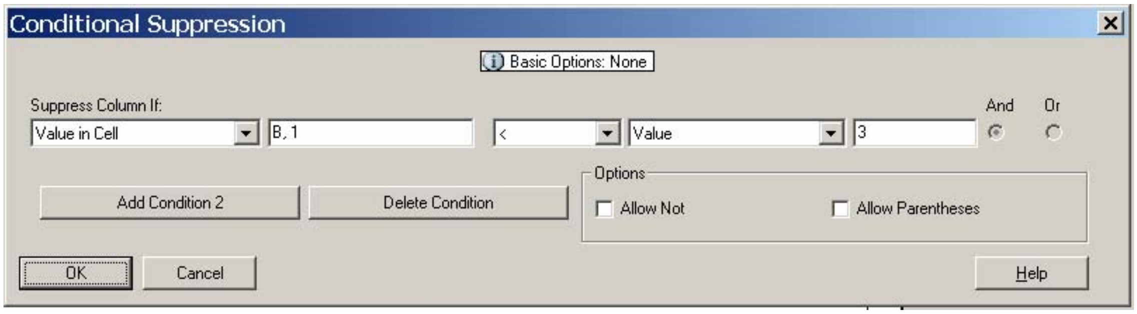
Single ('thin') member row needed for count function

Hidden Range function (Dec - Current POV)

Hidden Count of resulting columns from A

All periods must be hard-coded and conditionally supressed

Each period column is assigned a unique conditional suppression function according to the particular month. This is the suppression that March and Q1 would have:

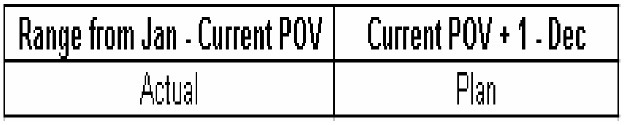


Conditional Suppression

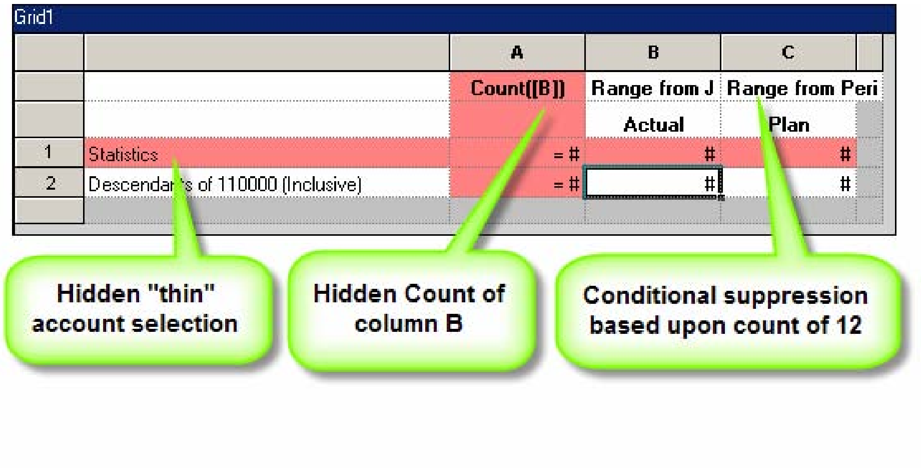
Basic Options: None

Suppress Column If: Value in Cell | B, 1 | < | Value | 3 | And ● Or ○

The second limitation of the range reiteration occurs when the currently selected period is the last period in the year. The second range reiterates (typically the budget or plan scenario), resulting in 12 months of actual and 12 months of budget data. To avoid this result, the count workaround can again be utilized to suppress the second range if the count function = 12.



Range from Jan - Current POV	Current POV + 1 - Dec
Actual	Plan



Grid1

		A	B	C
		Count([B])	Range from J	Range from Peri
			Actual	Plan
1	Statistics	= #	#	#
2	Descendants of 110000 (Inclusive)	= #	#	#

Hidden "thin" account selection

Hidden Count of column B

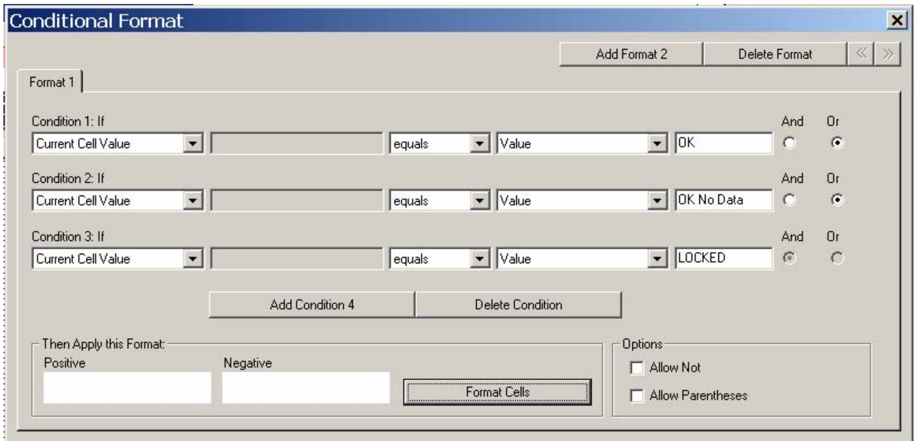
Conditional suppression based upon count of 12

Note: In both of these examples, the hidden thin row selection is only necessary if currently only fat selections are being used. If a thin row already exists in the report, no additional row needs to be added.

Conditional Suppression Limitation (Workaround)

Conditional suppression, as displayed by the previous section, can be an extremely useful function when designing advanced reports. The feature does not operate, however, on individual cells. If your report is to dynamically display the results of a text function, for example, only when it is relevant, conditional suppression is not the feature to use. Instead, a simple workaround of conditionally formatting the text of the cells to white (essentially an electronic white-out) when irrelevant can be used.

As an example, this particular report has a row of calcstatus (HFM) text functions to display the current calculation status (needs consolidation, needs calculation, etc.) of a data intersection. However, if the current status requires no attention (i.e., OK, Locked, etc.), the status should not appear. By specifying these exceptions in a conditional format dialog, the irrelevant results can be formatted with white text:



Conditional Format

Add Format 2 | Delete Format

Format 1

Condition 1: If — Current Cell Value — equals — Value — OK — And ○ Or ●

Condition 2: If — Current Cell Value — equals — Value — OK No Data — And ○ Or ●

Condition 3: If — Current Cell Value — equals — Value — LOCKED — And ● Or ○

Add Condition 4 | Delete Condition

Then Apply this Format:
Positive
Negative
Format Cells

Options
□ Allow Not
□ Allow Parentheses

In Conclusion

More elaborate and intricate reporting requirements need more innovative and detailed solutions. At times, a report design can become more convoluted than initially anticipated. Despite the inevitable limitations of any reporting system, Hyperion System 9 BI+ Financial Reports provides a framework with the flexibility to construct a wide range of outputs and cater to a diverse array of users. The next time you encounter a challenging report design issue, try to keep in mind all the tools at your disposal and how they may be combined and used in creative ways to form a solution.