



BI360 – Best Practices

Contents

Reports	2
Budgeting Templates.....	2
Line Item Details.....	4
Data Warehouse and SSIS	5
Form/Report Template Style	6
Project Management.....	6



Reports

Note that many of the items listed under reports should be used when building a budget template.

1. Remove all groupings/sorting that are not dimensions or that are not needed for grouping and sorting. This will have a big speed improvement.
2. Add amount <>0 and amount not null on the rows to limit the number of rows showing up. This will have a big speed improvement.
3. Use the “create sheet per value” rather than hardcoding tabs in the report.
4. Use account attributes to create row groupings (parent and child) – see white paper (BI360 White Paper - Using Attributes instead of Account Ranges.docx) for more details. This change will have, at minimum, a 70% increase.
5. Group periods using expanding groups and built-in period functions rather than having to hardcode periods.
6. Limit the filters to only dimensions and dimension descriptions (if used). Do not include every attribute.
7. Remove drilldown from the amount fields if they are not needed. This can speed up reports by at least 30%.
8. Do not create completely separate reports, such as Balance Sheet and Income Statement, on multiple tabs in a report. Use the Publisher to combine these reports together during execution.
9. Test the reports using security prior to having end-users start running reports.
10. Use the drill-down functionality when there is an issue with an amount to determine the amounts that calculate that specific cell.
11. Delete blank rows and columns to ensure that there is no hidden information in the file.
12. Don't mix period functions with period formulas. As example, don't have one column use the period functions while another column references a period formula such as {@Period-1}.
13. Don't copy bitmap files from one BI360 Template to another.
14. If a dimension is going to be used across the entire report, then the dimension should be referenced at the sheet level.
15. If nested expanding groups are used and blank transactions should be excluded, then it is best practice to set the filter to include the amount value <> (not in) 0 and amount not in NULL.
16. Incrementally build a report and save after each increment has been tested. If troubleshooting, then reverse the process.
17. Use the Layout Comments feature to understand the structure of the report design for analysis or troubleshooting (available in versions 3.5 or newer).

Budgeting Templates

1. Know the dimensions, values and attributes that need storing prior to building the template.
2. Test the templates using security prior to having end-users start budgeting.
3. Use only one tab over having forms across multiple tabs. It will be faster to execute and easier to trouble shoot, etc.
 - a. Use assignments to group individual spreadsheets together, rather than using multiple tabs in a single workbook.



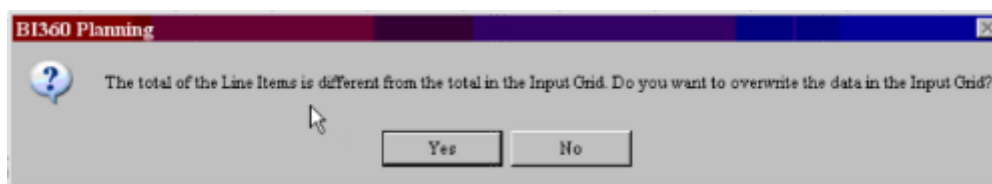
4. All attributes on a single transaction must be saved on the same form on a single row. As example, storing value 1 and value 2 to the same dimensions must be done on the same row to ensure proper storing.
 - a. If not, then you must use a different dimension code. If the dimension codes are the same in all dimensions, but on separate rows, then the second row will override the first row.
5. Try to limit the number of data grids. The more grids you have, then the harder it is to troubleshoot.
 - a. You can create a hidden section to create cleaner data settings.
 - b. If you need multiple grids, then check each one iteratively so that you ensure that all work individually.
6. If the dimension codes have leading zeros, then ensure that the value appears in the form and do not hard code into the SIM settings. Also, if the dimension code is being pulled, then use the text or parenthesis around the OSR dimension code formula to ensure that the leading zeros are not stripped.
7. Separate reports from budget input. They should not be on the same template or in the same Excel workbook as an input form.
8. If there is a need to copy an Excel file in order to create a new budget template, then complete this before creating the data settings. However, if the data settings have already been completed, then delete the "title" attribute file property because this contains a planning template ID.
9. Use the dimension "Category" to store the data. MAIN should be used for all GL transactions.
10. Storing type:
 - a. Store Changes: this should always work if you use a single tab. This will store much faster, but input may be slower. Store changes also allow an audit trail as the original information on who was responsible for the original transaction vs. the updated transaction. The force store will update both original and updated transactions.
 - b. Force Store: use if there are typically only calculations that need to be stored. Note the input will be fast, but the storing will be slower.
11. Use Excel data validation boxes if possible, as compared to manual input, to limit the selections a user can choose and to reduce the chances of input errors. Use the error alert to ensure that only the items in the drop-down box are selected.
 - a. Use indirect to get rid of blank rows.
 - b. Use indirect to link one data validation to another cell's value.
12. Disable the BI360 Planning input window if line-item detail and spreading are not needed.
13. The BI360 Planning input window is meant to be used for a monthly budget or forecast, not for daily or weekly budgets.
 - a. Spreading can be done over any number of months, but it must match the number of comparative months in the Interface settings.
14. Keep it simple. Don't overly complicate.
15. Use the IFERROR to ensure that there is no #N/A or #ERROR in your executed template.
16. Limit the size of budget templates.
17. Test and ensure that the data stores correctly in the warehouse. This can be accomplished through BI360 Report Composer, BI360 Warehouse Console, or by drilling down in the actual cell with the input.

18. If protecting worksheet, then determine if there should be access to hidden rows and columns. If not, then uncheck the ability to access locked cells.
19. Color code the hidden columns a different color.
20. Functions that may be useful to use in building templates:
 - a. VLOOKUP
 - b. HLOOKUP
 - c. ROW: typically used with the lookup functions.
 - d. COLUMN: typically used with the lookup functions.
 - e. SUMIF
 - f. IFERROR
 - g. MATCH
 - h. INDIRECT
 - i. INDEX
 - j. SUMPRODUCT

Line Item Details

Note: *Line Items Details* refer to the detailed budget transactions that are created when using the Line Item detail data entry window in the BI360 Planning module.

1. When using *Line Item Details*, it is recommended to specify shared dimensions as a *Sheet* dimension rather than a *Row* or *Column* unique dimension. This is because *Planning* builds line item details (LID) queries based on the global dimensions for performance reasons. As an example, if *Entity* is the only dimension defined as a global dimension, then all *Line Item Details* from the referenced *Entity* will be returned from the database. However, if more dimensions such as *Scenario*, *Category*, and other optional dimensions are specified, the query will be executed more efficiently.
2. Close the planning window prior to re-running the form.
3. If the account in the planning window is not the same as the account where the cursor is located, then close the planning window and reopen.
 - a. The other way to see this is if the line-items in the template don't match the line-items in the planning window.
4. If data has been stored and there is no need to enter data at this time, then close the form completely.
5. If the line-item details are not in the planning window, but are in Excel, then close the planning window and reopen by clicking **enter data**. This assumes that the line-item details are brought into Excel for displaying purposes.
6. Click **Update Excel** after entering in line-item detail for each account.
7. Click **save data** periodically, at least every 30 minutes. It is recommended to click **store data** after entering in the first account to ensure that storing is working properly.
8. If the following warning appears, then click **Yes**. If **No** is clicked, then all line-item details will be deleted.



Data Warehouse and SSIS

1. Sort and filter on dimensions (not attributes), as indexing is usually implemented on the dimension code and not the attributes.
2. Be sure that the Category dimension is always enabled and populated for all data sets (imported through the Warehouse Manager, SSIS, or stored from BI360 Planning).
3. When importing from GL's with an Account String (Dynamics GP, SL, etc.), only import individual account string segments (to different dimensions), not the entire string.
 - a. Typically this can be avoided as input can be limited to history.
 - b. Client needs to ensure that all segment values have descriptions in the ERP so these can also be loaded into the data warehouse.
 - c. It should be noted that the entire GL account string is a poor choice of dimension, as it limits the flexibility of reporting.
4. Map out modules, dimensions, attributes, and data types in the BI360 data warehouse prior to starting to build BI360 Planning or Reporting templates.
5. Use a standard SSIS package template wherever possible, for well-known ERPs. Use custom package structure only if called for by specific customer requirements.
6. Suggest and plan for basic operational notification with a success or failure of deployed packages (i.e. email alerts, file logging, as requested).
7. Emphasize to customer that SQL server optimization below the level of the application is required for best performance of the reporting environment. This means that the customer must take ownership of following Microsoft best practices for SQL implementation if not specified in the Statement of Work (SOW).
8. Add a simple index to the BI360 Warehouse. An example is below for a database that uses dimensions zero through four. Note the dimensions should be changed for each customer as not all will have four dimensions.

CREATE NONCLUSTERED INDEX IX_GLFact

ON [dbo].[f_Trans_GL] ([Scenario],[Category],[Entity],[Account])

INCLUDE ([Dim0], [Dim1], [Dim2], [Dim3], [Dim4], [Value1],[IntDate])

Form/Report Template Style

1. Ensure that the print settings are enabled and that it prints nicely.
2. Freeze pane is enabled.
3. Solver's color and style is as follows:
 - a. Recommended font is Calibri-10.
 - b. The background should be the white (or disable gridlines) in the color scheme or 255, 255, 250.
 - c. The input cells should be a yellow box with the scheme 255, 255, 225.
 - d. Headers should have a blue background with text in bold, white.

Project Management

1. Follow the process mapped out if there is no project manager assigned.
2. Follow the project manager's process if there is a project manager assigned.