IFS Business Reporter

High Level Design Guidelines

Contents

IFS Business Reporter - High Level Design Guidelines

This document provides some rather high level design guidelines related to IFS Business Reporter (BR). The idea is to provide some basic recommendations, discuss design size and performance related topics.

IFS Business Reporter is an Excel based reporting tool. The combination of Excel and BR flexibility enables report designers to create an almost infinite number of different type of reports. The following sections will discuss some general design guidelines and some tips & tricks.

Expectations

The expectations on BR should to some extent be high but it is also important to understand that just because a report can be created and visualized in Excel the expectations cannot be the same regardless of design, size, user Information Sources etc.

Report Size

The report size can be an important factor.

A customer reporting or presentation requirement can be transformed into a BR report by a BR designer. A rather easy trap to fall into is to create too large reports, i.e. a reports with large amounts of BR design instructions as well as ordinary Excel functionality.

So what does "*large*" mean and what can be the consequences of a large report?

A large report can be large in many ways. Normally this is related to design elements and how they are used. Please refer to **Design Element Limitations** section.

A general guideline would be not to use one single BR report to produce contents that could as well be represented by several reports. Assume that a report with 25 design sheets is created and that the executed report contains 40 sheets, due to use of Sheet Repeaters (i.e. repeating sheets). The single report could maybe be said to correspond to 20-30 separate reports. If another tool than BR was used, there would be say some 20 separate reports. Now only one report is produced.

But there is nothing that says that one design sheet corresponds to one report. It is more a matter of making a logical/practical limitation of the report size, trying to come up with a design that is understandable, maintainable while still producing a report with contents that has makes sense, has a value, for the customer.

One problem with a large report is of course the performance expectations. The larger a report gets, the higher the probability is that BR will generate more server SQL statements to be executed. This may lead to more time spent on the server side and thus a longer execution time. Many data sets does not necessarily mean that the client will get a large data volume to handle but the possibility certainly increases. The client processing will definitely increase with increased design size.

So a general rule of thumb is trying to keep the individual BR reports rather small.

A list of issues related to large reports:

1. Difficult to understand the design
2. Difficult to further develop the report
3. Difficult to maintain/correct the report
4. Easy to create incorrect cell references
5. Large amount of named ranges
6. Risk for memory problems
7. Reduced server performance
8. Reduced client rendering performance
9. By any chance if the report is corrupted it will be very difficult to create the same report again

Data Structure in IFS Applications

A very important thing to consider is the data structure in IFS Applications. For e.g. financial reporting, the way the code string is defined, how the code parts are related to each other etc will be have a big impact on a BR report.
If it is necessary to create a large number of different design elements just to filter out costs, revenues etc. per code part combination, then it is wise to start looking into the basic data setup before starting the report design phase. It is difficult to achieve efficient BR reports that has to overcome a bad underlying data structure instead of trying to solve the real problem.

Design Element Limitations

There are no warnings currently in IFS Business Reporter when it comes to how the report is using the available design features. This type of design awareness functionality will be added in future BR release as one way of highlighting for the designer that it might be necessary to consider redesign.

Below is a table with some recommended max design limits:

Check Point	**Suggested MAX limit**
Design Sheets	10
Design Rows	100
Design Columns	50
Design Cells	100
No of Client Data Sets	300
No of Server Data Sets	100

If a BR report contains design such that one of the above limits are exceeded then there is a chance that the report performance may suffer and redesign might be needed.
NOTE that there is no guarantee that a small amount of design always is a good thing and that a large design is a bad thing. But the above limits at least serves as a guideline, something to look out for.

Repeater Depth

The repeater concept is recommended instead of using a lot of individual design rows/design columns.
There is however one things that is worth noting:

- The more inner repeaters that are defined in a design row/column, the bigger the risk is to decrease performance. Trying to limit the number of inner repeaters to somewhere between 1 and 3 can be a good rule of thumb. Thus the amount of inner repeaters, the repeater depth, in general affects the rendering performance negatively.

Of course the rendering performance also depends on how much data BR retrieves.

Check Design Sheet Size

Always perform a check in each BR design sheet when it comes to active sheet size.
Press <Ctrl><End> to see what cell Excel considers being the last in a given sheet.
If this cell is lower down than the last BR design row or further out to the right than the last BR design column, then consider removing rows and columns that are “empty” and make sure that the “end” cell is at least close to the intersection between the bottom design row and right most design column.

Of course it is necessary to consider charts etc. that are not BR design elements and that are placed outside the BR design elements.

Report Performance

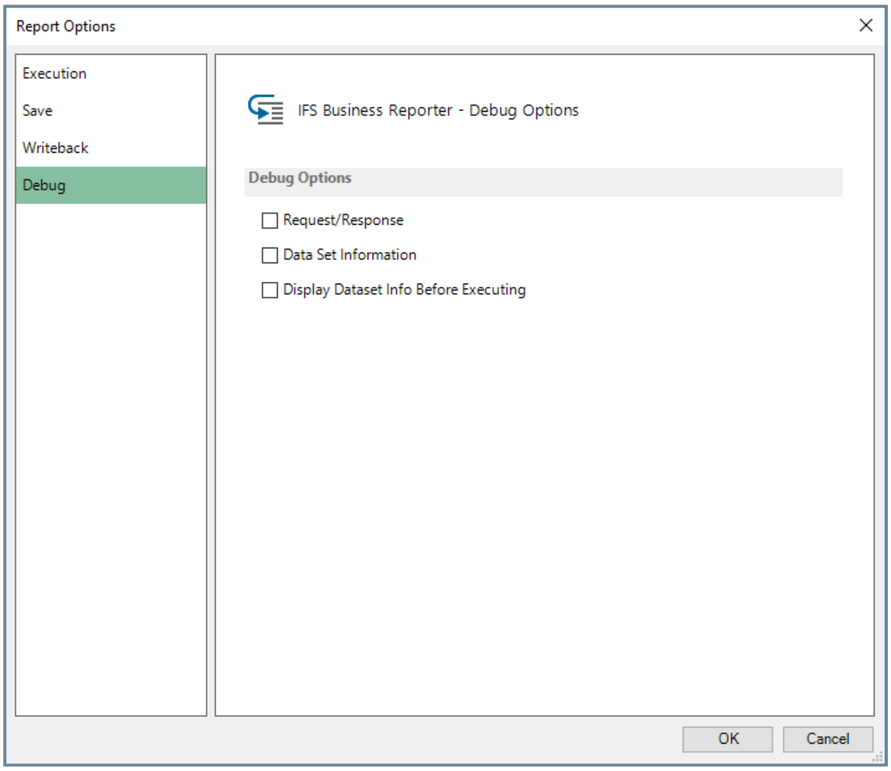
The performance of a report is affected by many factors.

- The complexity of the design, i.e. number of inner repeaters, number of design elements, references between cells etc.
- Number of data sets
- Used Information Sources
    - Attributes used in design -> some attributes cost little while some may cost more if an underlying expression is needed to extract the attribute value
    - Number of dimensions -> affects number of joins when executing the SQL query
- Online or Data Mart access
    - Data Mart access might be beneficial since all information in this case is calculated and stored in separate tables, compared to the online access where everything must be evaluated at execution time.

A designer should always investigate the performance before publishing a report for common use.

The first step would be to get a general feeling if it is ok or not.

Next step could be to enable the BR Debug Options. In BR Design Mode click on Report Book and then on Options:



Debug Options

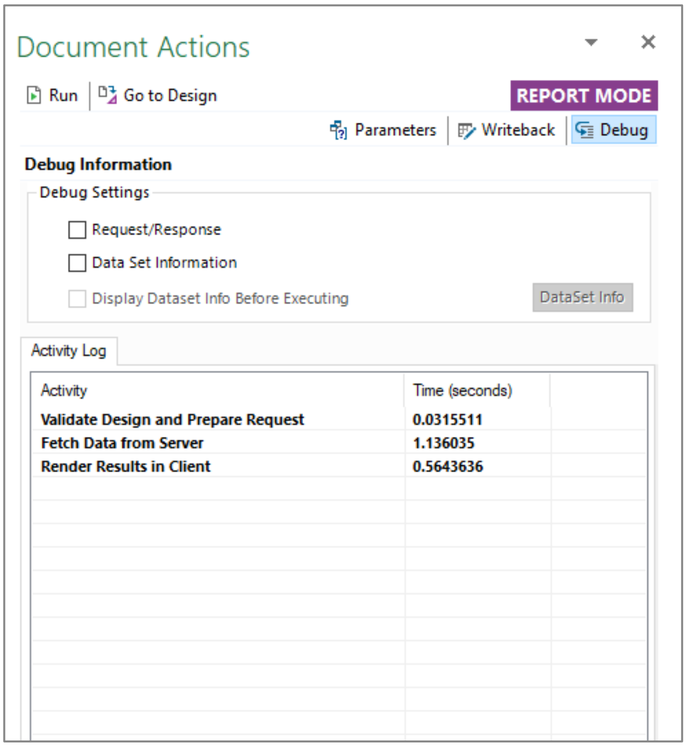
□ Request/Response

□ Data Set Information

□ Display Dataset Info Before Executing

Report will always display how long time the report has taken to execute and render even without enabling debug options.

Execute the report in BR and when ready, open the Options dialog in End User mode:



Debug Information

Debug Settings

□ Request/Response

□ Data Set Information

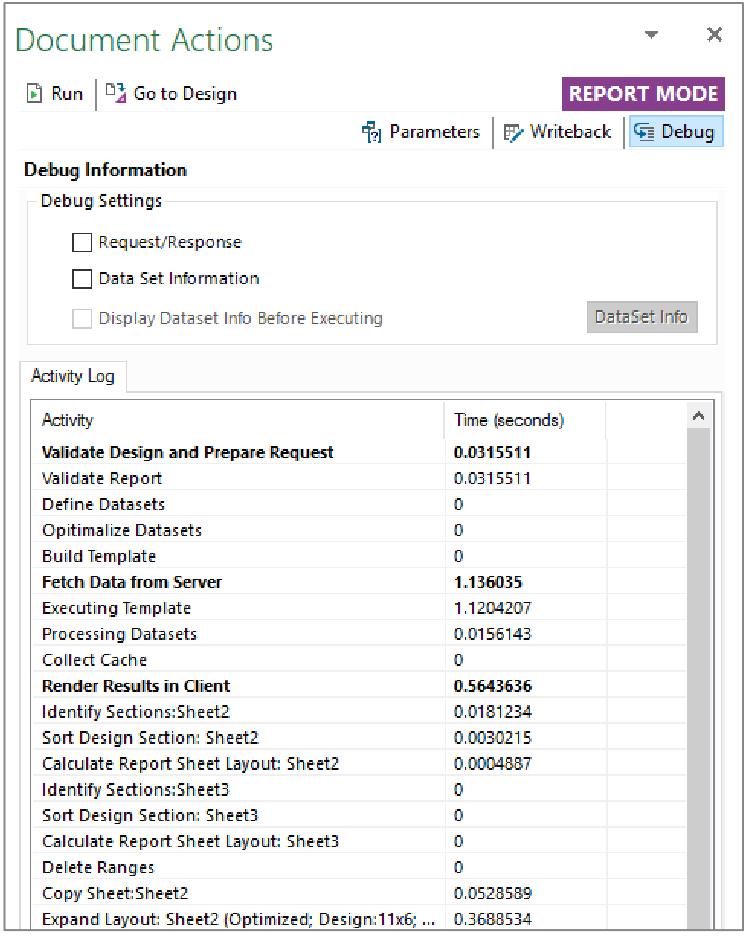
□ Display Dataset Info Before Executing

Activity	Time (seconds)
Validate Design and Prepare Request	**0.0315511**
Fetch Data from Server	**1.136035**
Render Results in Client	**0.5643636**

The output will show the time used on the server side and on the client side.

This information can be useful to find out if a report that takes time spends most part of the time in the client or in the server.

The above activities can be expanded by using RMB **Show Details**:



Debug Information

Debug Settings

□ Request/Response

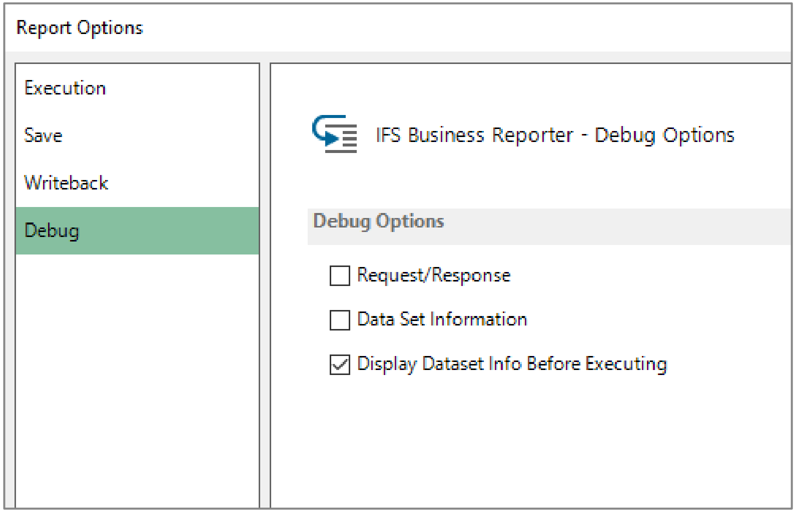
□ Data Set Information

□ Display Dataset Info Before Executing

Activity	Time (seconds)
Validate Design and Prepare Request	**0.0315511**
Validate Report	0.0315511
Define Datasets	0
Opitimalize Datasets	0
Build Template	0
Fetch Data from Server	**1.136035**
Executing Template	1.1204207
Processing Datasets	0.0156143
Collect Cache	0
Render Results in Client	**0.5643636**
Identify Sections:Sheet2	0.0181234
Sort Design Section: Sheet2	0.0030215
Calculate Report Sheet Layout: Sheet2	0.0004887
Identify Sections:Sheet3	0
Sort Design Section: Sheet3	0
Calculate Report Sheet Layout: Sheet3	0
Delete Ranges	0
Copy Sheet:Sheet2	0.0528589
Expand Layout: Sheet2 (Optimized; Design:11x6; ...	0.3688534

Another useful functionality is to figure out how many data sets the client is creating and how many server data sets this leads to.

Again in Design Mode, open the Options dialog and also enable **Display Dataset Info Before Executing**



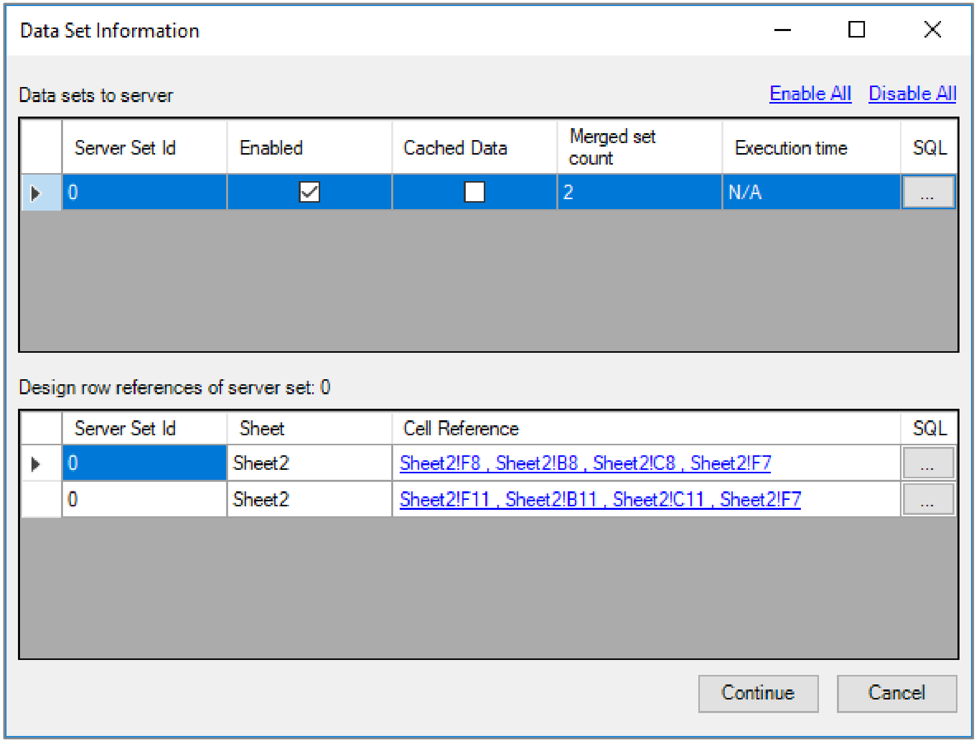
Debug Options

□ Request/Response

□ Data Set Information

☑ Display Dataset Info Before Executing

When the report is executed in BA, a dialog will be displayed that shows number of data sets:



Data sets to server

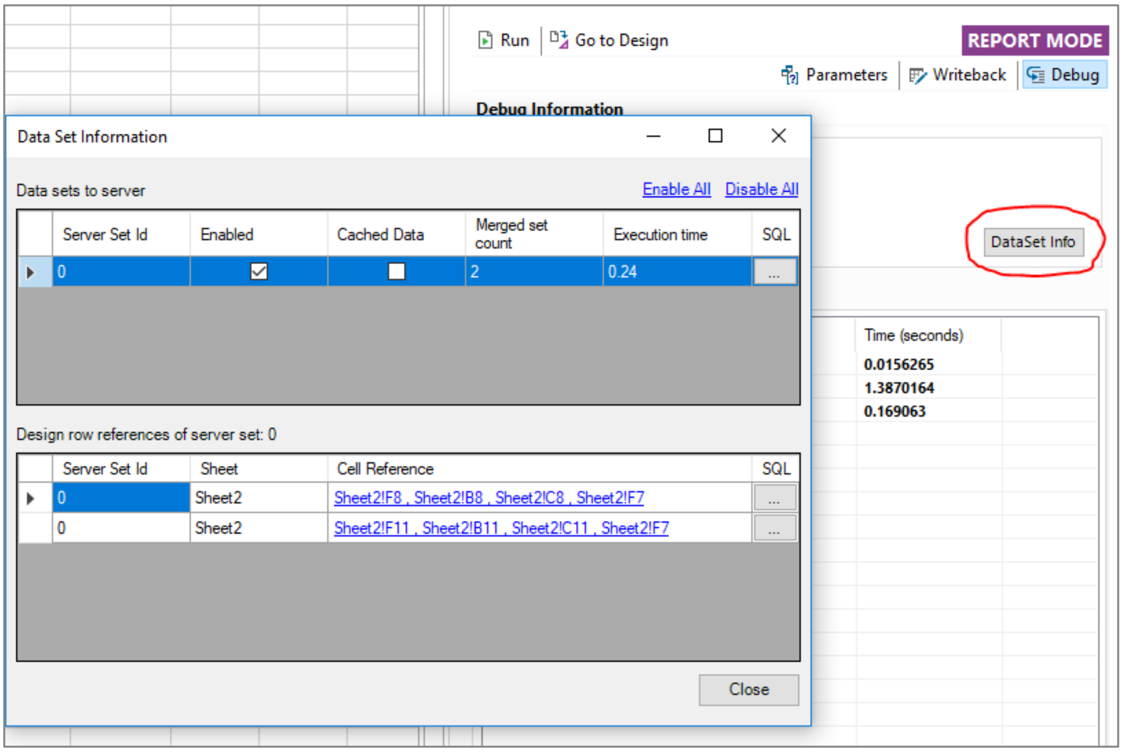
Server Set Id	Enabled	Cached Data	Merged set count	Execution time	SQL
0	☑	□	2	N/A	...

Design row references of server set: 0

Server Set Id	Sheet	Cell Reference	SQL
0	Sheet2	Sheet2!F8 , Sheet2!B8 , Sheet2!C8 , Sheet2!F7	...
0	Sheet2	Sheet2!F11 , Sheet2!B11 , Sheet2!C11 , Sheet2!F7	...

In the above case there are two server data sets and for each data set all related client data sets are listed. This dialog then gives the possibility to figure out how many server and client data sets the BR client design will generate and also the reference to the cells that generate the client statement.

When the report has been executed, in End User mode open the Options dialog and click on **Data Set Info** to launch the data set dialog:



Data Set Information

Data sets to server

Enable All Disable All

Server Set Id	Enabled	Cached Data	Merged set count	Execution time	SQL
0	☑	□	2	0.24	...

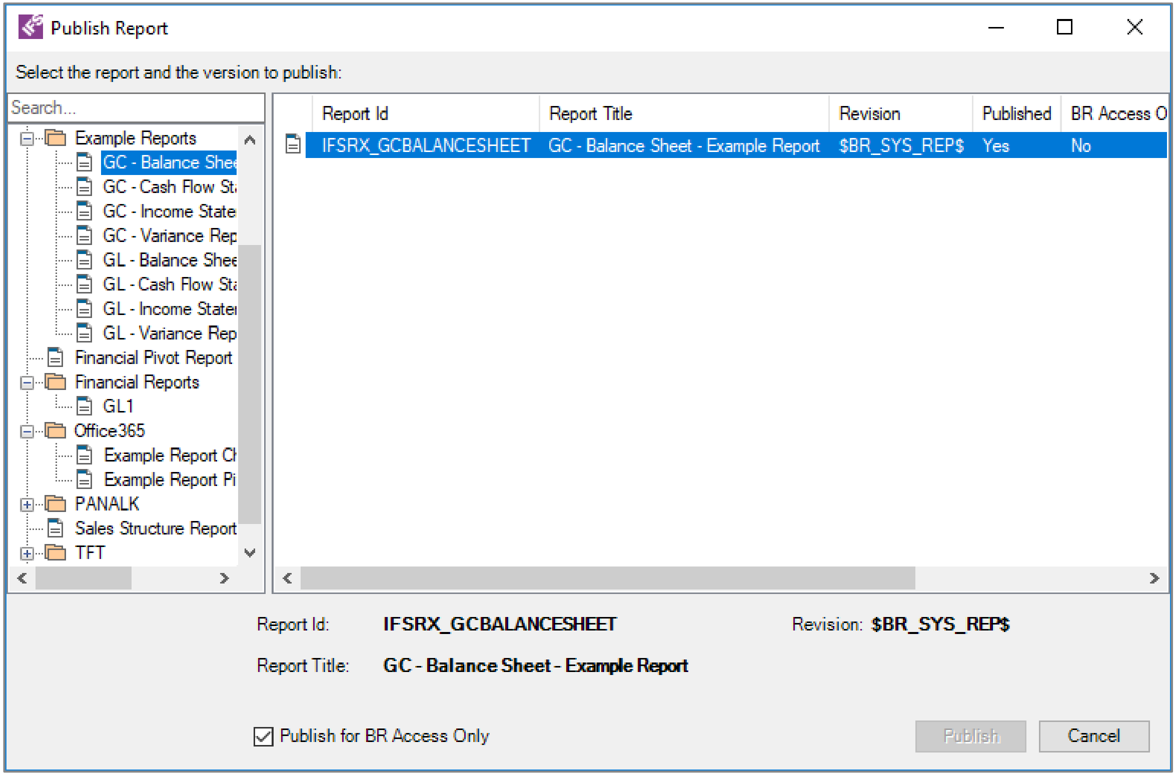
Design row references of server set: 0

Server Set Id	Sheet	Cell Reference	SQL
0	Sheet2	Sheet2!F8 , Sheet2!B8 , Sheet2!C8 , Sheet2!F7	...
0	Sheet2	Sheet2!F11 , Sheet2!B11 , Sheet2!C11 , Sheet2!F7	...

Time (seconds)
0.0156265
1.3870164
0.169063

As can be seen from the above picture each server data set also has info about the execution time.

To figure out more details about the SQL statements and how they can be optimized, the first thing would be to publish the BR report for **BR Access Only**. This is a publish option in BA:



Select the report and the version to publish:

Report Id	Report Title	Revision	Published	BR Access O
IFSRX_GCBALANCESHEET	GC - Balance Sheet - Example Report	BR_SYS_REP	Yes	No

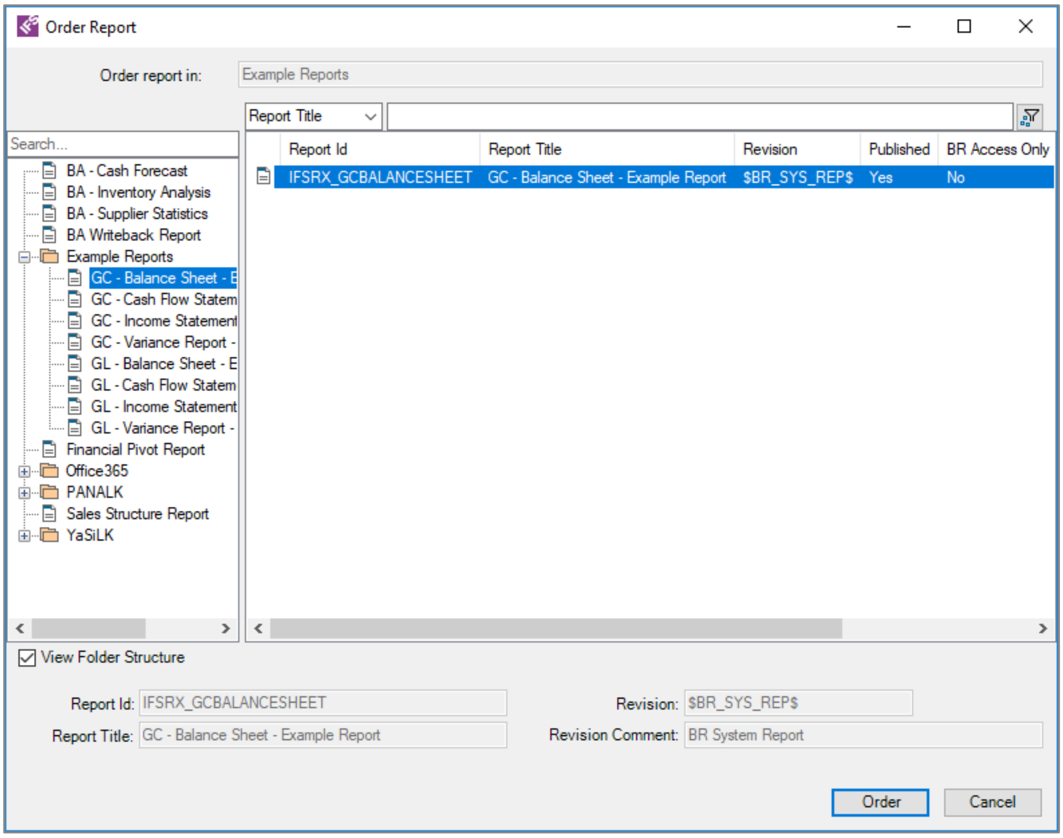
Report Id: **IFSRX_GCBALANCESHEET** Revision: **BR_SYS_REP**

Report Title: **GC - Balance Sheet - Example Report**

☑ Publish for BR Access Only

After having published the report it can only be accessed from BR but it is possible to use server side configuration functionality to investigate and sometimes even improve performance.

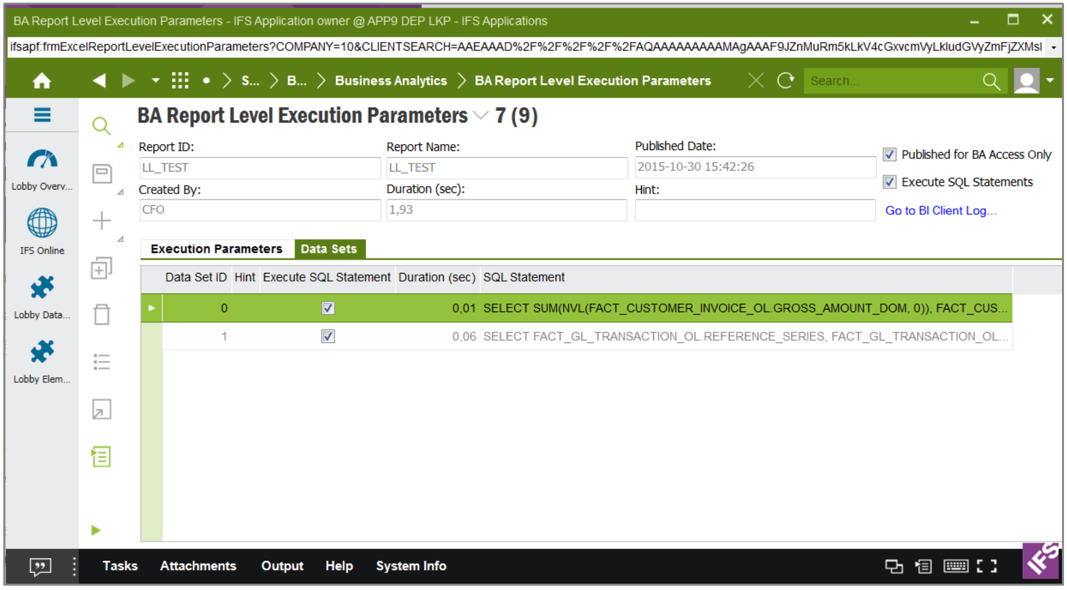
Now in BA, End User mode, use **Order Report** to execute the report:



Report Id	Report Title	Revision	Published	BR Access Only
IFSRX_GCBALANCESHEET	GC - Balance Sheet - Example Report	BR_SYS_REP	Yes	No

Report Id:	IFSRX_GCBALANCESHEET
Report Title:	GC - Balance Sheet - Example Report
Revision:	BR_SYS_REP
Revision Comment:	BR System Report

When the execution is ready, go to the IEE client and open the ***BR Report Level Execution Parameters*** form. It contains information about the total execution time, the time per data sets, SQL statement for each set etc.



BA Report Level Execution Parameters - IFS Application owner @ APP9 DEP LKP - IFS Applications
ifsapf:frmExcelReportLevelExecutionParameters?COMPANY=10&CLIENTSEARCH=AAEAAAD%2F%2F%2F%2F%2FAQAAAAAAAAAMAgAAAF9JZnMuRm5kLkV4cGxvcmVyLkludGVyZmFjZXMsI
S... > B... > Business Analytics > BA Report Level Execution Parameters
Search...
Lobby Overv...
IFS Online
Lobby Data...
Lobby Elem...
BA Report Level Execution Parameters 7 (9)
Execution Parameters
Data Sets
Go to BI Client Log...
Tasks
Attachments
Output
Help
System Info

Report ID:	LL_TEST
Report Name:	LL_TEST
Published Date:	2015-10-30 15:42:26
Created By:	CFO
Duration (sec):	1,93
Hint:	
☑ Published for BA Access Only	
☑ Execute SQL Statements	

Data Set ID	Hint	Execute SQL Statement	Duration (sec)	SQL Statement
0		☑	0,01	SELECT SUM(NVL(FACT_CUSTOMER_INVOICE_OL.GROSS_AMOUNT_DOM, 0)), FACT_CUS...
1		☑	0,06	SELECT FACT_GL_TRANSACTION_OL.REFERENCE_SERIES, FACT_GL_TRANSACTION_OL...

The above functionality gives the possibility to investigate the server data set execution performance and also to apply execution parameters on report or on data set level.

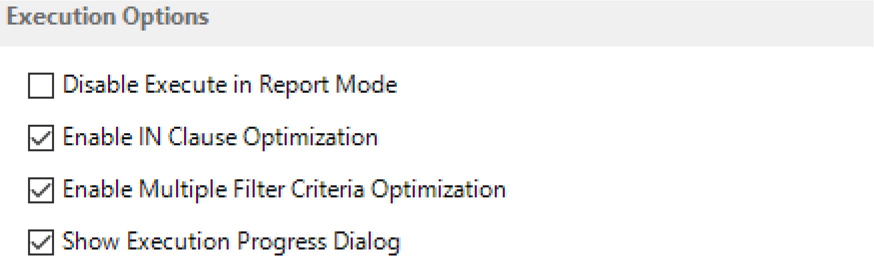
More info can be found in the Technical Documentation for each Applications track.

Number of Statements

BR always tries to optimize the report execution by looking for similar client statements and to merge them together. The objective is to reduce the number of statements executed on the server side.
Now since this optimization is done as far as possible, the designer should look out for a design that generates hundreds of server statements. Each server statement is represented by a SQL statement that joins necessary dimensions with the "driving" fact. The number of transactions in each referenced table, number of joins, conditions used, available indexes, Oracle version, db statistics etc will affect the performance for each set. Thus the number of server data sets can be a critical issue if there many of these sets take a rather long time to execute.
Also for a case where e.g. 1000 server statements are generated and each statement only takes 1 second, the total server execution may will be 1010 seconds, i.e. nearly 17 minutes.
So to reduce the number of server statements is always a good thing to strive for.

BR Design Execution Options

The BR client has some execution options that can important to consider.
In BR Apps 10 the options are:



Execution Options

☐ Disable Execute in Report Mode
☑ Enable IN Clause Optimization
☑ Enable Multiple Filter Criteria Optimization
☑ Show Execution Progress Dialog

Enable IN Clause Optimization and Enable Multiple Filter Criteria Optimization.
The data set optimization algorithm was improved in Apps 9 to enable merge of more client data sets than earlier into server data sets. These two options that should be enable by default and only changed if e.g. there is a suspicion that the optimization is not working correctly or if the statement generates more IN values in the SQL statement than possible to handle.

BR Execution Server

When a BR report has been designed, tested and set into production, it is highly likely that it will be ordered via IFS Reporting (Info Services). There are two general options:

- Using the Order Report dialog
  - The report will run and execute directly and when the result is ready the BR report will open and be rendered.
- Using a Scheduled task
  - Only the server part will be executed.
  - When the report is opened from the Archive the BR report will open and the rendering will start.
  - This option has the advantage that the user does not have to wait for the server execution. It is possible to configure that a mail should be sent to the user when the report is ready.
  - Still the user has to wait for the rendering of the report when opened from the archive.

A very nice option when it comes to BR reports is to also install the BR Execution Server. This server software should preferably be installed on a separate computer and the general recommendation is to run it on a server OS.
It possible to define which reports that the BR Execution Server should handle:

- All reports ordered from Reporting (Info Services)
- Only scheduled reports

The BR Execution Server will download the report definition and the report data as soon as the data is available (the server first has to execute all server side SQL statements and gather the total data). Next step will be to render the BR report with the downloaded report data before uploading it again to the IFS Report Archive.
The big benefit with this solution is that when a report ends up in the IFS Report Archive it is completely ready. This type of functionality can be very beneficial for the end user and especially when scheduling BR reports.