Performance Optimizing SAP BusinessObjects Reports Based upon SAP BW using BICS Connectivity

“How to performance optimize SAP BusinessObjects reports based upon SAP BW using BICS connectivity”

Applies to:
SAP BusinessObjects (BO) BI 4.0 Web Intelligence using Business Intelligence Consumer Services (BICS) Connection with SAP BW as data source.

Summary:
This paper provides a step-by-step description including screenshots to evaluate how SAP BW interprets SAP BO Web Intelligence reports using the BICS connectivity. This functionality, also known as tracing or performance analysis on BW & BO, can if used properly increase the performance and user convenience of SAP BusinessObjects Web Intelligence and other reporting tools using BICS tremendously.

The results of the tracing exercises can give important information on how to optimize BW in general and more importantly redesign BEx Query for the purpose of consuming data in Web Intelligence reports.

Without evaluating the effect (sometimes referred to as a query execution plan) on BW, when a Web Intelligence report is executed in BICS, connectivity users of Web Intelligence can experience less than optimum performance (Rephrase please). This paper will give you insights into how you trace yourself and it will give you an understanding of how Web Intelligence using BICS works.

Authors Bio

Kurt Holst (kurt.holst@sap.com) is an expert on SAP BusinessObjects Enterprise architecture, configuration and performance optimization. Kurt has worked with most major OLAP databases, SAP NetWeaver BW and relational databases together with different BI reporting tools. Before joining SAP Kurt has worked as a BI Architect & developer both internal and as an external consultant. Furthermore Kurt has been attached to the university as an external lecturer on Business Intelligence, Data Warehousing & Data Mining. Kurt is organizational with the Business Analytics Services team in SAP Denmark.

Anil Lad (anil.lad@sap.com), Senior BW consultant for valuable inputs & descriptions regarding BW.

Company: SAP
Last update: September 24 2013
Contents

1  Document update history ................................................................. 3
2  Background .................................................................................. 4
1  Steps required for setting trace up on SAP BW ................................ 7
   1.1 Clear existing cache to avoid misinterpretations ......................... 7
2  Steps required for setting trace up on SAP BO ................................ 10
3  Evaluate the trace logs gathered ...................................................... 13
4  BO performance considerations for BEx Queries ............................ 17
   4.1.1 Is Query stripping still relevant as of SAP BO 4 + SP04 – the new lean universe? ........................................ 20
   4.1.2 Building high performing Web Intelligence reports: .............. 20
5  Tracing Web Intelligence reports ...................................................... 27
   5.1 How to narrow the search to get the exact QueryView statement generated by WeBEn ....................... 27
6  Tuning based upon findings from log files ...................................... 30
   6.1 Using Wily IntroScope to trace the entire value chain from BW to Web Intelligence ......................... 30
   6.2 Setting up IntroScope in BusinessObjects CMC: ................. 31
   6.3 Fiddler .................................................................................... 32
   6.4 HTTP Watch ......................................................................... 32
   6.5 Wireshark ............................................................................. 32
   6.6 Performance Monitor .............................................................. 33
7  BW performance considerations for BEx Queries and BICS ............... 35
   7.1.1 BEx query – best practices .............................................. 35
   7.1.2 MultiProviders – best practices ....................................... 37
   7.1.3 InfoCube modeling-best practices .................................... 37
   7.1.4 Using many Key Figures – Data modeling options ............ 38
   7.1.5 Using Hierarchies .............................................................. 41
   7.1.6 Infocube on-going activities ............................................ 42
   7.1.7 Consistency checks ............................................................ 46
   7.1.8 OLAP cache ................................................................. 47
   7.1.9 Basis monitoring ............................................................... 49
   7.1.10 BW Accelerator .............................................................. 50
   7.1.1 BW on Hana ................................................................. 51
7  Performance evaluation .................................................................. 52
   7.2.1 Query statistics: Transaction RSDDSTAT ........................... 52
   7.2.2 Query Monitor: Transaction RSRT .................................... 56
   7.2.3 Trace Tool RSTT .............................................................. 59
   7.2.4 Comparing BW BEx Query runtime to BO Web Intelligence report ................................................................. 61
8  Reference Documentation ................................................................. 63
Copyright ......................................................................................... 64
# Document update history

<table>
<thead>
<tr>
<th>Date</th>
<th>Name</th>
<th>Alteration Reason</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.08.2012</td>
<td>Kurt Holst</td>
<td>Initial document</td>
<td>0.1</td>
</tr>
<tr>
<td>30.08.2012</td>
<td>Kurt Holst</td>
<td>Background and overview</td>
<td>0.2</td>
</tr>
<tr>
<td>09.10.2012</td>
<td>Kurt Holst</td>
<td>Tracing &amp; log files</td>
<td>0.3</td>
</tr>
<tr>
<td>17.10.2012</td>
<td>Anil Lad &amp; Kurt Holst</td>
<td>Webl. and BW considerations for high performance</td>
<td>0.4</td>
</tr>
<tr>
<td>19.10.2012</td>
<td>Kurt Holst</td>
<td>Adding best practices Webl</td>
<td>0.5</td>
</tr>
<tr>
<td>02.11.2012</td>
<td>Kurt Holst</td>
<td>Finalizing first draft to share internally for peer review and quality assurance</td>
<td>0.6</td>
</tr>
<tr>
<td>26.11.2012</td>
<td>Kurt Holst</td>
<td>Finalizing document in preparation to share on SAP Community Network</td>
<td>1.0</td>
</tr>
<tr>
<td>10.12.2012</td>
<td>Kurt Holst</td>
<td>Adding further details on tracing</td>
<td>1.1</td>
</tr>
<tr>
<td>10.01.2013</td>
<td>Kurt Holst</td>
<td>Adding further details BO tools general</td>
<td>1.2</td>
</tr>
<tr>
<td>10.06.2013</td>
<td>Kurt Holst</td>
<td>Update on Tracing the BICS Bookmark</td>
<td>1.3</td>
</tr>
<tr>
<td>22.08.2013</td>
<td>Kurt Holst</td>
<td>Update with latest patches &amp; Web Intelligence Querybookmark. Section 5.1.</td>
<td>1.4</td>
</tr>
<tr>
<td>24.09.2013</td>
<td>Kurt Holst</td>
<td>Update with SP05 patches &amp; Web Intelligence performance implications using groupings. Section 4.1.2.</td>
<td>1.5</td>
</tr>
</tbody>
</table>
2 Background

The BusinessObjects Enterprise platform is a highly flexible solution that allows users with different reporting needs to access this information.

To ensure that users achieve highest possible conveniences there is a need to optimize and improve maintenance related to BusinessObjects Enterprise services. There are many ways to optimize the BusinessObjects platform further – however, this depends on the context in which the platform is to be used. To evaluate whether the platform is performing at its optimum, there a number of ways to log activity and afterwards tune the necessary parameters to increase.

In general, these evaluation files can be divided into:

- Trace: contains technical information to debug the root cause of a problem for developers and support engineers
- Log: contains information about actions to take in order to resolve a problem for Business Intelligence Platform administrators

The cause of less than optimum performance, which users might experience with SAP BusinessObjects Web Intelligence using BW as data sources with the BICS connectivity, is in some cases related to design of the BEx Query. This article describes additional parameters of Web Intelligence on BW as data source, which can lead to optimized performance and improved user experience.

Outside the scope of this article is a very important aspect that can also lead to poor perceived performance when Web Intelligence is chosen to perform tasks that other BO tools do better – please refer to “Choose the right tool - product position”: use a dedicated OLAP tool such as Analysis for OLAP or Analysis for MS Office. It is advisable to consider using other tools in the BO portfolio for ad hoc or data explorations. Crystal Reports could also be used for ad hoc analysis – crystal only retrieves pages of data and is hence a better tool if the underlying amount of data in a report is considerable. Moreover SAP BO Explorer is also a choice to consider if the end user reporting requirements are of a more data exploring nature. The same goes for Analysis for OLAP which could also perform far better on a larger set of data in an ad hoc reporting scenario using BICS. https://scn.sap.com/docs/DOC-32449

This article will document that the use of trace functionality on BW will demonstrate how Web Intelligence reports are processed and also how to interpret this information gathered during tracing to improve reporting performance. The obvious question here is how reporting performance with Web Intelligence on BW can be improved by re-designing on either Web Intelligence or the SAP BW side?

When generating a Web Intelligence report based on BICS, which actually means a BEx Query developed in Query Designer, BI Metadata is transmitted from BW to Web Intelligence in order to enable users to select attributes, measures, hierarchies etc. With BICS connectivity there are no universes stored on the BusinessObjects Enterprise platform and Web Intelligence needs to generate a so-called transient universe in runtime (“on the fly”). In theory, this will imply that a BEx Query with a lot of metadata (key figures, restricted key figures, attributes, hierarchies etc.) could require more time to build this transient universe than fetching a universe on the SAP BusinessObjects Enterprise platform. With the former BusinessObjects Universe (UNV or UNX), Information & design tool universes pre-built and store universes in the BusinessObjects Enterprise platform.
The BICS connectivity is undergoing continuous development improvements and as of SAP BusinessObjects 4 + SP04 there are further enhancements - lean universe - to the BICS connectivity allowing better performance using Web Intelligence in development mode, as it optimizes the amount of BI metadata being sent to Web Intelligence.

Continuously improvements are also to be expected in Support Pack 05 as communication between BICS DSL Bridge layer and the SAP BO frontend tools has been optimized. In the following section, this will be described into further detail.

The safety belt, addressed in SAP BO 4.0 SP02 +Patch 16, limits the number of cells being sent to Web Intelligence. However BW has actually performed the full query and calculated the number cells to be sending back to Web Intelligence – just before BW would normally start to send cells back to Web Intelligence. This implies that you can now set the number of cells that are returned in a query are limited to the number that you specify. This limits the number of cells returned, but does not restrict the BW from processing all cells in the query. Read more on how to set the safety belt improving stability on the SAP BusinessObjects Enterprise platform: https://scn.sap.com/docs/DOC-31900

The intend of this article is to provide you with useful tips on how to enable additional trace features for both Crystal Reports, Web Intelligence reports and other tools using BICS, which run against a BW backend. These tips will improve your ability to analyze and read these traces. With this information, you will be able to perform more meaningful searches against both the SAP knowledge base and notes system.

There is a number trace and log files related to a BusinessObjects environment:

1) Tracing on the SAP BW environment using RSTT – transaction “trace tool”.
2) BusinessObjects CMC tracing for different servers and services for instance setting the log level for APS DSL Bridge, where you are able to see what is sent from DSL to BICS.
3) Tracing can also be applied to the database using ST05 – transaction to see the SQL statements etc.
4) When using the RSRT transaction collects query runtime statistics etc.

A number of articles and guidelines on how to integrate SAP BusinessObjects frontend tools with SAP BW already exist, however, this paper focuses on a more narrow technical perspective with SAP BW BEx Queries and BusinessObjects Web Intelligence or other tools using the BICS connectivity.

The scope: tracing Web Intelligence of reports on BW using BICS

Besides the points mentioned in this article, there are also a number of other factors to take into account in order to achieve highest possible performance in your BusinessObjects environment.

- For instance better performance can be achieved with optimization of the Web server & Web Application server splitting content into static and dynamic content (“Improving the User Experience in SAP BI Platform 4.0 with Apache and Wdeploy”, http://scn.sap.com/docs/DOC-6191).
- Furthermore, optimizing or up-scaling the BusinessObjects Enterprise backend servers and services ("How to size the APS", SAP Note 1694041).
- Etc.

**This article is divided into the following main sections:**

1) Steps to setup tracing on SAP BW – to **gather** tracing data.
2) The tracing data gathered in the trace files will be evaluated – to **evaluate**
3) Based on the information found in the trace files, a generic action-plan to increase performance using Web Intelligence on BW with the BICS connectivity will be create.

**Definitions:**

"**BICS**" is short for Business Intelligence Consumer Services, which is the connectivity that allows SAP BusinessObjects tools, such as Web Intelligence or Dashboard Designer.

"**DSL**" or The Dimensional Semantic Layer Bridge is what actually connects to a BEx Query and in the end BW. The Adaptive Processing Server hosts DSL. The Adaptive Server hosting the DSL Bridge should be sized for expected user load. It is recommended to that DSL is the only service hosted by a particular (split) Adaptive Processing Server in anything beyond a small test environment.

**Log** contains information about which action to take to resolve a problem for Business Intelligence Platform administrators.

**Trace** contains technical information to debug the root cause of a problem for developers and support engineers.
Steps required for setting trace up on SAP BW

The following steps describe how to collect data from the BW system during the execution of Web intelligence reports using BICS connectivity. Nothing can be customized on the SAP BusinessObjects side, when using the BICS connectivity.

Open your SAP GUI:

1.1 Clear existing cache to avoid misinterpretations

From the SAP GUI go to transaction code RSRT – Query Monitor. This is normally an area where you should ask your SAP Basis / administrator colleagues to assist performing these tasks.

Go to Cache Monitor:

From there select:
Clear the case for the current BEx Query:

To ensure you get all the proper results without the OLAP cache interfering, remember to clear the cache before you proceed with the actual tracing – use the RSRT transaction:
Delete the cache for the BEx query you use for testing purposes.

Use SAP Transaction “ST05 – Performance Analysis” which will allow you to trace:

For the purpose of tracing a Web Intelligence generated query select SQL Trace & RFC Trace.

Select “Activate Trace” if you want to trace on your own user and filter if you want to trace on another user.

Your trace status should now resemble this:

The BW system is now ready to log request from e.g. SAP BO Web Intelligence.
Steps required for setting trace up on SAP BO

If you already have your own BEx Query and OLAP connection skip this section and just run your Web Intelligence report already now.

After setting up the tracing logs to collect data during the execution of Web Intelligence reports on top of BW using the BICS, create an OLAP connection using SAP BusinessObjects Central Management Console (CMC) or SAP Information Design Tool (IDT).

The Central Management Console (CMC) is a web-based tool to perform regular administrative tasks, such as user, content, and server management. It also allows you to organize, and set security levels for all of your SAP BusinessObjects User Management System content. Because the CMC is a web-based application, you can perform all of these administrative tasks through a web browser on any machine that can connect to the server. All users can log in to the CMC to change their user preference settings. Only members of the administrators group can change management settings, unless explicitly granted the rights to do so e.g. unless the member has been delegated administrative rights.

In this example, CMC is used to create the BICS connection, which will be consumed by Web Intelligence and Analysis for OLAP later.

Central Management Console

OLAP Connections

Name: RS_SALES Q3
Description (optional): 
Provider: SAP NetWeaver Business Warehouse
Server Information: Server Type: Server
System: XEB
Server: xye...sap.corp
System Number: 02
Client: 800
Language: en
Save Language?: 
Connect to server to choose a cube: Connect
InfoProvider: Purchase Data for Compression
Query: E1 War Room, test safety belt C, large result
Authentication: Pre-defined
User: i070917
Password: **********
Save Cancel
From the BO frontend tools create a new Web Intelligence report using the connection just created (Bex).

Create a new Web Intelligence report based on the newly created BICS connection.
Select a number of dimensions and measures to the Web Intelligence report:

Web Intelligence Query Panel:

Web Intelligence Report panel:
3 Evaluate the trace logs gathered

The information gathered from the logs should be utilized in order to optimize performance and thereby, improve the user experience with SAP BusinessObjects.

The technical background for determining if BW is actually performing the full query generated by Web Intelligence is illustrated in the table below. The below screenshots from BW tracing (details on how to perform a trace in BW is deemed outside the scope of this article) show the RFC for BICS_PROV_GET_RESULT_SET which is the actual cells being transmitted from BW to the BusinessObjects Enterprise system (Adaptive Processing Server, Dimension Semantic Layer Bridge).

After running the Web Intelligence report stop the trace: Deactivate Trace

To see the recorded trace data click on: Display Trace
Trace:

Example of Web Intelligence query:

Selecting BICS_PROV_GET_RESULT_SET:

The trace in the BW system reveals that 2000 Cube records are retrieved by the BW OLAP from the database:
Furthermore, the trace also shows that 92281 bytes are transmitted to and retrieved 642 bytes from the BICS:

Selecting BICS_PROV_GET_RESULT_SET:

In your SAP BW system, using transaction code rsrt you can access query properties.
In the following screen, you will get several properties regarding caching, read mode, and you will have the option to check the **Use Selection of Structure Elements**. Selecting this option allows BW to optimize a query request coming from for instance Web Intelligence using BICS.

![Use Selection of Structure Elements]

You will now discover why it is Best Practice always to check this box as all your front-end tools will benefit from it.

The following is an example of a BEx query containing 4 Key Figures and a Web Intelligence report only containing 2 of them and the RSTT and SQL traces associated to this Web Intelligence report.

**1. BEx query:**

The **Use Selection of Structure Elements** impacts the SQL sent to the underlying database.

**Without the option** - the SQL trace shows that in the SELECT clause we have following Key Figures (BALANCE, DEBIT, CREDIT, QUANTITY), which is too much as we only need CREDIT & BALANCE.

```sql
SELECT
    [DI].[SID_OKISPER] AS [S__960],
    [DI].[SID_OKCURRENCY] AS [S__860],
    [DI].[SID_OKUNIT] AS [S__1367],
    SUM ([F].[BALANCE]) AS [Z__785],
    SUM ([F].[CREDIT]) AS [Z__847],
    SUM ([F].[DEBIT]) AS [Z__878],
    SUM ([F].[QUANTITY]) AS [Z__1178],
    COUNT(*) AS [Z__2818]
FROM
```

**With the option enabled** - the SQL trace shows that the SELECT clause is ok with only the needed Key Figures.

```sql
SELECT
    [DI].[SID_OKISPER] AS [S__960],
    [DI].[SID_OKCURRENCY] AS [S__860],
    SUM ([F].[BALANCE]) AS [Z__785],
    SUM ([F].[CREDIT]) AS [Z__847],
    COUNT(*) AS [Z__2818]
FROM
```

This means performance improvement at your database level with less data requested by the OLAP processor. This is really important when you have a query based on a multi-provider with 2 underlying’s cubes (one actual & one history).
4 BO performance considerations for BEx Queries

Within SAP BusinessObjects XI4.0 (actually from XI 3.1 SP3) a new feature called query stripping has been launched. Before speaking about the new feature (query stripping), it is very important to understand the micro cube concept.

The micro cube is a kind of cache that behaves as a non-live data source. Every object that is included in your query will be computed in the micro cube, irrespective of whether they are directly or indirectly used in the final rendered report. If you delete some objects in your rendered report, when refreshing the report, you will refresh the complete micro cube. This results in a performance impact as the data is retrieved by the query, which will not be directly or indirectly used in the report that will be rendered or displayed to the user.

Because the micro cube is more than a simple cache, its advantage is that you do not need to get an access to your live data source, when adding other reports, as everything is already loaded in the micro cube.

On the other hand, BEx is behaving in a different way: every time you add an object in your report, a query is regenerated; hence an access to the data base is done.

Query stripping is a new feature that will be available only starting XI 3.1 SP3. This is a new paradigm because by using this feature, you will only pay the price for what is consumed, and this is very similar to how BEx behaves. The introduction of the query stripping feature optimizes the query to fetch only DATA, which will be directly or indirectly rendered or displayed to the user, thus decreasing the response time and faster display of the report for the user. This feature will only be for BICS / OLAP universes, and it will be grayed out when using any RDBMS universe.

Query stripping must be configured in 2 steps that are part of the report generation. The first step is to enable query stripping within the query. This step enables the user to use query stripping within the report or not. If this query setting is not applied, the query stripping function within the document will not function.
The second step is to enable query stripping within the document. The query stripping function for the document is part of the ‘Document Properties’ and will apply for all reports (tabs) within the document. When enabled, the query stripping engine will determine what objects from the query are used within the document before executing the query against its source. A ‘stripped’ query will only retrieve data for those objects that are actually in use by the document. Any object that is in the query, but not used, will be ‘stripped’ from the query that is send to the data source. Objects that are within the document, but not used will be highlighted in ‘BOLD’ in the object panel.
Once a user adds an object in a report that is 'stripped', they will receive a #TOREFRESH message in the added column/row. After refreshing the document, the query will be extended with the 'added' object and return data.

When a user creates a report, by default the query stripping option box is disabled. When the option is enabled at the Query level, this property is not considered at the time of query generation until the document level property is enabled. After creation of the report, the report creator can enable the query stripping option at the Document level. The report can then be distributed to other users.
4.1.1 Is Query stripping still relevant as of SAP BO 4 + SP04 – the new lean universe?

With this release, the BICS layer is optimized to bring only what is needed and not all the BEx query metadata into the transient universe used by Web Intelligence. The optimizations imply that when executing a query from Web Intelligence less metadata is brought into the transient universe. The effect is better runtimes, since fewer RFC’s are actually performed on the BW system and fewer metadata is brought back to BO.

The benefit of the lean universe is only achieved when refreshing existing Webl. reports and not when developing new reports. When developing new Webl. reports the metadata is still fetched from the BEx Query.

In comparison to the query stripping that is optimizing the number of fields (data-columns, attributes etc.) brought into a report the lean universe is optimizing the metadata. In conclusion, the lean universe and query stripping functionalities are addressing completely different areas of optimizations and should both be leveraged where possible.

*Query stripping is set by default from SAP BO 4 + SP05.*

4.1.2 Building high performing Web Intelligence reports:

4.1.2.1 What is the effect of remove unused hierarchies in BEx Query?

BEx Query “Hierarchies” before and after. BEx Query designer screen shorts to show.

“Do not keep hierarchy not active”

Before:

![Before](image)

After – if hierarchies are not needed it is not enough to just uncheck “Activate Hierarchy Display”. But it must be deselected as shown below:

![After](image)
4.1.2.2 What is the effect of having many grouping sets in Web Intelligence?

When using delegated measures, Web Intelligence will create two data streams in order to delegate the aggregation to the BW server. This can have performance implications as the BEx variables will have to be fetched for each grouping set.

**Update with BO4.0 + Servicepack 05 (SP05):** a lot of performance improvements have been applied to the BICS connectivity. In relation to grouping sets as of SP05 Web Intelligence can now re-use the BEx variables from the first grouping set in the following groupings – of course only within the same BEx Query. Some BEx variables can depend their number of distinct values be quite time-consuming to pull into Web Intelligence and hence this is a good performance booster for the user experience in Web Intelligence.

Example – build a Web Intelligence document with delegated measures as shown below.
Building a grouping set in Web Intelligence. Drag fields from the available list onto the report pane.

Returning to the query and examining the grouping set 0:
Grouping set 1:

For each combination of dimensions and delegated key figures a grouping set will be created.

Web Intelligence requests total lines in separate requests.

The order of dimension in Web Intelligence Query Panel could be different in the actual Web Intelligence Report.

Web Intelligence Query Panel: A B C
Web Intelligence Report: A B C A

**Before servicepack 05**: For each grouping set a new BEx Query is created (BICS_PROV_OPEN incl. submit variables).

**After servicepack 05**: For each grouping set a new BEx Query is created (BICS_PROV_OPEN excl. submit variables) – meaning that the Web Intelligence query can re-use the variable data fetched from the first grouping set.

Totals and summary will have an impact on performance if the measures are set as delegated. If the measure is sum, count, etc., Web Intelligence is performing the aggregations. Only use delegated measures if they cannot be calculated in Web Intelligence.

4.1.2.3 What is the effect of having many BEx Queries in Web Intelligence?

Web Intelligence is processing multiple BEx queries (or other query connections) in sequential order – meaning that it is the total sum of all individual queries that determines the performance perceived by the report user. Furthermore multiple BEx Queries in a Web Intelligence report will consume from pool of maximum connections.
4.1.2.4 What is the effect of having Web Intelligence cache and BICS?

Monitor the cache settings in Web Intelligence Processing server to utilize that data is already fetched from BW using BICS. A simple way to perform this is to go to the Web Intelligence Cache folder and evaluate if new folders are created under the "\storage\docs if this is the case the case isn't utilized.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rites</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>sktop</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>wnloads</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>cent Places</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>&quot;\storage\docs&quot;</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>rikes</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>cuments</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>sic</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>tures</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>eos</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>outer</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>(C:)</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>P (D:)</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
<tr>
<td>profile (D:)</td>
<td>27.11.2012</td>
<td>File folder</td>
</tr>
</tbody>
</table>

4.1.2.5 What is the effect of having hierarchies with multiple top nodes?

Combining two BEx queries with multiple top nodes can have an impact on performance as Web Intelligence is expanding the full hierarchies.

To optimize the performance go to SAP GUI RSH1 and suppress the unneeded hierarchies.
Check the hierarchy Attribute settings for parameter “Suppress ‘Unassigned’ Node”:

4.1.2.6 Using Web Intelligence scheduled reports to enhance user experience

From either the BI Launchpad or the CMC use the schedule functionality to execute performance intensive demanding Web Intelligence reports.

Web Intelligence will create a report instance containing actual data. Using scheduling can greatly increase the user experience and in combination with Events it can be assured that the report isn’t executed before the BW system has loaded the data.
Users can then access report instances using the “History” functionality:

With the use of **Page mode** it allows Web Intelligence to leverage its Web Intelligence Cache Server.

The Page mode lets the BusinessObjects Enterprise platform server the user of Web Intelligence one report page at a time instead of sending all pages to the frontend. The report will open much faster than with “Quick Display mode” as Web Intelligence only fetches the first page of the report.

Limitations to using the page mode: don’t use page # of total pages which means that all pages have to be calculated.
5 Tracing Web Intelligence reports

The best practice is to use the Web Intelligence rich client to trace the time spend executing reports.
The Web Intelligence reports run using the BI platform has additional overheads in term of HTTP rendering etc.

5.1 How to narrow the search to get the exact QueryView statement generated by WebI.

As of SAP BusinessObjects SP04 + Patch 4 use the following approach to get the QueryView generated by Web Intelligence. From the CMC navigate to the Servers -> Services and select properties on the APS DSL Bridge Services (see How to APS for splitting the APS).

*This functionality was missing in some patches from BO4.0 SP05 and forward to SP06 Patch05.*

However thanks to the community giving feedback to this document the functionality is back in in SP06 Patch05 and will also be the same for all other branches (SP05_Patch, SP07_Patch, SP08 and BI4.1) as soon as possible.

Background on how to isolate and get the queryview executed by WebI. in 4 easy steps:

1. The process is to set add the parameter "-Dbicsbookmarkqv=true" on the server containing the DSL Bridge service.
2. Further trace log level should be set to high.
3. Also for good measure restart the DSL Bridge Service to have the parameter initialized.
4. Run the WebI report was executed but in the log file there is no bookmark.

The details:

Add the following to the command line parameter: `Dbicsbookmarkqv=true`

Scroll down to TraceLog Service and set level to "High":
Run the Web Intelligence report that you want to trace.

Open the aps_<Servername>_<Service>*.glf traces and look for BookMark and take the ID of the BookMark QueryView:

Noting or copying the Bookmark QueryView into the clipboard: 00AJ9JGY6YUJDSMUZY41S0UCB

From SAP GUI run transaction RSRT1:

Copy paste in the BOOKMARK ID in the “View” filed:
The QueryView run as generated by Web Intelligence.
6 Tuning based upon findings from log files

6.1 Using Wily IntroScope to trace the entire value chain from BW to Web Intelligence.

Wily IntroScope is an application monitoring tool that supports both JAVA & .NET technologies. Using wily, you can monitor your system at a high level. IntroScope monitors performance metrics, live and historical data. Furthermore, the ability to setup alerts on system performance and outages, you can generate reports. Wily can support visibility on end-to-end transactions.

In BOE 4.0, the full Wily Introscope metrics are collected from four parts:
1) BOE Java web application,
2) BOE Java based backend servers,
3) BOE C++ based backend servers and
4) BOE KPI metrics from some scripts.

Refreshing a WebI report:

Drilling down to the BICS:

What to look for:
GET_RESULT_SET is the actual retrieval of data from BW to BICS and then again to Web Intelligence.

Multiple BEx queries in one Web Intelligence report:
As can be deducted by the screen shot from above Web Intelligence runs each BEx query in serial mode which implies that the leadtime is a summary of all individual BEx Queries. When trying to achieve maximum performance in Web Intelligence, you should try to balance the number of BEx queries with the data granularity in the BW infocube.

6.2 Setting up IntroScope in BusinessObjects CMC:

Select Placeholders for each Node from the Servers menu. Consult SAP Note 1540591 – “Wily Introscope Setup for SAP BOE 4.0” for more detailed description.

Placeholders must be configured with the specifications of IntroScope & Solution Manager Diagnostic (SMD):
For more information on setting up Wily IntroScope please take a look at SAP Note 1540591.

6.3 Fiddler

Fiddler is a Web Debugging Proxy which logs all HTTP(S) traffic between your computer and the Internet. Fiddler allows you to inspect traffic, set breakpoints, and “fiddle” with incoming or outgoing data. Fiddler includes a powerful event-based scripting subsystem, and can be extended using any .NET language. 
http://www.fiddler2.com/fiddler2/

6.4 HTTP Watch

HTTP Watch is a tool that allows you to easily monitor HTTPS, HTTP and SPDY without using proxies or changing network settings.
http://www.httpwatch.com

6.5 Wireshark

Wireshark is a free and open-source packet analyzer. Like HTTP Watch it is used for network troubleshooting, analysis, software and communications protocol development.
http://www.wireshark.org/
6.6 Performance Monitor

Performance Monitor is used for network troubleshooting, analysis.

The Performance Monitor can be configured to capture data during a stress test.
Performance Optimizing SAP BusinessObjects Reports Based upon SAP BW using BICS Connectivity

System Monitor Log Properties

Performance Counters  File

Log file name: System Monitor Log

File name format: 

© 2012 SAP AG
 BW performance considerations for BEx Queries and BICS

7.1.1 BEx query – best practices

The design of BEx queries can have a significant impact on the performance. Sometimes long running queries are the result of poor design, not just the amount of data.

- It is significantly quicker when the query shows highly aggregated results that can then be drilled down with free characteristics. Move as few characteristics into the columns and rows as possible so that the number of cells sent to the frontend is kept as small as possible.
  ➔ Note: Web Intelligence (the BICS) does not look at whether the characteristics are placed under rows, free characteristics or columns.
- Where possible, define calculated and restricted key figures on the InfoCube instead of locally in the query.
  ➔ Web Intelligence will perform better if key figures are defined as summation. Key figures with ratio or other complex calculations will be database delegated in Web Intelligence (i.e. it pushes the calculation of these objects to the database server). Therefore, in a BEx Query, where possible, use summation for key figures.
  
  ![Local Calculations](image)

- The expected result set of the query should be kept as small as possible (max 1000 lines).
  ➔ Web Intelligence will fetch all transactions in BW that is in the result panel and store this information for further analysis in the local “microcube”.

- InfoCubes and MultiProviders are optimised for aggregated requests. A user should only report in a very restricted way on DataStore Objects (DSO) and InfoSets. In other words, only very specific records are to be read with little aggregation and navigation.

- All calculations that need to be made before aggregation (such as currency translation) should take place when loading data where possible (see note 189150).

- When using restricted key figures, filters or selections, try to avoid the exclusion option if possible. Only characteristics in the inclusion can use database indexes. Characteristics in the exclusion cannot use indexes. With selections it is better to include than exclude characteristic values.

- The setting “Result Rows” for each characteristic the in BEx query should be set to “Always Suppress”. Web Intelligence does not use the “Results Rows” and performance will improve if this option is set.
• The calculation of non-cumulative key figures takes a long time. In this case, the InfoCube must be compressed.

• Time characteristics should be restricted, ideally to the current characteristic value.

• Within structures: ensure that a filter order exists and that it is filtered at the highest level.

• Check code for all exit variables used in a report to make sure it is optimized and functioning.

• Move time restrictions to a global filter whenever possible.

• Reduce the data volume selected by introducing many mandatory variables.

• Define as few time-dependent objects as possible: joining the validity period is usually very expensive.

• Define only those characteristics as navigational attributes, which there is a business requirement for; navigational attributes need additional table joins at query runtime.

   ➔ Even though Web Intelligence does not distinguish between a normal characteristic and a navigational attribute, on the BW system there is still an impact on performance, when Web Intelligence reports are running on BEx Queries with navigational attributes.

• If hierarchies are used, minimize the number of nodes to include in the query results. Including all nodes in the query results (even the ones that are not needed or blank) slows down the query processing. The “not assigned” nodes in the hierarchy should be filtered out, and you should use a variable to reduce the number of hierarchy nodes selected.

   o When using hierarchies with a large amount of unassigned nodes, the performance will improve when suppressing the reporting on the unassigned nodes. This can be done for single hierarchies in transaction RSH1.
• Enable “Use Selection of Structure Elements” for queries which have several restricted and calculated key figures. (In transaction RSRT, enter query and select ‘Query Properties’)

- In the SQL statement only those KeyFigures (or structure elements) that are actually used, will be requested in the statement.
- If you select the flag “Use Selection of Structure Elements” then only the selected structure elements will be requested to be calculated, this means that the runtime of the RRK_DATA_GET function module will be shorter and the memory request for the structures DAT_N and SETXX will be shorter as well.

7.1.2 MultiProviders – best practices

• When the reporting scenario is to be extended, use a MultiProvider as central interface between query definition and basic InfoProviders. When another InfoProvider is added to the MultiProvider definition, the technical name of a query based on the MultiProvider remains unchanged.

• Use a MultiProvider to reduce the size of the basic InfoProviders. Advantages: parallel access to underlying basic InfoProviders, load balancing, resource utilization, query pruning.

• Make sure that your MultiProvider only retrieves data from relevant basic InfoProviders at query runtime by
  - Using constants in the design of the basic InfoProviders
  - Using different key figures in the design of the basic InfoProviders
  - Using characteristic 0_INFOPROV when designing a query on the MultiProvider
  - For more information, refer to How to… Create Efficient MultiProvider Queries guide

• Are you planning to use a MultiProvider? If so, you have to ensure that the characteristics you want to report exist in all basic InfoProviders.

• Do not use more than one non-cumulative InfoCube (InfoCube with at least one non-cumulative key figure) because this could lead to incorrect query results.

• Do not use calculations before aggregation on MultiProvider because this may lead to wrong query results.

• Avoid using only parts of compound characteristics in the constituent basic InfoProvider of a MultiProvider. For more information, see SAP note 702542.

7.1.3 InfoCube modeling-best practices

The data model has tremendous impact on both query AND load performance. For example, a bad dimension model (e.g. customer and material in one dimension instead of separate dimensions) can lead to huge dimension tables and thus slows down query performance, as it is expensive to join a huge dimension table to a huge fact table. Take these recommendations into account:

- Define as few time-dependent objects as possible: joining the validity period is usually very expensive.
- Define only those characteristics as navigational attributes, which there is a business requirement for; navigational attributes need additional table joins at query runtime.

- Try to keep the cardinality of dimensions small: don’t combine e.g. customer and material in one dimension if the two characteristics are completely independent.

- Try to re-design large dimensions: split them, move characteristics to other dimensions and use the Line Item dimension flag; as a rule of thumb, a dimension is too large (degenerated) when it’s cardinality exceeds 10% of the fact tables; document number and customer number are often candidates for Line Item dimensions; don’t use the option High Cardinality; use the report SAP_INFOCUBE_DESIGNS to analyze the size of dimensions and fact tables; the result is consistent when database statistics are up to date.

- Dimensional attributes generally have a performance advantage over navigational attributes for query operations. Fewer tables are joined to access the values of dimensional attributes than navigational attributes; therefore less overhead is seen during query execution. The requirements should be carefully analysed to determine whether a certain attribute needs to be navigational or if dimensional or display attribute will suffice. Be aware that by copying an attribute into a dimension has some business impact, whenever data changes.

- If you use non-cumulative key figures, avoid characteristic combinations that are very detailed (e.g. day / material / customer). Particularly, avoid using time characteristics that are too detailed (e.g. day).

- If you use non-cumulative key figures, use as few validity objects as possible.

- Use logical partitioning to reduce the sizes of the InfoCubes; for example, define InfoCubes for one year and join them via a MultiProvider (advantages: parallel access to underlying basis InfoCubes, load balancing, resource definition, query pruning). Beware that an excessive number of InfoCubes under a MultiProvider can generate a negative result in terms of performances.

- You need to partition InfoCubes by using Table Partitioning (also called physical partitioning). This is required as partitioning improves read performance. Fact tables can be partitioned over a time characteristic either OCALMONTH or OFISCPER. It is recommended that partitioning is done before loading any data.

### 7.1.4 Using many Key Figures – Data modeling options

During the BW data modeling phase, if there is a requirement for many key figures, then there are three different scenarios possible which can be used as a BW data model for such key figures:

1. All key figures of the data model are defined as InfoObjects and are added to the fact table of one InfoCube (Key Figure Model)
2. Data partitioning by a characteristic in an InfoCube Dimension (Account Data model) – one key figure of the data model is defined as InfoObject, which is added to the fact table of the InfoCube and the different semantic of this key figure is defined by a characteristic, which builds up a dimension of the InfoCube.
3. Data Partitioning by a MultiProvider – Detailed data of the sub scenarios (e.g. Sales Order, Delivery, Billing) are stored in the corresponding basic InfoCube. 
   - Detailed queries are defined with respect to the basic InfoCubes
   - Overview Queries use a MultiProvider, which represents the overall reporting scenario (e.g. Reports on Customer/Product/Sales Order Level).
### Dimensions and Characteristics

<table>
<thead>
<tr>
<th>SAP NetWeaver Business Warehouse object</th>
<th>Universe objects created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Group</td>
<td>Class</td>
</tr>
<tr>
<td>Characteristic</td>
<td>Subclass with dimension and detail objects</td>
</tr>
</tbody>
</table>
| Characteristic with hierarchy          | If data source is a BEx Query: Subclass containing dimension and detail objects for each hierarchy level in the currently defined hierarchy.  
                                       | If data source is an InfoCube: Subclasses containing dimension and detail objects for each hierarchy level for all hierarchies defined for the characteristic. |

### Attributes

<table>
<thead>
<tr>
<th>SAP NetWeaver Business Warehouse object</th>
<th>Universe objects created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigational attribute</td>
<td>Subclass with dimension and detail objects (identical to characteristic)</td>
</tr>
<tr>
<td>Display Attribute</td>
<td>Detail object for the dimension</td>
</tr>
</tbody>
</table>

### Structures

<table>
<thead>
<tr>
<th>SAP NetWeaver Business Warehouse object</th>
<th>Universe objects created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure based on Characteristics (BEx Queries only)</td>
<td>Class with single dimension object for the structure</td>
</tr>
<tr>
<td>KeyFigure Structure</td>
<td>Class</td>
</tr>
</tbody>
</table>
KeyFigures

<table>
<thead>
<tr>
<th>SAP NetWeaver Business Warehouse object</th>
<th>Universe objects created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Figure</td>
<td>Measure object in the class for the Key Figure structure with dimension objects for units/currency</td>
</tr>
<tr>
<td>Calculated Key Figure (BEx Queries only)</td>
<td>Measure and dimension objects (same as Key Figure)</td>
</tr>
<tr>
<td>Restricted Key Figure (BEx Queries only)</td>
<td>Measure and dimension objects (same as Key Figure)</td>
</tr>
</tbody>
</table>

Variables

<table>
<thead>
<tr>
<th>SAP NetWeaver Business Warehouse object</th>
<th>Universe objects created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables (BEx Queries only)</td>
<td>Filter mandatory in query. In the class for the dimension to which the variable applies, two dimension objects supporting the list of values, one for caption, one for description.</td>
</tr>
<tr>
<td>Key date variable (BEx Queries only)</td>
<td>Universe parameters defining key date variable in the universe</td>
</tr>
</tbody>
</table>

Reference: Integrating SAP NetWeaver BW with SAP Business Intelligence Solutions (BO100).

Use Cases for each scenario

<table>
<thead>
<tr>
<th>Scenario 1:</th>
<th>Scenario 2: Data Partitioning by a Characteristic (Account model)</th>
<th>Scenario 3: Data Partitioning by a MultiProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfoCube Contains All Key Figures (Key figure model)</td>
<td>Reporting Scenarios, which have to be extended easily: Additional key figures correspond to additional values of the partitioning characteristic.</td>
<td>Reporting scenario consists of several sub-scenarios: Focus queries use the basic InfoCubes of each sub scenario. Overview queries, which require data from several sub scenarios, use a MultiProvider.</td>
</tr>
</tbody>
</table>

Data Model

<table>
<thead>
<tr>
<th>Use Cases for each scenario</th>
<th>Data Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Closed&quot; Reporting Scenarios:</td>
<td>Data Model derived from the transaction data model in OLTP: All key figures defined as in the transaction data.</td>
</tr>
<tr>
<td>Fixed number of key figures</td>
<td>Logical partitioning of the transaction data by using a suitable characteristic: less complex fact table, but higher</td>
</tr>
<tr>
<td>Homogeneous data granularity</td>
<td>Design of basic InfoCubes reflects the sub-scenarios of the data model: less complex basic</td>
</tr>
</tbody>
</table>
### Performance Optimizing SAP BusinessObjects Reports Based upon SAP BW using BICS Connectivity

<table>
<thead>
<tr>
<th>OLTP system</th>
<th>number of dimensions compared to scenario 1</th>
<th>InfoCubes</th>
</tr>
</thead>
<tbody>
<tr>
<td>- complex and large fact table</td>
<td>- use partitioning characteristic, if it exists in OLTP.</td>
<td>- smaller fact and dimension tables</td>
</tr>
<tr>
<td>- number of dimensions small compared to scenario 2</td>
<td></td>
<td>- Re-use of basic InfoCubes in several MultiProvider.</td>
</tr>
</tbody>
</table>

#### 7.1.5 Using Hierarchies

##### 7.1.5.1 OSS Note 584216- Table RSDRHLRUBUFFER

For each hierarchy-child selection an own table is created (/BI0/02*). These tables are stored within the temporary buffer table RSDRHLRUBUFFER. If there are many different hierarchies or some very big hierarchies available in the system, it is possible that the entries within this buffer are changing rapidly.

There is a possibility to increase this buffer of hierarchy-selections in order to improve their reusability. This can be done within the hierarchy-buffer-table RSDRHLRUBUFFER. The number of hierarchy-selections, which are stored in this buffer, can be changed via the RSADMIN-parameter RSDRH_BUFFER_SIZE (OSS Note 584216). The standard buffer size is 200 entries. The parameter value is only taken into account if it exceeds 200. A useful value range is between 200 and 1000. You can use report SAP_RSADMIN_MAINTAIN to manage entries in table RSADMIN.
Example: If there are 90% of the entries with a timestamp of today or yesterday, or are there more than 170 entries included, an increase of the stored entries (i.e. \texttt{RSDRH\_BUFFER\_SIZE}) would be prudent.

7.1.5.2 **OSS Note 738098 – Performance problems with hierarchies**

Provided no other settings are made, a "remaining node" is automatically added to hierarchies. All characteristic values that are not in the hierarchy are attached to this "remaining node". If the characteristic has a lot of attributes, but only a fraction of these are in the hierarchy, this "remaining node" has a lot of leaves and the internal hierarchy display is very large. This can result in memory problems and longer runtimes. In general, a hierarchy should not contain more than 100,000 entries.

You can deactivate the function that adds the "remaining node" to a hierarchy (in the hierarchy maintenance: go to the hierarchy attributes; select "Suppress 'unassigned' node"). When the remaining node is suppressed, it is no longer displayed in the query.

7.1.5.3 **OSS Note 654243 – Processing very large hierarchies**

Hierarchies that have more than 100,000 nodes are not officially supported with regard to performance and handling. However, a customer may use them if s/he accepts the performance losses.

The following measures may improve the situation partially:

- You may adjust profile parameters of the application server to avoid dumps (\texttt{TSV\_TNEW\_PAGE\_ALLOC\_FAILED}). However, significant performance bottlenecks may still occur.
- If this is possible within a given scenario, you may also create the relevant hierarchy aggregates to speed up processing in the query.
- The attribute for displaying the remaining nodes should be set with a large amount of master data when you load large hierarchies or basic characteristics. This is to ensure that the attribute values that are not contained in the hierarchy are not additionally displayed in the query under the remaining nodes.

7.1.6 **Infocube on-going activities**

7.1.6.1 **Report SAP\_INFOCUBE\_DESIGNS**

Running report SAP\_INFOCUBE\_DESIGNS shows the database tables of an InfoCube, the number of records in these tables and the ratio of the dimension table size to the fact table size. If dimension tables are too large then they can cause badly performing table joins on database level. As the data volume grows and the data allocation changes over the time, this check should be executed regularly.

When loading transaction data, IDs have to be generated in the dimension table for the entries. If you do have a large dimension, this number range operation can negatively affect performance.

In the InfoCube star schema, a dimension table can be omitted if the InfoObject is defined as a line item. This means that SQL-based queries become easier. In many cases, the database optimiser can select better run schedules. However, this also has one disadvantage: You cannot include additional characteristics in a dimension that is marked as a line item at a later date. This is only possible with normal dimensions. If a dimension table has more than one characteristic with a high granularity then consider placing these characteristics into separate dimension tables. For example, a line item is an InfoObject (e.g an order number) for each of whose features one or a few facts in the fact table of the InfoCube are listed.
Table: Performance Optimizing SAP BusinessObjects Reports Based upon SAP BW using BICS Connectivity

<table>
<thead>
<tr>
<th>InfoCube</th>
<th>Rows</th>
<th>Density</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0FIAA_C02</td>
<td>6,773,817</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C02</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C02</td>
<td>8</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C02</td>
<td>40</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C02</td>
<td>13</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C02</td>
<td>164,262</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C02</td>
<td>207,882</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>130,669</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>9</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>32</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>13</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>74,631</td>
<td>57%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>93,278</td>
<td>71%</td>
<td></td>
</tr>
<tr>
<td>0FIAA_C03</td>
<td>2,631</td>
<td>2%</td>
<td></td>
</tr>
</tbody>
</table>

Example of SAP_INFOCUBE_DESIGNS report – InfoCube 0FIAA_C03 has 2 dimensions that require attention.

For good query performance, the number of rows in the dimensions tables should be small compared to the number of rows in the fact table. And, as a general rule, a dimension table should not be greater than 10-15% of the fact table.

Guidelines on how to limit the number of records in dimension tables:

- If an InfoObject has almost as many distinct values as there are entries in the fact table (high granularity), define the dimension of the InfoObject as a line item dimension.
  
  If defined in this manner, the system will write the data directly to the fact table (a field with the data element RSSID, which immediately shows the SID table of the InfoObject, is written in the fact table) instead of creating a dimension table that has almost as many entries as the fact table.

- Only group related characteristics into one dimension. Unrelated characteristics can use too much disk space and cause performance problems (for example, 10,000 customers and 10,000 materials may result in 100,000,000 records at its worst).
  
  Examples of related characteristics: Plant and Company Code (Plant belongs to a Company Code); Material and Material Group (but of course Material is also eligible for Line Item dimensions in case of high cardinality)

- If you cannot avoid characteristics with a high granularity and most of your queries do not use these characteristics, create an aggregate that stores summarised information. Do not use characteristics with a high granularity in this aggregate.

- Please note that the Line Item flag can have negative performance impact on F4 help usage (for which the setting 'Only Values in InfoProvider' is used (transaction RSD1 → Tab 'Business Explorer').)

- It is also worthwhile to try the checks in transaction RSRV. Use for example RSRV → All Elementary Tests → Transaction Data → Entries Not Used in the Dimension of an InfoCube.

7.1.6.2 Compression

InfoCubes should be compressed regularly. Uncompressed cubes increase data volume and have negative effect on query and aggregate build performance. If too many uncompressed requests are allowed to build up in an InfoCube, this can eventually cause unpredictable and severe performance problems. Basically, the F-fact table is optimised for writing (upload) and the E-fact table is optimised for reading (queries).
In the above example (also from report SAP_INFOCUBE_DESIGNS) there are NO entries in the E-fact table i.e. the InfoCube is not compressed

A well run and high performing BW system is not possible unless there is regular compression of InfoCubes and aggregates. A regular compression strategy should be defined with the business data owners. In line with the requirements of the business process, data in the F-fact table should be compressed regularly.

7.1.6.3 Create Aggregates

The main limited resources on the database server are:

• CPU capacity
• Memory capacity
• Number of physical I/O operations that can be performed efficiently

Expensive SQL statements read too much data from the database compared to what is needed in the query. Expensive SQL statements can have a negative impact on performance overall. Improper or unnecessary use of the database buffers results in displacements of other data blocks, and affects overall system performance. A query from a poorly tuned aggregate, even one with adequate performance, may adversely affect the performance of other queries in the system.

Example of InfoProvider which requires an aggregate (Source: Table RSDDSTAT_DM) → 72,424 records selected but only 32 records transferred.

If aggregates are missing, the user who started the query must wait. Even if the user is not dissatisfied with the runtime of a long-running query, it monopolizes resources that are shared between all users. The ratio between rows selected and rows transferred indicates potential performance improvements with aggregates. It should be investigated, which queries are performing badly; and then, which of these can be tuned by aggregates.

In general, aggregates should be considered where the ratio of records selected (DBSEL) to records transferred (DBTRAN) is greater than 10.

7.1.6.4 Delete aggregates

There are several reasons, why an aggregate might be unnecessary:

• There can be very similar aggregates that might be combined into a new aggregate.
• There can be aggregates, which are never used and which are no basis aggregates.
• There can be aggregates with an insufficient reduction factor compared to the InfoCube.
Performance Optimizing SAP BusinessObjects Reports Based upon SAP BW using BICS Connectivity

Example of Aggregate built for Infocube ZFI_I_003 (Source: Table RSDDSTATAGGR)

Valuation’ is the system’s best guess of judging how good an aggregate is. Usually, if the summarization ratio is high, the number of ‘+’ signs in the valuation will be more. The number of ‘-’ signs however indicate that it is not being used at all (Source: Transaction RSDDV). Therefore, in above example, Aggregate 100006 is NOT being used.

As aggregates contain redundant data and are especially created for performance reasons, unnecessary aggregates need disk space and have to be regularly maintained via roll-up and/or change run. Check the usage of your aggregates regularly. It can be the case that the merging of similar aggregates is possible. Some aggregates may not be used at all, or some aggregates could be similar in size to the InfoCubes and lead to no performance improvement.

7.1.6.5 Index maintenance and monitoring
The type of indexes and the administration possibilities can be quite different in SAP BW compared to SAP ERP. Assign responsibility for the creation, maintenance and monitoring of indexes. Responsible team members should have a good understanding of the performance values of indexes, but also the maintenance costs. They should also have a good awareness of the data flows and should check for missing and degenerated indexes on a regular basis. Tools and logs, which can be used to detect degenerated indexes include:

- Transaction RSRV
- Transaction RSA1: Right click on the InfoCube → Manage → Performance tab
- Transaction DB14 → Log for statistics collection
- Transaction DB02 → Missing indexes

7.1.6.6 Archive Data and Near Line Storage
Without archiving, unused data is stored in the database and DataStore objects and InfoCubes can grow unrestricted. This can lead to a deterioration of general performance. The benefits of BW archiving include:

- Reduction of online disk storage
- Improvement in BW query performance
- Increased data availability as rollup, change runs and backup times will be shorter
- Reduced hardware consumption during loading and queries
Near-line storage is a new category of data persistency that is similar to archiving. Using an NLS database enables you to transfer historical read-only data of your InfoProviders (for example, InfoCubes and DataStore objects) to an additional NLS database. However, unlike archiving, you can still access the separated data in the NLS database transparently from an SAP NetWeaver BW system.

7.1.7 Consistency checks

7.1.7.1 Transaction RSRV

This allows you to analyse and repair BW objects. With it, you can check your complete data model from both a BW functional and technical perspective. It guarantees the referential integrity of the star schema models for your InfoCubes, verifies that all underlying tables are correct, and that the data model is consistent from a technical and an administrative perspective. Additionally, you can check many of the underlying database tables, database parameters, and system parameters. It has both functional and technical checks:

- Functional checks include:
  - Master data: Data structure and value consistency
  - Transaction data: InfoCube Dimension, database, and data consistency; DSO object key relationships; etc.
  - Hierarchy: Definition and data consistency
  - Aggregates: Definition and data consistency
  - PSA

- Technical checks include:
  - Database parameters and partition integrity
  - Database indexes, statistics, database, and star schema information

7.1.7.2 Report ANALYZE_RSZ_TABLES

The report is designed as a check-tool for detecting and solving different types of inconsistencies in the main query definition database tables. It is useful to review this report to gain insight into query performance issues. The ANALYZE_RSZ_TABLES report performs checks in the tables that contain the elements of the query definition and query components. These tables are listed here:

- RSRREPDIR Directory of all reports
- RSZCOMPDIR Directory of all reporting components
- RSZCOMPIC Assignment reusable component \(\rightarrow\) InfoCube
- RSZGLOBV Global Variables in Reporting
- RSZELTDIR Directory of the reporting component elements
- RSZELTXREF Directory of query element references

Inconsistencies in customizing RSZ* tables may result in functional and performance issues for many critical queries in the system.

Example extract from Report Analyze_Rsz_Tables showing red alerts.
7.1.8 OLAP cache

One way to increase the performance in certain situations is to use the OLAP cache. When you use the OLAP cache, you can save results, which were determined by directly executing a query, in such a way that they can be used for multiple executions and users, this increases the performance. It is recommended to use the OLAP cache. The usage of the OLAP cache helps to avoid additional workload on the database server significantly. The aim of the cache is to provide the result of a query much quicker out of the cache then requesting it again from the database. After a query is run for the first time, the result is saved in the storage systems of the OLAP cache. Even subsets of queries, which have been run previously, can be extracted from the cache by the OLAP processor to attend to a query.

OLAP cache settings can be made at various levels:
- Transaction RSRCACHE

This setting ensures that all queries use OLAP cache by default, but the settings can be overwritten at both the InfoProvider and/or query level.

The above example indicates that the Global Cache is active
• InfoProvider level (Transaction RSDIPROP → select InfoCube)
This setting ensures that all queries, using the InfoProvider, will initially default to the settings of the InfoProvider. These settings can be overwritten at the Query level.

**InfoProvider: Edit Properties**

<table>
<thead>
<tr>
<th>InfoProvider</th>
<th>ZFI_T_003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>TM1 Top Down(2012)</td>
</tr>
</tbody>
</table>

**Query / Cache**

- **Read Mode**: Query to Read When You Navigate or Expand Hierarchies
- **Data Integrity**: All InfoProviders up to released status (RQTS0)
- **NLS Usage**: Near-Line Access Switched Off (Default)
- **Cache Mode**: 5 Blob/Cluster Added
- **Update Cache Objects in Delta Process**: Checked
- **Delta Cache Class**: CL_RSD_DC_SUPPORT_INFOCUBE

Query level (Transaction RSRT → select query → Properties). Any settings made at this level only apply to that specific query.

**Query Monitor**

- **Query**: ZFI_M_003/ZFI_M_003_BOOKING_Q001

**Query Properties**

- **Read Mode**: Checked
- **Cache Mode**: 5 Blob/Cluster Added
- **Update Cache Objects in Delta Process**: Checked

**Cache Mode – best practice**
Please consider start using the latest SAP cache mode on your queries. Cache Mode 5 (BLOB/Cluster Enhanced) is a best practice since its release, being faster and more reliable than the other modes (1-4).
Filling OLAP cache

Another possible point of performance improvement is to 'warm-up' the OLAP cache of the query while the system is in a 'cold use' time. Pre-filling OLAP cache allows pre-calculation of the queries and the cache is filled with the data. When the users execute a query, which has OLAP cache setting enabled, the execution time is reduced considerably, as the pre-calculated data is already stored in cache and there is no need to hit the database.

This option is available in BEx Broadcaster using option → Fill OLAP Cache

7.1.9 Basis monitoring

The normal basis monitoring also has to be performed on a BW system, including (among others) the following logs and monitors:

- Operation System Monitor (transaction ST06N)
  
  This monitor provides an overview of the current CPU, memory, I/O and network load on an application server instance. You can also access a 24-hour overview of the hardware/OS load. Important indicators are CPU utilisation, memory pages out / in (indicates swapping, depending on operating system), disk utilisation (indicates I/O bottlenecks) and LAN throughput (indicates network problems). The CPU utilisation should be less than 70%; not more than 20% of the physical memory should be swapped.

- Tune Summary (transaction ST02)
  
  Ensure that the instance parameters are set as recommended in SAP Note 1044441. The single-record buffer usually contains BW master data tables. The generic buffer contains most BW-specific control tables (RS*-tables). The table buffer call statistics can be monitored in transaction ST10. It shows buffer type, usage, and invalidations. In case master data tables get very large, investigate the recommendations provided in SAP Note 550784 regarding the deactivation of the table buffering. The Export/Import shared memory contains the OLAP query cache. Note that the Export/Import shared memory does not swap. Usually, the Export /Import buffer is also used intensively in BW systems. Ensure that the Export/Import buffer as well as the Export/Import shared memory are large enough.
Example of transaction ST02 with heavy buffer swaps which can lead to a decrease in the overall system performance.

- Database Monitor (transaction ST04)
  The database monitor checks important performance indicators in the database, such as database size, database buffer quality, and database indexes. The detailed functionality depends on the underlying database management system.
  - Data buffer quality (should be usually at least > 90%)
  - Check exclusive lock-waits for detecting deadlocks
  - Check DB log for special DB problems
  Ensure that you have configured your database according to the SAP Note relevant to your database management system. SAP Note 1044441 provides an overview of relevant SAP Notes for all database management system.

- Table/Index Overview (transaction DB02)
  The Table/Index Overview (direct access with transaction DB02 or using transaction ST04) reports on the table spaces, tables, and indexes. You can see the space statistics (total and free space) and its temporal growth, and you can check for missing indexes (be sure to refresh the space statistics regularly).

7.1.10 BW Accelerator

BW Accelerator can be used to improve the read performance of BW queries on InfoCubes. It enables quick access to any data and is especially useful for sophisticated scenarios with unpredictable query types, high data volume and high frequency of queries. In addition, it reduces the maintenance effort for:

- Maintaining Aggregates
- Automated caching, and
- Building database indexes

In general, query processing time comprises three layers:

- Data access (including aggregation), [Data Manager layer]
- Calculations [OLAP layer]
- Client rendering [Front End layer]

a) The overwhelming majority of query performance issues are due to long data access and aggregation times and this is what BWA principally addresses.

<table>
<thead>
<tr>
<th>Performance Situation (source: Table RSDDSTAT_DM)</th>
<th>Without BWA</th>
<th>With BWA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected rows (DBSEL) are very high</td>
<td>Define aggregates (materialized views, indexed views, summary tables), especially using filters</td>
<td>[no problem]</td>
</tr>
<tr>
<td>Transferred rows (DBTRANS) are very low</td>
<td>Define aggregates</td>
<td>[no problem]</td>
</tr>
</tbody>
</table>

b) And, BWA7.20 with BW7.30 provides a significant improvement for queries that are also slow due to excessive calculation (OLAP) time.
7.1.1 BW on Hana

Migration of your BW system from a conventional database onto the in-memory SAP HANA database will provide further benefits compared to BWA:

- With HANA Optimized InfoCubes, data loading is quicker, which results in an earlier availability of data for reporting. There are no more dimension tables. The development time is reduced because time is not spent in modelling dimensions and categorising them into line item or high cardinality. Since SAP HANA has all the features of BWA, no additional accelerators are required. No additional changes are required like changing queries, MultiProviders and other dependent processes. In the end, a compression is not required anymore since there is no need for E tables and extra storage space to store compressed data.
- The compression due to columnar storage in SAP HANA and the elimination of redundant data shrinks database size dramatically. Depending on the cardinality of the data, you should expect at least a 5x to 10x compression factor.
- There is no need to build aggregates, manage indexing, or spend excessive amounts of time tuning queries.
7.2 Performance evaluation

7.2.1 Query statistics: Transaction RSDDSTAT

There are three main processing areas during query execution:
- Front-end,
- OLAP processor, and
- Data selection (which includes Database and BW Accelerator).

When analyzing query performance, first identify where the performance problem is located and focus on the tuning possibilities for that processing area. In order to do so, use transaction RSDDSTAT to switch on statistical recording for single objects like queries, web templates, workbooks or whole InfoProviders. Only when statistics are switched on, runtime information in tables RSDDSTAT_OLAP and RSDDSTAT_DM will be recorded:
- The RSDDSTAT_OLAP view contains the data from the events from the areas for the front end and calculation layer of the analytic engine
- The RSDDSTAT_DM view contains the data from the events from the area for the aggregation layer and analytic engine

The level of detail can be influenced by a parameter. It is possible to define a default value for all objects. You can also find the find Transaction RSDDSTAT from RSA1 via menu navigation -> Tools -> Settings for BW statistics:

For the purpose of BW query runtime statistics, the system can collect data for the following types of BW objects:
- BEx Query
- BEx Web template
- BEx Workbook
- InfoProvider
On the Maintenance of Statistics Properties screen (transaction RSDDSTAT), you can edit the statistics properties for individual objects as well as the default settings for the object types listed above.

The runtime reading starts with the first user action (for example with the initial execution of a Web Intelligence report) and finishes, when the session is ended by the user (log out). All times for this session are saved under the same SESSIONUID.

This can involve numerous user actions (such as navigation steps, updating of Web template, planning). Each user action is defined as a step and generates a new STEPUID if it contains events relevant for the statistics. The times for the event are summarized under this UID.

Events within a step are assigned to the relevant context (such as Front End, OLAP, DataManager):

<table>
<thead>
<tr>
<th>EVENT ID</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500 to 2530</td>
<td>OLAP Cache times</td>
<td>Includes read time for Cache entries, Counts of the Read Accesses to the Cache and Counts of the Write Accesses to the Cache</td>
</tr>
<tr>
<td>3000 to 3999</td>
<td>OLAP times</td>
<td>Includes the time that is needed to check the query definition, the time that is needed to group together the data requests to the data manager or to read the OLAP cache, the time taken to start the query, read the query definition, execute the variable exists, and replace the variables</td>
</tr>
<tr>
<td>4000 to 4999</td>
<td>Times for Authorisations</td>
<td>Includes the time taken to determine the authorized individual values and intervals, authorized hierarchy nodes, time taken to buffer the authorization data</td>
</tr>
<tr>
<td>6000 to 6013</td>
<td>Times for Input Help</td>
<td>Includes the time taken to read the records from the database, the time taken to execute the hierarchy node value help</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>9000 to 9011</td>
<td><strong>Data Manager</strong>&lt;br&gt;Includes the time in the data manager if the data manager is called from the OLAP, the number of transported records, aggregated from all read accesses (counter event) vs the number of read records, aggregated from all the read accesses (counter event)</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td><strong>Remote Call of a BEx Function Module</strong></td>
<td></td>
</tr>
<tr>
<td>13001 to 14505</td>
<td><strong>BI Consumer Services (BICS)</strong>&lt;br&gt;Includes Java time to load the result set, ABAP time to load the result set, time for synchronization in case of status change (navigation) from Java to ABAP when loading, time for master data access, etc.</td>
<td></td>
</tr>
</tbody>
</table>

Table **RSDDSTAT_OLAP**

**Analysis of runtimes in view RSDDSTAT_OLAP**
In the above example, the query took 4.5 seconds for OLAP and...

0.59 seconds for DATAMANAGER

Table RSDDSTAT_DM

In the above example, 48,314 records were selected and 14,212 were transferred without using an aggregate.
7.2.1.1 High OLAP time
The high OLAP time means that the BEx query spends a large part of the runtime in the OLAP processor. So this query needs to be looked at design level. This query is selecting a large number of records and then a high number of cells are transferred to the front-end.

- Check if you need to perform all the calculation/formatting being performed.
- Have your query designers/developers investigate if there is a more efficient way of to do the calculations. Some calculations can be anticipated in the data staging.
- Please consider that a special calculation, which is done by the OLAP processor might be very time consuming and CPU intensive. Additionally, those features can have a negative impact on the use of aggregates.
- Avoid extensive usage of “cell calculation”.
- Very complex authorization concepts can have a large impact on the query runtime.

7.2.1.2 High DataManager time
The high DataManager time means that the BEx query spends most of the time on the database. The following tuning measures may help:

- If there are unselective accesses to the database causing the long runtime, the filter set in the Web Intelligence report is not restrictive enough (e.g. data is requested for millions of assets, etc.). In this situation redesigning the Web Intelligence report in regards to the Mandatory Filters can help improving the performance.
- If the part provider is a basic InfoCube, the read time is high (column "READ[s]"), and the selected to transferred row ratio (DBSEL/DBTANS) is HIGH, building aggregates for the involved InfoCubes can help to reduce the database runtime.
- If query is on the DSO object, trace the Query and check for the possibility for index creation on the expensive statement.
- There are also additional factors like statistics and indexes of the InfoCube, InfoCube compression, parallel processing of the query etc., which affects query performance.
- To improve database runtime consider to implement BW Accelerator.

7.2.2 Query Monitor: Transaction RSRT
RSRT is also another useful tool for query analysis. It should be used to investigate individual queries and their executions/navigations in detail and to check whether query executions spend their time on the database, in the OLAP processor or in the front-end. RSRT can be used to get a general technical impression of a query definition and the usage of performance relevant features.
7.2.2.1 Technical information

From the technical information screen, you get an overview of important, technical, reference numbers for a query.

The technical information is grouped by category for the selected query. Depending on the query definition, all or parts of the categories are displayed. The first line of the HTML control contains links to the categories that are displayed in table form in the lower area of the page.

System entries are specified in the first column for some of the categories:

- The symbol 🟢 means that the value of the technical reference numbers corresponds exactly to the system recommendation.

- The symbol ⚠️ means that the value of the technical reference numbers might not correspond to the system recommendation.

- The symbol 🔴 means that the value of the technical reference numbers does not correspond to the system recommendation. In this case, changes may be useful. This does not necessarily mean that an error exists.
From the Technical Information screen, you get an overview of important technical reference numbers for a query.

### 7.2.2.2 Query properties

Some query settings can only be made in RSRT - From the Query Properties dialog box you can make settings for a BEx query with regard to the read mode, the cache mode, the selection of structure elements, the optimization mode and the calculation accuracy:

**Read mode** - the read mode determines how the OLAP processor gets data during navigation
- (H) Query to be read when you navigate or expand hierarchies. The amount of data transferred from the database to the OLAP processor is the smallest in this mode. However, it has the highest number of read processes.
- (X) Query to read data during navigation. The OLAP processor only requests data that is needed for each navigational status of the query in the Business Explorer. The data that is needed is read for each step in the navigation.
- (A) Query to read all data at once (A). There is only one read process in this mode.

The read mode (H) improves performance in almost all cases compared to the other two modes. The reason for this is that only the data, which the user wants to see, is requested in this mode.

**Cache mode** - the cache mode defines whether and how the query results and navigational states calculated by the OLAP processor should be saved as highly compressed data in a cache:

<table>
<thead>
<tr>
<th>Property</th>
<th>Main Memory Cache</th>
<th>Query Aggregate</th>
<th>Main Memory Cache with Swapping</th>
<th>Persistent Cache per Application Server</th>
<th>Cross-Application Server Persistent Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Mode</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Displacement</td>
<td>Yes (LRU algorithm and deletion)</td>
<td>No</td>
<td>Yes (LRU algorithm and displacement)</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
7.2.2.3 Execute and debug

The Query Monitor (transaction RSRT) allows you to execute queries and to trace queries in a debug mode with several parameters (e.g. do not use aggregates, do not use buffer, show SQL statement). In the debug mode, you can investigate if the correct aggregate(s) is/are used and which statistics the query execution generates. To check the reasons, you can switch off the usage of aggregates, and switch to no parallel processing (for more details read in the MultiProvider section) or display the SQL statement and the run schedule. Moreover, you can set some performance-related query features in RSRT: read mode and OLAP query cache mode. If you discover single queries that you wish to improve in terms of performance, you should execute them in RSRT in the debug mode, and analyse which aggregates are used and which statistics are generated.

7.2.3 Trace Tool RSTT

This tool allows you to record all the actions from a BEx frontend session along with their timings. This tool can be very useful identifying which steps within BEx process takes the longest time.
- RSTTT should be used to investigate individual query executions including variable input in the variable selection screen and/or navigational steps.
- RSTT can be used to trace all navigational steps and variabes, (even from other users) having the possibility to replay the trace

In order to set up the trace follow the following steps to record and then view an RSTT trace:
1. Click on the Trace Tool Menu in the left navigation, then click on User Activation, then enter your user id, and then click the "Activate" button. This will activate your user id for RSTT tracing.
2. You will notice a row added to the Trace User table. This indicates that the tracing is active for that user.

3. Next, log on to the BEx Designer and execute the query that you want to trace. When you have completed your actions in the BEx Designer, go back to the RSTT transaction and click the "Deactivate" button to deactivate tracing for your user id.

4. Click on Traces in the left hand navigation, click on the newest trace, which should be at the top of the trace history list, and then click the Display button. This will display your trace results.
5. The results of your trace are now visible. The sequence of programs that were called is listed. The runtime for each step is listed. This can be useful in order to identify the biggest contributors to overall time. Note: You can also highlight a row and click the Parameters button to see the values passed to the program to get a better idea of what is going on.

7.2.4 Comparing BW BEx Query runtime to BO Web Intelligence report

In order to determine where the problem could exist (i.e. either in SAP BW or in SAP BO), it is important to ensure that the runtimes collected are comparable. In essence, the output from a BEx query should be similar to the output in the Web Intelligence report (i.e. the number of cells output in each case should be similar)

For example, in a recent investigation, a client complained that the runtime of a report in Web Intelligence took 50% longer to run. The client provided below statistics as evidence:

<table>
<thead>
<tr>
<th></th>
<th>Elapsed Time to Refresh Report (Seconds)</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEx</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>WEBi</td>
<td>15</td>
<td>50%</td>
</tr>
</tbody>
</table>

The underlying reports run to collect the above statistics are:
a) BEx query
Performance Optimizing SAP BusinessObjects Reports Based upon SAP BW using BICS Connectivity

The runtime of 10 seconds collected related to a BEx query, which had 1 characteristic and 4 key figures in the output.

b) Web Intelligence

The runtime of 15 seconds collected related to a WEBi report which had many characteristics (and included a hierarchy) compared to the BEx query above.

Once the BEx query had been changed to retrieve the same data as the WEBi report, it was found that the runtime had increased for the BEx query:

<table>
<thead>
<tr>
<th>Elapsed Time to Refresh Report (Seconds)</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEx</td>
<td>13</td>
</tr>
<tr>
<td>WEBi</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>13%</td>
</tr>
</tbody>
</table>

By establishing the correct runtime comparison, it is possible to focus on the appropriate areas when investigating further. Consequently, the approach should be as follows:

- Check if output contains the exact same amount of data
- Check if the same selections/variables are used.
# Reference Documentation

The following documents are referenced in this document:

<table>
<thead>
<tr>
<th>DOCUMENT NAME</th>
<th>LOCATION</th>
<th>ISSUE DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP BusinessObjects increasing stability by setting limits on max. retrievable cells from SAP BW into Web Intelligence using BICS.</td>
<td><a href="https://scn.sap.com/docs/DOC-31900">https://scn.sap.com/docs/DOC-31900</a></td>
<td>20-09-2012</td>
</tr>
<tr>
<td>How to create efficient MultiProvider queries.</td>
<td><a href="http://www.sdn.sap.com/irj/scn/go/portal/prtport/docs/library/uuid/b03b7f4c-c270-2910-a8b8-91e0f6d77096?QuickLink=index&amp;overrideayout=true">http://www.sdn.sap.com/irj/scn/go/portal/prtport/docs/library/uuid/b03b7f4c-c270-2910-a8b8-91e0f6d77096?QuickLink=index&amp;overrideayout=true</a></td>
<td>06/2006</td>
</tr>
<tr>
<td>Integrating SAP NetWeaver BW with SAP Business Intelligence Solutions (BO100).</td>
<td>SAP Education course BO100.</td>
<td>2011</td>
</tr>
<tr>
<td>Wily Introscope Setup for SAP BOE 4.0</td>
<td>SAP Note 1540591</td>
<td></td>
</tr>
<tr>
<td>Added memory usage when splitting Adaptive Processing Server (APS) services.</td>
<td>SAP Note 1640036</td>
<td></td>
</tr>
<tr>
<td>BI 4.0 Consulting:- How to size the Adaptive Processing Server</td>
<td>SAP Note 1694041</td>
<td></td>
</tr>
<tr>
<td>Selecting the Right SAP BusinessObjects BI Client Product Based on your Business Requirements for SAP BW Customers</td>
<td><a href="https://scn.sap.com/docs/DOC-32449">https://scn.sap.com/docs/DOC-32449</a></td>
<td></td>
</tr>
</tbody>
</table>