

# Performance & Scalability Best Practices and Tuning Recommendations for SAPBO XI 3.1 on BW



## Applies to:

SAP BusinessObjects XI 3.1 SP2 or greater, NetWeaver 7.01 SP5 or greater

## Summary

This whitepaper summarizes performance and scalability related best practices, tuning and configuration to help make your SAP BusinessObjects/BW deployment perform at its best. There are plenty of additional whitepapers which delve into significant detail – this whitepaper is meant to be a handy summary for an administrator who is familiar with both SAP BusinessObjects XI 3.1 and NetWeaver 7.01

**Author:** Tom Turchioe with contributions by Jay Thoden Van Velzen and Ingo Hilgefort

**Company:** SAP

**Created on:** 20 August 2010

## Author Bio



Tom Turchioe works for SAP as the Product Director responsible for SAP BusinessObjects Platform and Client Tool performance and scalability. He joined the company in 2003 where he had the same responsibility for BusinessObjects e6.x and key components of XI. He's also worked in the field helping customers with architecture, deployment, and performance and scalability related topics.

## Table of Contents

Prologue.....	3
Background.....	4
Detailed Best Practices and Tuning Recommendations .....	4
SAPBO Specific Best Practices Including WebIntelligence .....	4
Universe General Best Practices – Creating and Updating Objects .....	4
Web Intelligence General Best Practices .....	5
Web Intelligence via Data Federator Universes .....	5
Four Tips for Optimizing Performance .....	6
Use Query Drill Option Where Possible .....	6
Scheduling Vs. On-Demand Reporting .....	6
Query Drill for Hierarchies.....	6
Static Filtering with Web Intelligence Query Filters .....	6
Use Inclusive Member Filters Rather than Exclusive Ones .....	6
Dynamic Filtering with Web Intelligence Filters.....	7
Large LOVs for Prompting .....	7
Reports with High Data Volume .....	7
Reducing the Number of Rows per Request by Using Guided Navigation .....	8
Creating Queries for Master Data Reporting.....	9
Optimizing Detail Object Syntax.....	9
Adding Keys to Objects Used in an LOV for Filtering.....	9
Xcelsius with SAP BW - SAPBO Specific (related to Xcelsius with SAP BW - BW specific in the next section).....	9
Review the Use of "Query on Drill" Option .....	10
Consider Removing Unused or Redundant Detail Objects .....	10
Optimize Object Syntax .....	10
List of Values (LOV) Best Practices .....	10
BW specific best practices .....	11
Check Performance of BEx Queries .....	11
Navigational Attributes - 1 of 2.....	11
Static Filtering with BEx Query Restrictions .....	11
Navigational Attributes - 2 of 2.....	11
BI Performance Tuning - Impact of Data Modeling .....	11
Navigational Attributes .....	12
Restricted and Calculated Key Figures.....	12
Xcelsius Against SAP BW (related Xcelsius against SAP BW - SAPBO Specific in previous section). This Section Contains the BW Specific Considerations When Using Xcelsius .....	13
SAPBO specific tuning and configuration .....	13
Recommended SBOP and BW Patch Levels.....	13
BW Specific Tuning and Configuration .....	14
Set Specific RSRT Property.....	14
BW via ODA - Why 7.01 SP3?.....	15
Activate OLAP Cache and Delta Cache.....	15
SAP Notes on BW Performance .....	16
Related Content.....	17
Copyright.....	18

## Prologue

Best practices and tuning only help within the context of design choices already made. For example, if a document retrieves gigabytes of data through SAP BusinessObjects (SAPBO) WebIntelligence on an OLAP universe, for export to Excel and further analysis, no amount of tuning or optimization will result in a deployment that performs well, nor scales well.

The reader is strongly encouraged to perform a "sanity check" - are you doing something that makes sense? Are you using the SAPBO tools in the way they are intended? Are there performance and scalability issues which are largely caused by inappropriate design choices, not following good BI practices, misunderstanding of tool capabilities or incomplete/incorrect business requirements? An example would be to deploy WebIntelligence for use cases where BEx Explorer makes better sense - realizing that BEx Explorer will be supported for at least another 5 years as of this writing. One size does not fit all.

We've seen far too often where our BI tools are used to do large data dumps, because that's how it has always been done. For example, labeling 4GB data requests as "business requirements". We've also observed dashboards built with Xcelsius that emulate documents which should be implemented using WebIntelligence because the documents need cross tab drill functionality.

Sometimes, thinking "less is better" is improperly applied in an attempt to eliminate overhead. For example, using direct ORA connectivity to BW DSO tables without Data Federator (SQL, bypassing BW layers) because you assume that Data Federator creates the overhead and might not yield the performance gains you expect when it is really the structure of the DSO tables and possibly metadata issues are the likely source of the overhead.

Before you start tuning and optimizing SAPBO, consider if the design and implementation of the deployment is:

- Advised/recommended/in line with the product roadmap
- Being done in a way where there is better approach using other SAP or SAPBO BI tools
- Follows a "top-down" approach, rather than a "bottom-up" (where IT builds data structures, and BI systems without any knowledge or understanding of business/reporting requirements)

It is our experience that performance and scalability issues raised by our customers are usually due to inappropriate application of our tools/products. We are aware there are some things which could be better in the current product versions, but finding the appropriate tool, thinking about the BI solution you're putting together, and following standard BI best practices (and a "top down" approach) are much more likely to provide significant performance gains with good scalability - much more so than query tuning. Pulling large quantities of data (in GBs) through OLAP universes will not yield satisfactory performance and scalability no matter how much tuning, optimizing, etc. is done.

## Background

We work continuously to improve the performance and scalability of SAP Business Objects Enterprise deployments running against SAP BW. This includes running large scale load tests to characterize the performance and interaction in an SAPBO/BW stack, bug fixes, and enhancements specifically targeted at making SAPBO/BW deployments perform and scale better over time.

A key component of these efforts is effectively communicating the correct and necessary information to our field and customers on how to deploy SAPBO against BW successfully. Without this, our customers would only obtain performance and scalability improvements when we deliver changes in SAPBO and BW software and make changes to OOB (out-of-box) parameters. As we learned from past experience, our customers see maximum benefit in terms of successful deployments with a multipronged approach including software improvements, bug fixes and proper communication/education.

This white paper contains performance and scalability related best practices, tuning and configuration information gleaned from over 200 internal web pages, wikis, documents and PowerPoints as well as a number of interviews with field and services people who have successfully deployed or help deploy SAPBO against BW. There are many articles and wikis on SDN that go into great detail. This white paper is designed to be a handy summary for an administrator who is familiar with SAPBO XI 3.1 and BW 7.01. Keep in mind we continuously learn new approaches, and tuning and configuration recommendations, so you should regularly check SDN for new information.

## Detailed Best Practices and Tuning Recommendations

The information below is organized into 4 areas:

- BOE specific best practices including WebIntelligence - this includes document and universe design best practices which are performance and scalability related
- BW specific best practices - this includes cube design and BEx query design related to performance and scalability
- BOE specific tuning and configuration
- BW specific tuning and configuration
- Miscellaneous recommended SAP notes

### SAPBO Specific Best Practices Including WebIntelligence

This includes document and universe design best practices which are performance and scalability related.

#### Universe General Best Practices – Creating and Updating Objects

- The common best practice is to push as much as possible to the BW query. In some cases, where clauses created at the universe layer can be sent down to BW and sometimes not. When they aren't, restriction of data is occurring at the client (with respect to BW) side and more data than necessary is being moved across the wire to the SAPBO deployment.
  - Objects like a calculated keyfigure or a restricted keyfigure or a filtered measure should be created as part of the BW query. You should always push restrictions and calculations as far down the stack as possible.
  - Especially considering that in XI 4.0 all our tools can connect to the BW query layer directly, you should push the above to the BEx query level.

If there are existing Where clauses at the Universe layer (Universe on BW queries)

- You will need to re-add them if you run an Update of the structure because the Wizard removes the Where clause
  - Always copy an object and hide the original

## Web Intelligence General Best Practices

1. Understand that primarily the WebIntelligence Query Panel definition influences how much data is retrieved from BW, and not the BEx query design or the WebIntelligence Report layout. In XI 3.1 SP3 and greater, query stripping is a new feature when running against OLAP universes. The query issued by WebIntelligence to the OLAP universe will only request the information rendered/presented to the user on the report. So if the query has 10 result objects and the report only shows/uses 5 (i.e. a calculated column relies on an object that is otherwise not displayed) - and query stripping is enabled at the document and query level - the query sent to the OLAP universe will only have those 5 objects. Use caution if the user can edit the document and is likely to add a query panel result object to the document that was previously missing from the document. Doing this will result in another query because that new object isn't in the microcube.
2. In some situations extremely large queries and universes can adversely affect performance
  - a. Use transaction RSRT to get a detailed breakdown
3. When refreshing the data in the WebIntelligence report, the entire microcube is refreshed. In XI 3.1 SP3 and greater, query stripping will reduce the size of the microcube and thus reduce the amount of data that Web Intelligence will fetch from BW.

## Web Intelligence via Data Federator Universes

The best practices in this section also apply to OLAP Universe sources. Keep in mind you will lose SSO (as of this writing (August 2010)) at the BW layer unless you create mirror accounts in BW as well as in Data Federator (DF) with the same password. This should be addressed in the next service pack after XI 3.1 SP3. Query stripping (introduced in XI 3.1 SP3) is not available when running via Data Federator.

When running Web Intelligence against BW via a DF universe, remember to:

1. Reduce the amount of cells being received from the SAP BW system.
  - a. Only include items in the WebIntelligence Query Panel that are being leveraged in the report
  - b. Up to XI 3.1 SP2, WebIntelligence will not automatically optimize the query based on the report definition.
2. Report design considerations
  - a. Leverage report linking (aggregated reports linking to more detailed reports)
  - b. Leverage Query Drill mode for hierarchical reports
3. Master Data Style Reports
  - a. In scenarios where the largest parts of the dataset retrieved are master data (i.e. display attributes) consider breaking up the single query into two
    - i. One for the master data display attributes
    - ii. One for the actual data
  - b. Merge Dimensions functionality can be used to consolidate the dataset
4. Leverage Publications functionality whenever possible to schedule the report refresh in background to reduce end user wait time
  - a. This has been greatly improved and extended in XI 3.1 for WebIntelligence (Multi-Pass Bursting)

## Four Tips for Optimizing Performance

Four tips you should always apply from an end-to-end system perspective

1. Optimize BW Query definition (not the WebIntelligence query panel)
2. Optimize MDX execution on SAP NetWeaver BW (notes)
3. Optimize Universe definition and generation of MDX on BOE (WebIntelligence)
4. Optimize Report definition and generation of MDX on WebIntelligence

## Use Query Drill Option Where Possible

The "Use Query Drill" option in the Web Intelligence Document Properties dialog box may significantly improve drill-down performance

## Scheduling Vs. On-Demand Reporting

One of the first factors to consider when designing a report is whether it is necessary to have the report run on-demand or if the reporting need can be met by having users access scheduled instances of the report. In general, if it is possible to minimize the number of times a report is run against the BW system, it is desirable to do so. So, it is recommended to use scheduling when practical. The primary benefits of scheduling rather than viewing on-demand are:

1. Vastly improved viewing response time for the user
2. Overall reduction in burden on the BW system versus having many ad-hoc queries run.

## Query Drill for Hierarchies

Multiple hierarchies can be defined at the Characteristic level. They will be automatically exposed in the universe. Universe Designer capabilities to define and maintain drill-down paths in Web Intelligence should be leveraged to enable Web Intelligence users to drill down according to BI hierarchies as with any other custom universe hierarchies. It is important to note that the Use Query Drill option that is available in Web Intelligence Document Properties dialog helps to significantly improve the drill down performance. With XI 3.0 release, this option is not activated by default while it is starting XI 3.1 release. This option can make Web Intelligence behave more like BEx in terms of fetching limited result sets when drilling down.

Therefore it must be kept activated when there is an expectation to have drill down performance in Web Intelligence as fast as within BEx.

## Static Filtering with Web Intelligence Query Filters

Defining filters in the Web Intelligence query panel rather than in the underlying BI query provides a lot of flexibility and allows a single BI query and single Universe to be reused for many Web Intelligence reports. By following a few simple guidelines, it is possible to implement quite well-performing queries using static Web Intelligence filters.

However, the general rule is that any prompting should be defined in the BW query already not the Web Intelligence query panel.

## Use Inclusive Member Filters Rather than Exclusive Ones

Avoid using the operators Not Equal To, Not In, Not between, etc. in the filter pane of the Web Intelligence query panel. Due to the need to resolve filters to member-sets, these types of filters do not perform well. When practical (typically in cases where the set of members selected is relatively small), replace these types of filters with the inclusive equivalent. If this is not possible, consider doing the filtering in the BI query instead (see Static filtering with BEx Query restrictions). This recommendation is a BW specific item.

## Filter on indexed values

See next section on Dynamic filtering with Web Intelligence Filters

## Dynamic Filtering with Web Intelligence Filters

When filtering on high cardinality characteristics, in order to avoid the need to resolve member captions to member-unique names when viewing (see Resolving of Web Intelligence query filters to member selection), ensure that any characteristics which are filtered in Web Intelligence are filtered on indexed values. In order to ensure this, three things are required:

1. The object which is being filtered must have a key associated with it. That key must be the "technical name" of the underlying characteristic in BI.
2. Adding keys to objects used in an LOV for filtering for details.
3. The value(s) for the filter must be selected from the LOV, rather than being typed in manually.
4. The prompt must be configured to "Select only from list" in the prompt options dialog.

If all of the above criteria are not met, the value entered or selected will have to be resolved to the member-unique name before the report is run, causing needless overhead. The degree to which this is important varies depending on the cardinality of the characteristic: low-cardinality characteristics will not incur a severe penalty in doing member caption lookups. In any case, doing these lookups will always incur some overhead.

Note: There is one case when the above recommendation to always use technical names as keys for filtering is invalid. When working with multiple Web Intelligence queries in a single document, Web Intelligence will share the prompt for filters in different queries which share the same prompt name. In the case where this sharing is desired and the underlying characteristic being filtered is not from the same InfoObject for both queries, it is essential to not have a key specified for the Universe object being filtered, as doing so will result in the technical name for the first object being used, which will not be a valid identifier for the other object (based on a different underlying InfoObject) being filtered. In the case where this sharing is not desired, it is necessary to simply name the two prompts differently.

## Large LOVs for Prompting

When generating an LOV for prompting on high cardinality characteristics, even retrieving the member set for the LOV can be very expensive. In such cases, the user will commonly have to use the search functionality in the prompt page in order to find the desired values. If it is not necessary to present the user with an initial list to choose from, it may be desirable to enable delegated search for the characteristic in the LOV. This will force the user to enter a pattern to match before any LOV values are returned, and will only request the member set from BI which matches the user's specified pattern.

Delegated search is enabled in the Properties tab of the object properties dialog in the universe Designer.

## Reports with High Data Volume

The OLAP BAPI interface is not suitable to be used for queries which return a high volume of data in a single request. This is due both to internal limitations within the OLAP processor and to the flattening process which occurs before the data can be consumed by Web Intelligence. The volume of data returned can be measured by the number of cells returned. In general, it is desirable to reduce this number to the minimum required for the reporting requirement. This can be done by reducing the number of columns or rows returned in the request.

## Reducing the size of rows by optimizing Web Intelligence queries

### ***Remove unused fields from the query***

This only applies prior to XI 3.1 SP3. SP3 and greater have query stripping which render this unnecessary. A description of query stripping is provided earlier in this document.

While this may seem simple and not obviously necessary, it can have a profound impact. During the process of WebIntelligence query and report creation, it is possible that fields may have been added in the query panel which are not actually used in the report. It is very important to review the query definition before publishing a document, and remove any fields from the query definition which are not actively used or desired to be made available during analysis. Web Intelligence will not optimize the query at run-time to remove those fields which are not required.

It may also be desirable to customize the universe definition to remove any dimension or detail objects which are found to be redundant, avoiding the possibility of a user inserting multiple versions of the same thing into a report.

### ***Refactor queries to extract more constant master data***

When creating reports which contain a lot of rows of data and display a lot of master data columns (typically Web Intelligence Detail objects / BI characteristic properties) which does not change per-row, it is worth considering refactoring a single query into multiple queries to separate the more constant master data from detail records. Note that it is important to weigh the inherent cost of making additional queries against the savings realized by removing static master data from the mass result set. This approach should only be used when the number of unique master data values to be retrieved is at least an order of magnitude greater than the number of detail rows, and the number of master data fields is relatively large.

## Reducing the Number of Rows per Request by Using Guided Navigation

Another approach to reducing the number of cells returned per request, and indeed the total number of cells, is to employ more guided navigation techniques in reporting, rather than presenting the user with both high-level aggregates and details up front. This technique is appropriate when the total set of data exposed by a report is vast, and the user is likely to be interested in all of the highly aggregated data but only specific details. There are two main methods to achieving this: Using Drill in the report, and using report linking.

### **Using Drill**

Drill can be used within your Web Intelligence report as long as you have a hierarchy defined. It is possible to use either BI hierarchies or custom hierarchies defined in the Universe for drill. In order to have only the data for the current drill context fetched, rather than the entire dataset being fetched up-front, ensure that "Use query drill" is checked in the Web Intelligence document properties.

As the query used to process the drill is essentially the same as any other filter request, it is important to use ensure that objects to be used for drill also have an index defined when drilling, as specified in the section Static filtering with Web Intelligence Query filters.

### **Using Report Linking**

As another alternative to using drill, you may choose to use report linking. In this case, you would define an initial report which contained only the highly aggregated levels of data which the user will use to decide where more information is desired. Report linking is much more flexible in that you may define links at any level desired, and reports linked to do not have to maintain the same formatting (or indeed, have much data in common at all with the source report). All that is required is a relationship between the data in the source context and the data in the target.

It is important to follow the same techniques when linking to a report as are followed in the other dynamic filtering cases specified in the section Dynamic filtering with Web Intelligence filters, in order to avoid unnecessary member caption resolution in processing the query for the target report.

## Creating Queries for Master Data Reporting

In order to create efficient Master Data queries for reporting in Web Intelligence, it is necessary to understand what each type of Master Data query will request from the BI system. Following are three high-level types of Master Data queries.

### Query containing a single level of a single characteristic

In this case, master data values will be read directly using OLAP BAPI calls, without executing any MDX (and so without using the OLAP processor). This will result in all members being returned, whether or not there are any corresponding entries in any fact tables for them. The performance of such a query will be good, but may result in retrieving members which are not relevant.

### Optimizing Detail Object Syntax

For queries on BI Universes that include only the key and medium name detail objects of a Dimension, it is possible to modify the generated syntax of the objects to improve query performance, due to some internal details of the OLAP BAPI interface.

### Adding Keys to Objects Used in an LOV for Filtering

As indicated in the sections on filtering, it is important that objects which have a List of Values (LOV) associated with them should always have a key which points to the underlying technical name for the Characteristic values represented by the objects. By default, all Dimension objects automatically created when generating a BI Universe have this defined. In cases where customization has been done or where detail objects are used in this way, this needs to be done manually.

### Xcelsius with SAP BW - SAPBO Specific (related to Xcelsius with SAP BW - BW specific in the next section)

The best practices in this section apply regardless of the data source.

In XI 3.1 SP3, a new set of Xcelsius services were added to the SAPBO platform which markedly improve response time and increase scalability. These services must be enabled at the web tier and SAPBO tier to override the use of WebIntelligence for servicing QaaWS requests from Xcelsius. This new set of services is designed to enable large dashboard deployments using QaaWS. Internal benchmarks have shown up to a 10x increase in scalability. Of course, the results you will obtain depend on solution complexity, hardware, network and database response time. The best results are obtained when data can be shared between multiple dashboard users which enable SAPBO to cache the result sets. No changes are needed for existing dashboards because the Web Service URL remains unchanged - it is simply routed to the new Xcelsius services. Note these new Xcelsius services only function with relational universes (which would include using Data Federator to access BW data). Please refer to the release notes for XI 3.1 SP3 for other limitations.

Also available in XI 3.1 SP2 is the ability of Xcelsius to connect directly to BW via BICS. This completely bypasses SAPBO.

When deploying Xcelsius for the enterprise, other considerations need to be made in addition to all of the guidance for OLAP Universes and Web Intelligence. This is due to the fact that the OLAP Universe is predominantly used as the back-end for Xcelsius dashboards via the Query as a Web Service (QaaWS) component.

With Xcelsius, implementers also need to use specific best practices for Xcelsius as well, to ensure they do not overload the Flash engine, jeopardizing a fast response time for the live dashboards. End users for dashboard applications typically have little patience for poor performing dashboards, so it is critical that implementers bear in mind the following Xcelsius established best practices:

1. Make sure that you only pull in the data that you absolutely need and must display in the dashboard. Xcelsius dashboards run within the Flash engine inside the browser, which is not designed for aggregating data, performing large mathematical calculations, or processing large datasets that require a large amount of memory. The guideline for data volume is to try to keep your dataset ideally to 500 rows; 5,000 rows maximum

2. Be smart in how queries are initiated. If the dashboard has multiple tabs, try to only pull in data for subsequent tabs when those tabs are accessed. Running all of the queries on initial load can delay the moment the dashboard first displays, which may be longer than an end-user is willing to wait. Splitting the queries up between tabs can make the dashboard far more responsive
3. Where a live query is not feasible, LiveOffice is a good option. However, the LiveOffice documents should also be optimized to ensure good dashboard performance.
4. Ensure that the customer is on the latest patch levels for Xcelsius, BusinessObjects Enterprise, and BW. A number of product issues have been fixed, so the latest FixPacks provide all of the resolutions for issues identified earlier. The FixPacks are synchronized and rely on each other, so it is not sufficient to only patch the BOE servers or the BW system alone.

### Review the Use of "Query on Drill" Option

Selecting this option means that each time an end-user performs a drill in the report, a connection to BW is made to retrieve only the necessary data for the drill context. An advantage of this is that the data within the report is kept to a minimum. However, a disadvantage if the connection to BW is slow, and query performance is not optimal, then the end-user will have to wait a long time for the data to return on each drill. An alternative way is to extend the scope of analysis (to the level of data commonly used e.g. L01 to L05) in the query which will allow end-users to drill within the microcube of the report. This way, only one set of retrieval of data from BW is required. If further levels of data are required (e.g. L06), then the end-user can drill outside the scope defined, which would initiate another query to the BW. However, this alternative also requires the volume of data returned in the report to be considered.

### Consider Removing Unused or Redundant Detail Objects

(Note this is not a requirement but something you may consider)

You have the option to remove any detail objects from the Universe which are redundant or unlikely to add any value to reporting, in order to prevent report designing users from including them unnecessarily in queries. This is due to the fact that the universe is parsed prior to executing queries hence smaller more agile universes will parse faster.

### Optimize Object Syntax

For performance reasons, it is recommended that Key objects refer to the NAME attribute. It is recommended that Name objects refer to the DESCRIPTION attribute.

### List of Values (LOV) Best Practices

#### Review LOV association

It is recommended to review the association of List of Values (LOV) to objects. Currently in the universe, all dimension objects has a LOV associated. This should be reviewed to ensure that LOV's are created as necessary, and to ensure large LOV generation is avoided. Where LOV's are required for an object, ensure that:

1. Keys (being the technical name) are defined for the object as this will be used in filtering the report query. Using keys will make the query more performing.
2. Values are indexed on BW. Objects used for filtering in a query should be filtered on indexed values, for better performance.
3. Where possible, use delegated search where the LOV generated would be large. Large LOV will take a longer time to generate and display to the end-user.

#### Reduce LOV Prompt Sets

Many of the Prompts use huge list of dates that require ABAP processing. These dates range from the year Zero (AD) to 9999 (AD) It is highly unlikely that Unilever have dates in all but a small range of these values. It is therefore recommended to reduce the set of dates as part of a cube redesign.

## BW specific best practices

This includes cube design and BEx query design related to performance and scalability.

### Check Performance of BEx Queries

Consult the following note on the Support website:

[http://help.sap.com/saphelp\\_nw70/helpdata/EN/43/e39fd25ff502d2e10000000a1553f7/frameset.htm](http://help.sap.com/saphelp_nw70/helpdata/EN/43/e39fd25ff502d2e10000000a1553f7/frameset.htm)

<https://www.sdn.sap.com/irj/sdn/bi-performance-tuning>

### Navigational Attributes - 1 of 2

A large number of navigational attributes defined in the underlying InfoProvider may impact the overall performance (please refer back to SAP best practices for data modeling). In fact, navigational attributes are used as dimensions in MDX query. This leads SAPBO generating a lot of "Crossjoin" operators (in the MDX statements) and might return thousands of useless rows. The presence of navigational attributes has nothing to do with the SAPBO engine generating crossjoins.

Structures defined in the BEx Query that are based on characteristics are included in the universe as single-dimension objects with the elements of the structure as dimension members.

### Static Filtering with BEx Query Restrictions

When the suggestions in Static filtering with Web Intelligence Query filters cannot be practically followed, it may be necessary to consider altering or duplicating the relevant BEx query to impose the restriction. This has the advantage of always producing the best-performing report, but the disadvantage of added maintenance overhead and the need for potentially more BEx queries and universes to be defined if the filtered values needed are not the same for all consumers of the existing BEx query.

Note: It is not necessary to update your universe after making this change to an existing BEx Query.

### Navigational Attributes - 2 of 2

Characteristic attributes can be converted into navigational attributes. They can be selected in the query in exactly the same way as the characteristics for an InfoCube. In this case, a new edge/ dimension is added to the InfoCube. During the data selection for the query, the data manager connects the InfoProvider and the master data table ('join') in order to fill the Query.

From a pure performance point of view, you should model an object on a characteristic rather than on a navigational attribute.

## BI Performance Tuning - Impact of Data Modeling

### Line item dimension and high cardinality

When compared to a fact table, dimensions ideally have a small cardinality. However, there is an exception to this rule. For example, there are InfoCubes in which a characteristic document is used, in which case almost every entry in the fact table is assigned to a different document. This means that the dimension (or the associated dimension table) has almost as many entries as the fact table itself. We refer here to a degenerated dimension. You can use the indicators line item and high cardinality to execute the following optimizations:

**Line item:**

The dimension contains precisely one characteristic. This means that the system does not create a dimension table. Instead, the SID table of the characteristic takes on the role of dimension table. Removing the dimension table has the following advantages: When loading transaction data, no IDs are generated for the entries in the dimension table. This number range operation can compromise performance precisely in the case where a degenerated dimension is involved. A table having a very large cardinality is removed from the star schema. As a result, the SQL-based queries are simpler. In many cases, the database optimizer can choose better execution plans.

Nevertheless, it also has a disadvantage: A dimension marked as a line item cannot subsequently include additional characteristics. This is only possible with normal dimensions.

**High cardinality:**

This means that the dimension is to have a large number of instances (that is, a high cardinality). This information is used to carry out optimizations on a physical level in depending on the database platform. Different index types are used than is normally the case. A general rule is that a dimension has a high cardinality when the number of dimension entries is at least 20% of the fact table entries. If you are unsure, do not select a dimension having high cardinality.

A simple way to retrieve the necessary information is to leverage the program SAP\_INFOCUBE\_DESIGNS.

A very generic rule is that the dimension table should not take more than 10% of the fact table.

**Navigational Attributes**

Characteristic attributes can be converted into navigational attributes. They can be selected in the query in exactly the same way as the characteristics for an InfoCube. In this case, a new edge/dimension is added to the InfoCube. During the data selection for the query, the data manager connects to the InfoProvider and the master data table ('join') in order to fill the Query.

From a pure performance point of view, you should model an object on a characteristic rather than on a navigational attribute, because: In the enhanced star schema of an InfoCube, navigational attributes lay one join further out than characteristics. This means that a query with a navigational attribute has to run an additional join (compared with a query with the same object as a characteristic) in order to arrive at the values. This is also true for DataStore objects. For the same reason, in some situations, restrictions for particular values in the navigational attribute (values that have been defined in the query) are not taken into account by the database optimizer when it creates run schedules. This can result in inefficient run schedules, particularly if the restrictions are very selective. In most cases, you can solve this problem by indexing the navigational attribute in the corresponding master data tables (see below). If a navigational attribute is used in an aggregate, the aggregate has to be adjusted using a change run as soon as new values are loaded for the navigational attribute (when master data for the characteristic belonging to the navigational attribute is loaded). This change run is usually one of the processes critical to the system performance of a productive BI system. This is why avoiding using navigational attributes, or not using navigational attributes in aggregates, you can improve the performance of this process. On the other hand, not using navigational attributes in aggregates can lead to poor query response times. The data modeler needs to find the right balance.

**Restricted and Calculated Key Figures**

A general recommendation on the need for calculations and formulas being leveraged as custom objects in the data set is to leverage the capabilities of the Restricted Key Figures and Calculated Key Figures as much as possible. The main reasons for this recommendation are:

1. Restricted Key Figures and Calculated Key Figures can be built as part of the underlying BI query, and in that way they will be processed from the BW backend. This allows you to ensure that these elements are not causing any unnecessary overhead for the overall performance.

Xcelsius Against SAP BW (related Xcelsius against SAP BW - SAPBO Specific in previous section). This Section Contains the BW Specific Considerations When Using Xcelsius

In addition to the SAPBO specific Xcelsius best practices earlier in this document, the following best practices should be followed and are particular to the BW layer:

1. To guarantee fast response times, implementers may need to create highly specific BEx queries to base the Universes on, and keep the Universes small. In particular, keep the number of Key Figures low to increase query performance. If necessary, use more Universes rather than less.
2. Push aggregation down to BW. This is a proven and industry standard strategy to keep the data sets small and make queries run fast. Expectations around the response of dashboards are very different from refreshing operational reports.
3. Keep in mind the Xcelsius services (new for XI 3.1 SP3) do not support OLAP Universes but can be leveraged if you access BW via Data Federator.

## SAPBO specific tuning and configuration

### Recommended SBOP and BW Patch Levels

If you're running NW BW 7.01 SP5 & BOE XI 3.1 SP2 FP2.2 you should have the latest fixes and enhancements to maximize BOE performance against BW.

## BW Specific Tuning and Configuration

### Set Specific RSRT Property

Check the "Use Selection of Structure Elements" Query Property in RSRT as seen here:

**Query Monitor**

Execute Execute + Debug Generate Report Properties Messages Help Texts Generation Log  
 Performance Info Technical Information Cache Monitor Query Variants IGS Test Get Variant Create Variant  
 ABAP Web Java Web

Query

Query Display   Parameter 1  Parameter 2

Query Properties -> Move into Query Designer

Read Mode	H Query to Read When You Navigate or Expand	<input checked="" type="checkbox"/> InfoProvider Setting
Req. Status	0 All PartProviders Up to Released Status (RQT)	<input checked="" type="checkbox"/> InfoProvider Setting
Cache Mode	1 Main Memory Cache Without Swapping	<input checked="" type="checkbox"/> InfoProvider Setting
	<input checked="" type="checkbox"/> Update Cache Objects in Delta Process	<input checked="" type="checkbox"/> InfoProvider Setting
	<input checked="" type="checkbox"/> Use Selection of Structure Elements	
	<input type="checkbox"/> Calculate w/ Packed Numbers	
	<input type="checkbox"/> no parallel processing	
	<input type="checkbox"/> Generation log	
Optimization Mode	0 Query Will Be Optimized after Generation	
	<input type="checkbox"/> All Basic Key Figures Input Ready	
Disaggregation on Totals	Default	
Statistic Det.	1 Only Front End/Calculation Layer Data	

Checking this simple property helps improve performance in situations where there are many Restricted Key Figures and other calculations in the Query Definition

If analysis of the query performance indicates very high EVENTID 3200 times, and/or the FEMS number is very high try enabling this property and check performance.

Checking this ensures the structure elements are sent to the database for processing.

## BW via ODA - Why 7.01 SP3?

NW 7.01 SP3 has the following improvements which will benefit SAPBO on BW deployments

1. Avoid unnecessary sorting
  - a. MDX standard imposes results to be sorted
  - b. However, those sorts are typically ignored or not required by WebIntelligence
  - c. Sorting can be avoided by using a new UNORDER() function in SAP's MDX
2. Leaner memory consumption during flattening
  - a. Optimized flattening algorithm
  - b. Non-standard result structure
3. Leaner communication ODA - OLAP BAPI
  - a. Use compressed data exchange via binary XML
4. Availability
  - a. SAP NetWeaver BW 7.01 SP 3
  - b. Business Objects Enterprise XI3.0
  - c. Fix Pack 2 w/ LAFix2.1
  - d. SAP Integration Kit Fix Pack 2 w/LAFix2.1

## Activate OLAP Cache and Delta Cache

A new cache mode is delivered with Support Package 16 for BI 7.0 and currently, the new OLAP cache in B2P is not active yet (SP17 for BI7.0 is implemented in B2P). The Delta cache, which is especially important to uncompressed InfoCubes, is not in use for some queries at this moment. This means that the OLAP has to fetch the data from database again once new requests are sent to InfoCubes, while delta cache is not enabled. This can result in an increase of OLAP time as well as Database time and eventually the entire query runtime. It is recommended to activate the new OLAP Cache mode by running the report SAP\_RSADMIN\_MAINTAIN and setting "Update cache objects in delta process".

## SAP Notes on BW Performance

As a rule, it is preferable to import a relatively recent Support Package rather than individual corrections...

These SAP notes will improve the performance and stability of the MDX (OLAP BAPI) interface in BW. If applicable, apply the advance correction in the relevant SAP Note. Refer to the Composite Note (1142664) for the latest performance improvements for the MDX interface.

1142664	MDX: Composite SAP note about performance improvements
1156101	MDX: Composite SAP note for incorrect data
1157922	System error in CL_RSR_HIERARCHY_DIM and SELECT_SIDS_3
1161911	OLE DB for OLAP: Many read accesses to table RSRREPDIR
1162349	OLE DB for OLAP: Restriction on text and performance
1164552	Performance improvement if zero suppression is active
1169205	MDX: Filter transfer in the case of cross join
1170323	MDX: Performance when filtering on hierarchy node
1172076	Performance improvement in special situation
1230303	MDX: Buffering of hierarchy metadata
1236241	MDX: System error in CL_RSR_HIERARCHY_DIM SELECT_SIDS_3
1237104	Performance problems occur when reading the master data
1238661	Slight performance improvement in internal business volume
1240165	Performance optimization for hierarchy authorizations
1252372	OLE DB for OLAP: Additional corrections for new MDX parser
1257723	OLE DB for OLAP: Short length specified for MEMBER_CAPTION
1265072	MDX: No data in a very specific situation

## Related Content

You will find the following material useful as an adjunct to this white paper or in providing additional information.

[http://help.sap.com/saphelp\\_nw70/helpdata/EN/43/e39fd25ff502d2e10000000a1553f7/frameset.htm](http://help.sap.com/saphelp_nw70/helpdata/EN/43/e39fd25ff502d2e10000000a1553f7/frameset.htm)

<https://www.sdn.sap.com/irj/sdn/bi-performance-tuning>

<http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/20232>

<http://www.sdn.sap.com/irj/boc/bi-solution-architecture>

## Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.