

# Improve your Reporting Performance in SAP BusinessObjects Planning and Consolidation Version for NetWeaver



## Applies to:

SAP BusinessObjects (SBOP) Planning and Consolidation version for NetWeaver 7.x, based on a NetWeaver BW 7.01 platform. For more information, visit the [Enterprise Performance Management homepage](#).

## Summary

This white paper covers the main aspects to be considered when creating any EVDRE reports or input templates for SBOP Planning and Consolidation version for NetWeaver (SBOP PC\_NW or BPC) to achieve optimal performance. Areas addressed include: EVDRE definition, RSDRI queries, and Tips and Tricks in the specific context of SBOP PC\_NW.

**Author:** Rob Marshall

**Company:** SAP

**Created on:** 15 June 2011

## Author Bio



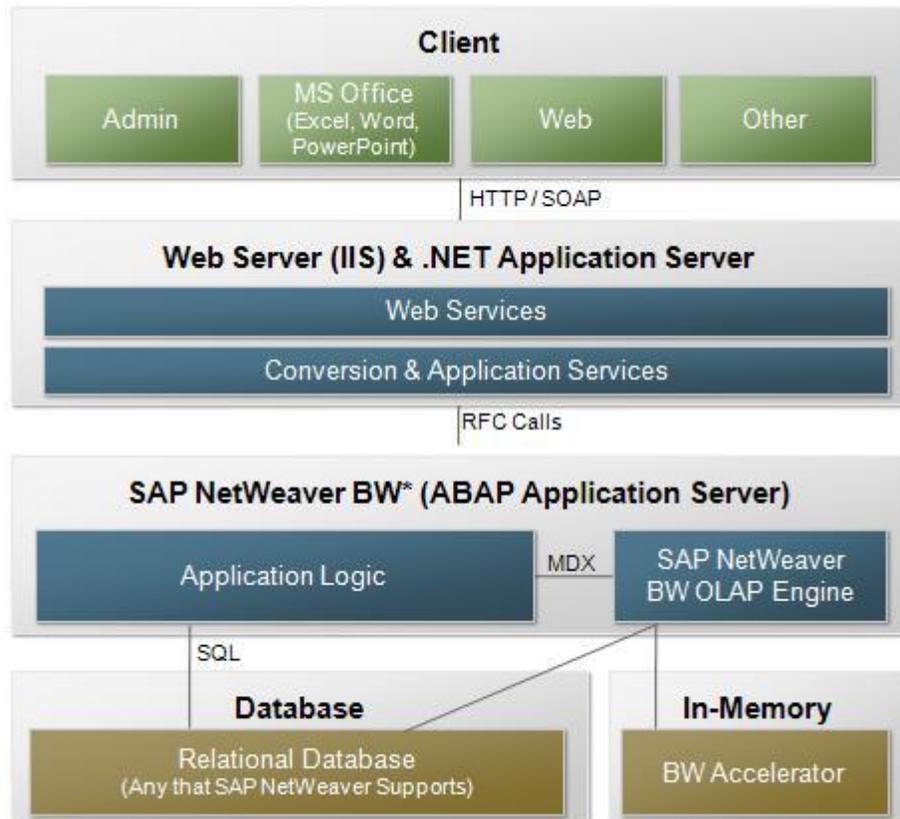
Rob Marshall is a certified BPC consultant working in the SAP CSA (formerly RIG) group. Rob has been working with BPC for over seven years starting with OutlookSoft 4.0.

---

## Table of Contents

SBOP PC_NW Architecture.....	3
Anatomy of an EVDRE .....	3
Types of EVDRE Queries .....	6
Sparsity Coefficient.....	6
UJSTAT .....	6
RSDRI Events .....	7
MDX Events .....	8
Tips for better Performance.....	9
Create RSDRI queries .....	9
Limit MDX queries.....	9
Create Multiple EVDRE's .....	9
Expansions .....	9
Orphaned Data .....	10
Custom Suppression.....	11
EVDRE's by ACCTYPE .....	13
Summation of the fly .....	13
Summary.....	16
Related Content.....	17
Copyright.....	18

## SBOP PC\_NW Architecture



SAP BusinessObjects Planning and Consolidation, version for NetWeaver is a four layer application composed of a client stack, a .NET Application server, the NetWeaver ABAP Application back-end, and the database server.

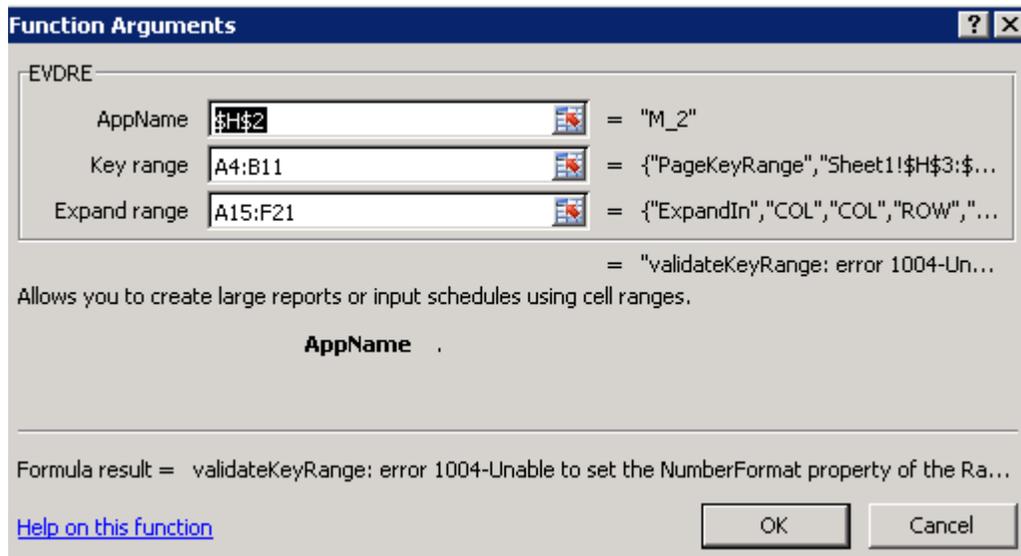
This paper focuses on the report performance between the client tier and the database/OLAP engine and how to maximize the performance in getting returned.

When executing EVDRE reports from the users' computer, the request goes through the .Net Layer to the Shared Query Engine (SQE) on the ABAP application server. The SQE then instructs the query to either retrieve the data from the cube (MDX) or directly from the fact tables (RSDRI).

### Anatomy of an EVDRE

EVDRE stands for Everest Dynamic Range Expansion and it is the recommended way to query the BPC database from the MS Excel frontend of BPC.

Within all EVDRE's, there are three main parts needed to be defined as they control the behavior and formatting of the EVDRE report: the Application, Key ranges, and the Expand ranges (what dimensions members the report is rendering).



- Application – used to denote which cube or application the data is currently stored in.
- Key Ranges – used to denote where the columns and rows begin and end.
- Expand Range – denotes the dimensions and dimension members that will be rendered

#### Key Ranges:

RANGE	VALUE
PageKeyRange	Sheet1!\$H\$3:\$H\$14
ColKeyRange	Sheet1!\$HB\$37:\$HV\$38,Sheet1!\$HX\$37:\$HY\$38
RowKeyRange	Sheet1!\$H\$50:\$J\$6321
CellKeyRange	
GetOnlyRange	
FormatRange	
OptionRange	Sheet1!\$A\$25:\$B\$41
SortRange	

- PageKeyRange: This directs the EVDRE to the current view or what intersections the report should return data for.
- ColKeyRange: Defines the range of column cells that the report should render data. As you can see above, you can have multiple expansions defined which are separated by a comma. In the above example, columns will be defined from cells HB37 to HV38 as well as HX37 to HY38. You might have noticed the column ranges HB37:HV37 and HX37:HY38 are hidden, this will be explained in the Trips and Tricks section.
- RowKeyRange: Defines the range of row cells that the report should render data. As with column expansion, you can also define multiple row expansions.
- OptionRange: Defines the options of the report. The most common options are listed in the picture below:

Option	Value
AutoFitCol	Y
Bottom	
DumpDataCache	
ExpandOnly	
HideColKeys	Y
HideRowKeys	Y
NoRefresh	
NoSend	
ShowComments	
ShowNullAsZero	
SortCol	
SumParent	
SuppressDataCol	
SuppressDataRow	
SuppressNoData	
Top	
QueryByMemberSet	Y

### Expand Ranges:

PARAMETER	EXPANSION 1	EXPANSION 2	EXPANSION 3	EXPANSION 4	EXPANSION 5
ExpandIn	COL	COL	ROW	ROW	ROW
Dimension	ACCOUNT	RPTCURRENCY	SUPPLIER	CUSTOMER	PRODUCT
MemberSet	BUS_STATUS,TECH_CODE,TECH	LC USD	BAS	BAS	BAS
BeforeRange					
AfterRange					
Suppress					(PART_VOLUME,BAS(200
Insert					

- ExpandIn – defines where to expand, either the columns (COL) or row (ROW)
- Dimension – defines which dimension is being expanded.
- MemberSet – defines the members to be expanded. This can be an MDX command like SELF, BAS, MEMBERS, DEP (dependants), or All. Also, as evidenced in the above first expansion, you can list members, separated by commas if you know exactly which members you want data for. Additionally, in the second expansion, you can utilize pipes “|” which denotes a second expansion. Using pipes within a single EVDRE will often, but not always, have better query performance than using multiple EVDRE’s.
- BeforeRange – used to insert formulas in the column or row before the expansion begins.
- AfterRange - used to insert formulas in the column or row after the expansion ends. For example, if you would like a summation of the columns, you can enter a formula in a cell that is not within the expansion cells (ColKeyRange), reference those cells with EVRNG(the cells containing the formula), and the report will create an extra column.
- Suppress – If you would like to suppress the data (bring back only rows/columns having data), you can enter a “Y” or a particular intersection of data (Sales,2009.JAN).
- Insert (Optional) – Setting the parameter to Y for any row or column will allow the user to perform a runtime insertion of suppressed row or column values.

## Types of EVDRE Queries

To understand the theory of an EVDRE, you must first understand the different types of queries that can be produced from an EVDRE. There are two types of queries executed from the SQE:

- RSDRI – is the function group in SAP NetWeaver BW that deals with access to InfoProviders. In the SQE, an RSDRI query calls the internal SAP NetWeaver BW functions to read data from an InfoProvider and accesses the E and F fact tables. This is the preferred query mode as RSDRI queries are faster than MDX queries. However, RSDRI queries do not bring in aggregated data, just base level information.
- MDX – SAP NetWeaver BW exposes OLAP BAPIs to access data from an InfoProvider through MDX. The MDX query method in SAP BPC generates the MDX statements required to retrieve data with the SAP BPC application logic applied. It is always used when there are dimension member formulas, measure formulas, or hierarchy parent values (2009.TOTAL) to be retrieved and are often slower than RSDRI queries.

Prior to release 7, service pack 3, there was an additional type of query, a cell query. This was associated with only MDX type of queries and was often very slow as it would retrieve individual cells. Since this the performance was so slow, all queries are now Axis queries. Axis queries request a contiguous region of data. Due to this, often times the query fetches more data than is needed, however this is still faster than individual cell retrievals.

## Sparsity Coefficient

Prior to version 7 service pack 3, an administrator could customize when a cell query was used by manipulating the Sparsity Coefficient. Now that cell queries are no longer available, the Sparsity Coefficient is no longer used.

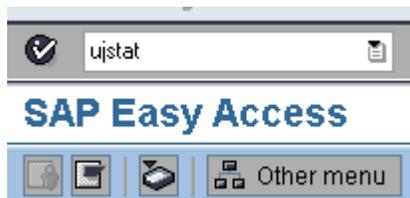
## UJSTAT

Within the SAP GUI, there is a transaction (UJSTAT) allowing a user to understand what type of query (RSDRI or MDX) is being used as well as how long a query takes to run. In order for UJSTAT to work, you must turn on the BPC\_STATISITCS in the web administration portal, specifically in the Set Application Parameters.

### Administration >> Set Application Parameters

Type	KeyID	Value	Delete
(R)	APPROVALORG		<input type="checkbox"/>
	BPC_STATISTICS	ON	<input type="checkbox"/>

To get to this transaction, login to the SAP GUI and type in UJSTAT



On the next screen, enter the appropriate Start Date, Start Time, Appset, and Application. If multiple people are running reports at the same time, it would be wise to also enter the User ID to segment out the queries to just the ones you run.

Selections	
Statistics Session	
User ID	
Statistics Start Date	10.01.2011 to 10.01.2011
Statistics Start Time	21:30:00 to 23:59:00
Action ID	
AppSet	ELITE_TRAINING_EVD
Application	M_5

After an EVDRE has successfully completed in SBOP Planning and Consolidation, go to transaction UJSTAT to see the results.

Hierarchy	AppSet ID	Appl. ID	Seq.	Start Date	Start Time	Runtime	T Runtime	C Runtime	Nr of Recs
Shared Query Engine Read	ELITE_TRAININ... M_5		0	10.01.2011	21:45:30	0.109000	0.109000	0.109000	0
Shared Query Engine Read	ELITE_TRAININ... M_5		0	10.01.2011	21:45:04	7.343000	7.343000	7.343000	0
Unwrap XML	ELITE_TRAININ... M_5		1	10.01.2011	21:44:56	0.000000	0.000000	0.000000	0
Run Axis Query	ELITE_TRAININ... M_5		2	10.01.2011	21:44:56	0.000000	7.328000	7.328000	1,450
Check Validation for Axis Query	ELITE_TRAININ... M_5		3	10.01.2011	21:44:56	0.109000	0.109000	0.109000	0
Check Member Security (Axis)	ELITE_TRAININ... M_5		4	10.01.2011	21:44:56	0.000000	0.000000	0.109000	0
move_2 slicer	ELITE_TRAININ... M_5		5	10.01.2011	21:44:56	0.000000	0.000000	0.109000	0
Get Dimension and Member In	ELITE_TRAININ... M_5		6	10.01.2011	21:44:56	4.344000	4.344000	4.453000	0
Build Sub Queries for Multiple	ELITE_TRAININ... M_5		7	10.01.2011	21:45:01	0.000000	0.000000	4.453000	0
Miscellaneous Processing Tirr	ELITE_TRAININ... M_5		8	10.01.2011	21:45:01	0.000000	0.000000	4.453000	0
Run Sub Query for Measures	ELITE_TRAININ... M_5		9	10.01.2011	21:45:01	0.000000	2.890000	7.343000	0
Miscellaneous Processing	ELITE_TRAININ... M_5		10	10.01.2011	21:45:01	0.000000	0.000000	4.453000	0
Convert_rsdri_input	ELITE_TRAININ... M_5		11	10.01.2011	21:45:01	0.015000	0.015000	4.468000	0
Run RSDRI Query	ELITE_TRAININ... M_5		12	10.01.2011	21:45:01	0.000000	2.875000	7.343000	0
RSDRI Query Kernel	ELITE_TRAININ... M_5		13	10.01.2011	21:45:01	0.000000	2.844000	7.312000	1,450
Miscellaneous Proc	ELITE_TRAININ... M_5		14	10.01.2011	21:45:01	0.000000	0.000000	4.468000	0
RSDRI_INFOPROV	ELITE_TRAININ... M_5		15	10.01.2011	21:45:01	2.844000	2.844000	7.312000	1,450
Rsdri_aggregate_d	ELITE_TRAININ... M_5		16	10.01.2011	21:45:04	0.000000	0.000000	7.312000	1,450
Convert RSDRI Result	ELITE_TRAININ... M_5		17	10.01.2011	21:45:04	0.016000	0.016000	7.328000	0
Process SQE Badl	ELITE_TRAININ... M_5		18	10.01.2011	21:45:04	0.000000	0.000000	7.328000	1,450

The screenshot informs the user of several key factors for this query:

- This query is a RSDRI query
- The Runtime, Technical Runtime, and Cumulative Runtime
- The number of records returned (Nr of Recs)

## RSDRI Events

When a report or input template contains multiple EVDRE's, it is important to know which one you are debugging. To do this, double click on the Unwrap XML event and look for "Filter."

```
Support Details
</Dim>
<Dim>
<Name><![CDATA[MEASURES]]></Name>
<Value><![CDATA[PERIODIC]]></Value>
</Dim>
<Dim>
<Name><![CDATA[PRODUCT]]></Name>
<Value><![CDATA[OTHER_SEATBELTS]]></Value>
</Dim>
</CurrentView>
<SAP_PASSPORT></SAP_PASSPORT><Machine_Name>
PHLN00466506A</Machine_Name>
</Context>
Filter:<parameter><CV application="M_5" DATASRC="TOTALADJ"
ISSUANCE="ISSUE_3_0_0914" MEASURES="PERIODIC"
RPTCURRENCY="LC" VEHICLE="TOT_VEHICLES"
VEHICLE_REGION="ALL_REGIONS"/><Axes><column
dimension="ACCOUNT"
members="APPLICATION_RATE|ALLOCATION_RATE"/><column
dimension="TIME" members="2010.JAN|2011.JAN|2012.JAN|
2013.JAN|2014.JAN"/><row dimension="SUPPLIER"
members="CHIHUAHUA_DAB_SW|CHIHUAHUA_SW"/><row
dimension="CUSTOMER" members="C_OEM"/><row
dimension="PRODUCT" members="398333|432998|433010|
433021|482227|482596|483605|487714|488870|493834|556158|
```

This will tell you what the current view is as well as what data will be returned for the column and row axis.

By utilizing UJSTAT, we learn the below RSDRI query took 7.34 seconds to run and of that, it took 4.34 seconds to download the dimension cache and 2.844 seconds to return 1,450 records from the infoprovider.

Task	AppSet ID	Appl. ID	Seq.	Start Date	Start Time	Runtime	T Runtime	C Runtime	Nr of Recs
Shared Query Engine Read	ELITE_TRAININ... M_5		0	10.01.2011	21:45:04	7.343000	7.343000	7.343000	0
Unwrap XML	ELITE_TRAININ... M_5		1	10.01.2011	21:44:56	0.000000	0.000000	0.000000	0
Run Axis Query	ELITE_TRAININ... M_5		2	10.01.2011	21:44:56	0.000000	7.328000	7.328000	1,450
Check Validation for Axis Query	ELITE_TRAININ... M_5		3	10.01.2011	21:44:56	0.109000	0.109000	0.109000	0
Check Member Security (Axis)	ELITE_TRAININ... M_5		4	10.01.2011	21:44:56	0.000000	0.000000	0.109000	0
move_2_slicer	ELITE_TRAININ... M_5		5	10.01.2011	21:44:56	0.000000	0.000000	0.109000	0
Get Dimension and Member Information	ELITE_TRAININ... M_5		6	10.01.2011	21:44:56	4.344000	4.344000	4.453000	0
Build Sub Queries for Multiple Hierarchies	ELITE_TRAININ... M_5		7	10.01.2011	21:45:01	0.000000	0.000000	4.453000	0
Miscellaneous Processing Time	ELITE_TRAININ... M_5		8	10.01.2011	21:45:01	0.000000	0.000000	4.453000	0
Run Sub Query for Measures	ELITE_TRAININ... M_5		9	10.01.2011	21:45:01	0.000000	2.890000	7.343000	0
Miscellaneous Processing Time	ELITE_TRAININ... M_5		10	10.01.2011	21:45:01	0.000000	0.000000	4.453000	0
Convert_rsdri_input	ELITE_TRAININ... M_5		11	10.01.2011	21:45:01	0.015000	0.015000	4.468000	0
Run RSDRI Query	ELITE_TRAININ... M_5		12	10.01.2011	21:45:01	0.000000	2.875000	7.343000	0
RSDRI Query Kernel	ELITE_TRAININ... M_5		13	10.01.2011	21:45:01	0.000000	2.844000	7.312000	1,450
Miscellaneous Processing Time	ELITE_TRAININ... M_5		14	10.01.2011	21:45:01	0.000000	0.000000	4.468000	0
RSDRI_INFOPROV_READ	ELITE_TRAININ... M_5		15	10.01.2011	21:45:01	2.844000	2.844000	7.312000	1,450
Rsdri_aggregate_data	ELITE_TRAININ... M_5		16	10.01.2011	21:45:04	0.000000	0.000000	7.312000	1,450
Convert RSDRI Result	ELITE_TRAININ... M_5		17	10.01.2011	21:45:04	0.016000	0.016000	7.328000	0
Process SQE BAdI	ELITE_TRAININ... M_5		18	10.01.2011	21:45:04	0.000000	0.000000	7.328000	1,450

### MDX Events

An EVDRE that creates an MDX statement is shown below, designated by the event MDX Query Kernel.

Hierarchy	AppSet ID	Appl. ID	Seq.	Start Date	Start Time	Runtime	T Runtime
Shared Query Engine Read	ELITE_TRAININ... M_5		0	10.01.2011	21:44:46	9.187000	9.187000
Unwrap XML	ELITE_TRAININ... M_5		1	10.01.2011	21:44:37	0.000000	0.000000
Run Axis Query	ELITE_TRAININ... M_5		2	10.01.2011	21:44:37	0.000000	9.141000
Check Validation for Axis Query	ELITE_TRAININ... M_5		3	10.01.2011	21:44:37	0.109000	0.109000
Check Member Security (Axis)	ELITE_TRAININ... M_5		4	10.01.2011	21:44:37	0.000000	0.000000
move_2_slicer	ELITE_TRAININ... M_5		5	10.01.2011	21:44:37	0.000000	0.000000
Get Dimension and Member Information	ELITE_TRAININ... M_5		6	10.01.2011	21:44:37	4.344000	4.344000
Build Sub Queries for Multiple Hierarchies	ELITE_TRAININ... M_5		7	10.01.2011	21:44:42	0.000000	0.000000
Miscellaneous Processing Time	ELITE_TRAININ... M_5		8	10.01.2011	21:44:42	0.000000	0.000000
Run Sub Query for Measures	ELITE_TRAININ... M_5		9	10.01.2011	21:44:42	0.000000	4.734000
Miscellaneous Processing Time	ELITE_TRAININ... M_5		10	10.01.2011	21:44:42	0.000000	0.000000
Process Slicer	ELITE_TRAININ... M_5		11	10.01.2011	21:44:42	0.000000	0.000000
Run Sub Query for Multiple Hierarchies	ELITE_TRAININ... M_5		12	10.01.2011	21:44:42	0.000000	4.734000
Process Axis	ELITE_TRAININ... M_5		13	10.01.2011	21:44:42	0.016000	0.016000
Process Formula	ELITE_TRAININ... M_5		14	10.01.2011	21:44:42	0.000000	0.000000
Axis Query Kernel	ELITE_TRAININ... M_5		15	10.01.2011	21:44:42	0.000000	4.718000
Miscellaneous Processing Time	ELITE_TRAININ... M_5		16	10.01.2011	21:44:42	0.000000	0.000000
MDX Query Kernel	ELITE_TRAININ... M_5		17	10.01.2011	21:44:42	0.000000	4.718000

Within the MDX Query Kernel, the event Mddataset\_create\_object is useful in that it shows the actual MDX created.

Hierarchy	AppSet ID	Appl. ID	Seq.	Start Date	Start Time	Runtime	T Runtime	C Runtime	Nr of Recs
Process Formula	ELITE_TRAININ... M_5		14	10.01.2011	21:44:42	0.000000	0.000000	4.469000	0
Axis Query Kernel	ELITE_TRAININ... M_5		15	10.01.2011	21:44:42	0.000000	4.718000	9.187000	4,541
Miscellaneous Processing Time	ELITE_TRAININ... M_5		16	10.01.2011	21:44:42	0.000000	0.000000	4.469000	0
MDX Query Kernel	ELITE_TRAININ... M_5		17	10.01.2011	21:44:42	0.000000	4.718000	9.187000	4,541
Miscellaneous Processing Time	ELITE_TRAININ... M_5		18	10.01.2011	21:44:42	0.000000	0.000000	4.469000	0
Mddataset_create_object	ELITE_TRAININ... M_5		19	10.01.2011	21:44:42	0.000000	0.000000	4.469000	0
Mddataset_select_data	ELITE_TRAININ... M_5		20	10.01.2011	21:44:42	2.156000	2.156000	6.625000	0
Get_mdx_data	ELITE_TRAININ... M_5		21	10.01.2011	21:44:44	0.000000	2.516000	9.141000	4,541

Performance Statistics Report		Support Details
Field Description	Cell Contents	
Statistics Session	D9NDUD97YE0Z6B6M0VXD7B9C2	<pre> WITH MEMBER [Measures].[PERIODIC] AS 'IF({/CPMB/BDWI79 PARENTH1].[2/CPMB/ACCTYPE]="INC",-[Measures]. {/CPMB/SDATA},IF({/CPMB/BDWI79 PARENTH1].[2/CPMB/ACCTYPE]="EXP",[Measures].{/CPMB/SDATA},IF {/CPMB/BDWI79 PARENTH1].[2/CPMB/ACCTYPE]="AST", ([Measures].{/CPMB/SDATA}, CLOSINGPERIOD({/CPMB/BD1H1Z PARENTH1].[LEVEL02})),IF({/CPMB/BDWI79 PARENTH1].[2/CPMB/ACCTYPE]="LEQ",-([Measures].{/CPMB/SDATA}, CLOSINGPERIOD({/CPMB/BD1H1Z PARENTH1].[LEVEL02}))-[Measures].{/CPMB/SDATA}}))";SOLVE_ORDER=3 SELECT NON EMPTY({/CPMB/BD1H1Z PARENTH1].[2009.JAN], {/CPMB/BD1H1Z PARENTH1].[2009.FEB], {/CPMB/BD1H1Z PARENTH1].[2009.MAR], {/CPMB/BD1H1Z PARENTH1].[2009.APR], {/CPMB/BD1H1Z PARENTH1].[2009.MAY], {/CPMB/BD1H1Z PARENTH1].[2009.JUN], </pre>
Sequence Number	19	
Event ID	000002031	
Parent Event Id	000002019	
Stats Start Date	20110110	
Stats Start Time	214442	
Stats Start Time (D)	20110110204442.0370000	
Runtime (seconds)	0.000000	
Number of Records	0	
NWBI Statistics UUID		
Actual Caller	D9NDUD97YETLA155Y4J1LVMYQ	

By using this event, the user can understand the MDX being created.

## Tips for better Performance

By keeping the following tips in mind when creating a report or input template, the performance will be maximized.

### Create RSDRI queries

RSDRI queries are the fastest way to get data out of the database, as long as the database is properly tuned and indexed. The easiest way to do this is to bring back base level members or non aggregated data for the column and row expansions. If 2010.TOTAL is specified in the current view and the expansion is SELF, the SQE will create a MDX call to the database. On the other hand, if BAS is specified in the expansion, the SQE will create a RSDRI query.

### Limit MDX queries

While MDX queries don't always perform as well as RSDRI queries, they are not bad if you use them correctly. To do this, limit the amount of columns needed in the MDX statement. Typically, this will create multiple EVDRE's where you can segregate the base members from the calculated members, thus easing the burden in the number of columns returned or need for the MDX query.

### Create Multiple EVDRE's

It is best practice to divide up one large query into multiple EVDRE's. Not only will this help the in the performance of the report, but it will also help the application server deliver the information in a more timely manner since it will not have to store the information. Additionally, the developer will be able to segment RSDRI queries and MDX, so performance will be better.

### Expansions

When using multiple EVDRE's within a report or input template, make sure the report only expands on what it really needs. For instance, if the first EVDRE expands time along the columns and Products along the rows, the second EVDRE doesn't need to make another database call for the rows.

The first expansion is defined as ...

PARAMETER	EXPANSION 1	EXPANSION 2
ExpandIn	COL	ROW
Dimension	TIME	PRODUCT
MemberSet	BAS	BAS
BeforeRange		
AfterRange		
Suppress		Y
Insert		

And the second EVDRE expansion is defined without the Row expansion.

PARAMETER	EXPANSION 1
ExpandIn	COL
Dimension	TIME
MemberSet	BAS(2011.TOTAL)
BeforeRange	
AfterRange	
Suppress	
Insert	

For this to work, the second and subsequent EVDRE's RowKeyRange must be the same as the initial EVDRE.

### Orphaned Data

When having multiple EVDRE's, it is possible to have orphaned data when a smaller dataset is returned after a larger dataset is returned. For instance, if the report returns 1,000 rows, and the second run only brings back 750 rows, you will still have the initial 250 rows "left over."

210	792689	792689	140.83	161.26	302.09	241.52	281.65	180.96	321.79	321.79	302.09	281.65	281.65	223.64
211	792696	792696	487.80	559.51	1,047.30	836.90	977.96	628.86	1,116.65	1,116.65	1,047.30	977.96	977.96	776.50
212	792697	792697	119.78	137.12	256.11	204.89	239.56	153.67	273.45	273.45	256.11	239.56	239.56	190.18
213	445273	445273											250.99	
214														25,201.50
215														1,299.22
216														

To guard against this, encapsulate the second or more EVDRE's within the scope of the initial EVDRE. To demonstrate this, create an EVDRE and change the ColKeyRange to the below:

A	B
EVDRE:OK	
RANGE	VALUE
PageKeyRange	Sheet1!\$E\$3:\$E\$13
ColKeyRange	Sheet1!\$A\$45:\$A\$45
RowKeyRange	Sheet1!\$H\$47:\$H\$213
CellKeyRange	
GetOnlyRange	
FormatRange	
OptionRange	Sheet1!\$A\$25:\$B\$41
SortRange	

On the second EVDRE, change the ColKeyRange to =EVRNG(W45:AH45)

G	H
EVDRE:OK	
RANGE	VALUE
PageKeyRange	Sheet1!\$K\$3:\$K\$13
ColKeyRange	Sheet1!\$W\$45:\$AH\$45
RowKeyRange	Sheet1!\$H\$47:\$H\$213
CellKeyRange	
GetOnlyRange	
FormatRange	
OptionRange	Sheet1!\$A\$25:\$B\$41
SortRange	

Now, in the report area where data is returned (where the initial EVDRE was supposed to be – cells J45:U45), use cell referencing to the data area of the first EVDRE.



Hierarchy	AppSet ID	Appl. ID	Seq N	Start Date	Start Time	Runtime
Shared Query Engine Read	ELITE_TRAININ... M_5		0	11.01.2011	00:36:39	7.375000
Unwrap XML	ELITE_TRAININ... M_5		1	11.01.2011	00:36:32	0.000000
Run Axis Query	ELITE_TRAININ... M_5		2	11.01.2011	00:36:32	0.000000
Check Validation for Axis Query	ELITE_TRAININ... M_5		3	11.01.2011	00:36:32	0.828000
Check Member Security (Axis)	ELITE_TRAININ... M_5		4	11.01.2011	00:36:33	0.000000
move_2_slicer	ELITE_TRAININ... M_5		5	11.01.2011	00:36:33	0.000000
Get Dimension and Member In	ELITE_TRAININ... M_5		6	11.01.2011	00:36:33	4.469000
Build Sub Queries for Multiple	ELITE_TRAININ... M_5		7	11.01.2011	00:36:37	0.000000
Miscellaneous Processing Tirr	ELITE_TRAININ... M_5		8	11.01.2011	00:36:37	0.062000
Run Sub Query for Measures	ELITE_TRAININ... M_5		9	11.01.2011	00:36:37	0.000000
Miscellaneous Processing	ELITE_TRAININ... M_5		10	11.01.2011	00:36:37	0.000000
Convert_rsdri_input	ELITE_TRAININ... M_5		11	11.01.2011	00:36:37	0.000000
Run RSDRI Query	ELITE_TRAININ... M_5		12	11.01.2011	00:36:37	0.000000
Wrap xml after QUERY	ELITE_TRAININ... M_5		19	11.01.2011	00:36:39	0.000000

If we change the suppression from Y to SALES,2009.TOTAL, UJSTAT will show two events have been created and have taken longer than the original query.

Hierarchy	AppSet ID	Appl. ID	Seq N	Start Date	Start Time	Runtime	T Runtime	C Runtime
Shared Query Engine Read	ELITE_TRAININ... M_5		0	11.01.2011	00:41:17	11.031000	11.031000	11.031000
Shared Query Engine Read	ELITE_TRAININ... M_5		0	11.01.2011	00:41:06	8.906000	8.906000	8.906000

With that said, custom suppression can be used, if used correctly. The main thing to remember is to create an RSDRI query for the suppression and stay away from MDX. For example, if the current view contains Sales as the account but we wanted to suppress all data where PART\_VOLUME for all of 2009, we could enter (PART\_VOLUME, 2009.TOTAL). To take it one step further, if we wanted to suppress rows for multiple years, we could enter (PART\_VOLUME, [2009.TOTAL,2010.TOTAL]), but this will turn into MDX query. The following example amplifies this:

We created an ad hoc report with the following with expansion ranges.

PARAMETER	EXPANSION 1	EXPANSION 2	EXPANSION 3	EXPANSION 4
ExpandIn	COL	ROW	ROW	ROW
Dimension	TIME	SUPPLIER	CUSTOMER	PRODUCT
MemberSet	BAS	BAS	BAS	BAS
BeforeRange				
AfterRange				
Suppress				PART_VOLUME,[2009.TOTAL,2010.TOTAL]
Insert				

By looking in transaction UJSTAT, we will see the run time being excessively long due to the axis query creating an MDX statement to return the data.

Shared Query Engine Read	SAP_M2 M_2	0	10/20/2010	00:37:42	2,120.087613	2,120.087613	2,120.087613	0
Unwrap XML	SAP_M2 M_2	1	10/20/2010	00:02:22	0.021891	0.021891	0.021891	0
Run Axis Query	SAP_M2 M_2	2	10/20/2010	00:02:22	0.000250	2,120.062640	2,120.084531	409
Check Validation for Axis Query	SAP_M2 M_2	3	10/20/2010	00:02:22	0.279750	0.279750	0.301891	0
Check Member Security (Axis)	SAP_M2 M_2	4	10/20/2010	00:02:22	0.000434	0.000434	0.302325	0
move_2_slicer	SAP_M2 M_2	5	10/20/2010	00:02:22	0.000021	0.000021	0.302346	0
Get Dimension and Member In	SAP_M2 M_2	6	10/20/2010	00:02:22	12.756739	12.756739	13.059085	0
Build Sub Queries for Multiple	SAP_M2 M_2	7	10/20/2010	00:02:35	0.012920	0.012920	13.072005	0
Miscellaneous Processing Tirr	SAP_M2 M_2	8	10/20/2010	00:02:35	0.007780	0.007780	13.079785	0
Run Sub Query for Measures	SAP_M2 M_2	9	10/20/2010	00:02:35	0.000239	2,107.007828	2,120.087613	409
Miscellaneous Processing	SAP_M2 M_2	10	10/20/2010	00:02:35	0.010295	0.010295	13.090319	0
Process Slicer	SAP_M2 M_2	11	10/20/2010	00:02:35	0.000191	0.000191	13.090510	0
Run Sub Query for Multiple	SAP_M2 M_2	12	10/20/2010	00:02:35	0.000225	2,106.997103	2,120.087613	409
Process Axis	SAP_M2 M_2	13	10/20/2010	00:02:35	0.075256	0.075256	13.165991	0
Process Formulas	SAP_M2 M_2	14	10/20/2010	00:02:35	0.011838	0.011838	13.177829	0
Axis Query Kernel	SAP_M2 M_2	15	10/20/2010	00:02:35	0.000195	2,106.909784	2,120.087613	409
Miscellaneous Proc	SAP_M2 M_2	16	10/20/2010	00:02:35	0.005865	0.005865	13.183889	0
MDX Query Kernel	SAP_M2 M_2	17	10/20/2010	00:02:35	0.000233	2,106.903724	2,120.087613	409

By changing the suppression to PART\_VOLUME,[BAS(2009.TOTAL,2010.TOTAL)], the statistics in UJSTAT are much better since an RSDRI query is produced.

PARAMETER	EXPANSION 1	EXPANSION 2	EXPANSION 3	EXPANSION 4
ExpandIn	COL	ROW	ROW	ROW
Dimension	TIME	SUPPLIER	CUSTOMER	PRODUCT
MemberSet	BAS	BAS	BAS	BAS
BeforeRange				
AfterRange				
Suppress				PART_VOLUME,[BAS(2009.TOTAL,2010.TOTAL)]
Insert				

Shared Query Engine Read	SAP_M2_M_2	0	10/20/2010	01:04:01	584.142850	584.142850	584.142850	0
Unwrap XML	SAP_M2_M_2	1	10/20/2010	00:54:17	0.022235	0.022235	0.022235	0
Run Axis Query	SAP_M2_M_2	2	10/20/2010	00:54:17	0.000252	584.106912	584.129147	1,856
Check Validation for Axis Query	SAP_M2_M_2	3	10/20/2010	00:54:17	0.367243	0.367243	0.389730	0
Check Member Security (Axis)	SAP_M2_M_2	4	10/20/2010	00:54:17	0.000444	0.000444	0.390174	0
move_2_slicer	SAP_M2_M_2	5	10/20/2010	00:54:17	0.000021	0.000021	0.390195	0
Get Dimension and Member In	SAP_M2_M_2	6	10/20/2010	00:54:17	12.815215	12.815215	13.205410	0
Build Sub Queries for Multiple	SAP_M2_M_2	7	10/20/2010	00:54:30	0.012833	0.012833	13.218243	0
Miscellaneous Processing Trr	SAP_M2_M_2	8	10/20/2010	00:54:30	0.097598	0.097598	13.315841	0
Run Sub Query for Measures	SAP_M2_M_2	9	10/20/2010	00:54:30	0.000214	570.827009	584.142850	0
Miscellaneous Processing	SAP_M2_M_2	10	10/20/2010	00:54:30	0.010278	0.010278	13.326333	0
Convert_rsdri_input	SAP_M2_M_2	11	10/20/2010	00:54:30	0.007053	0.007053	13.333386	0
Run RSDRI Query	SAP_M2_M_2	12	10/20/2010	00:54:30	0.076684	570.809464	584.142850	0

Also, notice the far right column in the two pictures of UJSTAT, the MDX query returned 409 rows of data while the RSDRI query returned 1,856. Even though the RSDRI query returned more rows, it was almost 400% faster.

### EVDRE's by ACCTYPE

When an EVDRE is rendering more than one ACCTYPE (INC, EXP, AST, LEQ), it is best to create multiple EVDRE's rather than using a pipe ("|") or comma (",") due to the MDX performed by the Measures dimension. For example, we created one EVDRE rendering one EXP account and one AST account on the columns, as depicted below.

PARAMETER	EXPANSION 1	EXPANSION 2	EXPANSION 3	EXPANSION 4
ExpandIn	COL	ROW	ROW	ROW
Dimension	ACCOUNT	SUPPLIER	CUSTOMER	PRODUCT
MemberSet	NET_PRICE,PART_VOLUME	BAS	BAS	BAS
BeforeRange				
AfterRange				
Suppress				PART_VOLUME,[2009.JAN,2009.FEB]
Insert				

By viewing UJSTAT, the report renders in just over 27 seconds.

Shared Query Engine Read	SAP_M2	M_2	0	10/20/2010	03:35:17	1.819052	1.819052	1.819052
Shared Query Engine Read	SAP_M2	M_2	0	10/20/2010	03:35:14	25.253592	5.253592	25.253592

But, if we create multiple EVDRE's, by account, the report actually returns faster, even though the exact same data is being returned.

Hierarchy	AppSetID	Appl. ID	Seq N	Start Date	Start Time	Runtime	T Runtime	C Runtime
Shared Query Engine Read	SAP_M2	M_2	0	10/20/2010	03:47:47	1.806921	1.806921	1.806921
Shared Query Engine Read	SAP_M2	M_2	0	10/20/2010	03:47:45	1.783619	1.783619	1.783619
Shared Query Engine Read	SAP_M2	M_2	0	10/20/2010	03:47:42	17.944541	17.944541	17.944541

### Summation of the fly

It is not widely known, but you can do summations in the Key Range of an EVDRE. This helps in creating more RSDRI queries than MDX due to the fact you are only bringing back base members instead of aggregated or calculated members. For example, create a sample report out of APSHELL defined with Activity down the rows and Time along the columns; this produces an MDX query because the Account dimension is an aggregated member (summation of three accounts)

- CE0004000 - Personal Costs (3)
  - CE0004010 - Bonus Expense
  - CE0004020 - Wages and Salaries
  - CE0004030 - Social Contributions

App	PLANNING	Planning Application
CATEGORY	ACTUAL	ACTUAL
MEASURES	PERIODIC	Periodic
P_ACCT	CE0004000	Personal Costs
P_ACTIVITY	A001	Total Activities
P_CC	All_CC	All Cost Centers
P_DATASRC	UPLOAD	Uploaded Data
RPTCURRENCY	LC	Local Currency
TIME	2008.TOTAL	2008.TOTAL

Line	Running the manufacturing lines	2008.JAN	2008.FEB	2008.MAR	2008.APR	2008.MAY	2008.JUN
Repair	Repairing/maintenance of manufacturing lines	2,299,412.50	2,298,758.87	2,299,412.50	2,299,412.50	2,299,412.50	2,299,412.50
Consult	Consulting						
Corp	Corporate management	18,072.60	17,300.19	18,072.60	18,062.10	18,072.60	18,072.60
Plant	Plant management	1,860,146.64	1,862,860.89	1,862,131.14	1,823,374.06	1,862,062.89	1,862,062.89
Corp_Act	Corporate activities						
Mfg_Act	Manufacturing activities						
A005	No activity						
A001	Total Activities	4,177,631.73	4,178,919.95	4,179,616.23	4,140,848.66	4,179,547.98	4,179,547.98

By looking in UJSTAT, this will produce an MDX query.

MDX Query Kernel	AP SHELL_...	PLANNING	17 27.10.2010	18:27:29	0,000000	0,422000	0,625000	36
Miscellaneous Processing Time	AP SHELL_...	PLANNING	18 27.10.2010	18:27:29	0,000000	0,000000	0,203000	0
Mddataset_create_object	AP SHELL_...	PLANNING	19 27.10.2010	18:27:29	0,000000	0,000000	0,203000	0
Mddataset_select_data	AP SHELL_...	PLANNING	20 27.10.2010	18:27:29	0,281000	0,281000	0,484000	0
Get_mdxi_data	AP SHELL_...	PLANNING	21 27.10.2010	18:27:29	0,000000	0,141000	0,625000	36
Mddataset_delete_object	AP SHELL_...	PLANNING	30 27.10.2010	18:27:30	0,000000	0,000000	0,625000	0
Process SQE Badl	AP SHELL_...	PLANNING	31 27.10.2010	18:27:30	0,000000	0,000000	0,625000	36

Instead of having the aggregated account in the key range, place the actual accounts as seen below.

App	PLANNING	Planning Application
CATEGORY	ACTUAL	ACTUAL
MEASURES	PERIODIC	Periodic
P_ACCT	CE0004010,CE0004020,CE0004030	#NODATA
P_ACTIVITY	A001	Total Activities
P_CC	All_CC	All Cost Centers
P_DATASRC	UPLOAD	Uploaded Data
RPTCURRENCY	LC	Local Currency
TIME	2008.TOTAL	2008.TOTAL

Line	Running the manufacturing lines	2008.JAN	2008.FEB	2008.MAR	2008.APR
Repair	Repairing/maintenance of manufacturing lines	2,299,412.50	2,298,758.87	2,299,412.50	2,299,412.50
Consult	Consulting				
Corp	Corporate management	18,072.60	17,300.19	18,072.60	18,062.10
Plant	Plant management	1,860,146.64	1,862,860.89	1,862,131.14	1,823,374.06
Corp_Act	Corporate activities				
Mfg_Act	Manufacturing activities				
A005	No activity				

Run RSDRI Query	AP SHELL_RJM	PLANNING	12 27.10.2010	18:36:48	0,000000	0,250000	0,359000	0
RSDRI Query Kernel	AP SHELL_RJM	PLANNING	13 27.10.2010	18:36:48	0,000000	0,250000	0,359000	27
Convert RSDRI Result	AP SHELL_RJM	PLANNING	17 27.10.2010	18:36:48	0,000000	0,000000	0,359000	0
Process SQE Badl	AP SHELL_RJM	PLANNING	18 27.10.2010	18:36:48	0,000000	0,000000	0,359000	27

You will notice, not only does this produce a RSDRI query, but it is faster and also brings back the same data. This can also work if the requirements for a report are to bring back full year totals (2009.Total). Instead of creating an MDX query by bringing in the aggregated amount, you can create formula's creating the twelve months in the year.

Taking the same report as above and changing the columns to the aggregated year will produce an MDX query.

App	PLANNING	Planning Application
CATEGORY	ACTUAL	ACTUAL
MEASURES	PERIODIC	Periodic
P_ACCT	CE0004010,CE0004020,CE0004030	#NODATA
P_ACTIVITY	A001	Total Activities
P_CC	All_CC	All Cost Centers
P_DATASRC	UPLOAD	Uploaded Data
RPTCURRENCY	LC	Local Currency
TIME	2008.TOTAL	2008.TOTAL

Line	Running the manufacturing lines	20,694,058.84
Repair	Repairing/maintenance of manufacturing lines	
Consult	Consulting	
Corp	Corporate management	159,553.28
Plant	Plant management	16,717,149.78
Corp_Act	Corporate activities	
Mfg_Act	Manufacturing activities	
A005	No activity	

In UJSTAT....

Axis Query Kernel	APSHELL_RJM	PLANNING	15	27.10.2010	18:43:52	0,000000	1,703000	1,812000	3
Miscellaneous Proc	APSHELL_RJM	PLANNING	16	27.10.2010	18:43:52	0,000000	0,000000	0,109000	0
MDX Query Kernel	APSHELL_RJM	PLANNING	17	27.10.2010	18:43:52	0,000000	1,703000	1,812000	3

By adding formula's (EVPRO's) and referencing all twelve months in the Key Range, as shown below, this will produce a RSDRI query.

App	PLANNING	Planning Application
CATEGORY	ACTUAL	ACTUAL
MEASURES	PERIODIC	Periodic
P_ACCT	#NODATA	#NODATA
P_ACTIVITY	Total Activities	Total Activities
P_CC	All Cost Centers	All Cost Centers
P_DATASRC	Uploaded Data	Uploaded Data
RPTCURRENCY	LC	Local Currency
TIME	2008.JAN,2008.FEB,2008.MAR,2008.APR,2008.MAY,2008.JUN,2008.JUL,2008.AUG,2008.SEP,2008.OCT,2008.NOV,2008.DEC	#NODATA

Line	Running the manufacturing lines	20,694,058.84
Repair	Repairing/maintenance of manufacturing lines	
Consult	Consulting	
Corp	Corporate management	159,553.28
Plant	Plant management	16,717,149.78
Corp_Act	Corporate activities	
Mfg_Act	Manufacturing activities	
A005	No activity	

In UJSTAT, a RSDRI query is produced.

Run RSDRI Query	APSHELL_RJM	PLANNING	12	27.10.2010	18:52:05	0,000000	0,203000	0,328000	0
RSDRI Query Kernel	APSHELL_RJM	PLANNING	13	27.10.2010	18:52:05	0,000000	0,203000	0,328000	3
Convert RSDRI Result	APSHELL_RJM	PLANNING	17	27.10.2010	18:52:05	0,000000	0,000000	0,328000	0
Process SQE Badl	APSHELL_RJM	PLANNING	18	27.10.2010	18:52:05	0,000000	0,000000	0,328000	3

A couple of notes on the above report, the formula was created using the function EVPRO and EVCVW to take into account the current action pane changing. The actual formula is as follows:

=EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".JAN,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".FEB,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".MAR,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".APR,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".MAY,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".JUN,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".JUL,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".AUG,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".SEP,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".OCT,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".NOV,"&EVPRO(\$H\$2,EVCVW(\$H\$2,"TIME"),"YEAR")&".DEC"

Secondly, notice the highlighted box over the column. It now shows LC because the EVDRE will not be able to interpret the description when multiple members are specified. To get around this, place a base member from one of the other dimensions, in this case RPTCURRENCY, and link to the Time dimension in the description.

Lastly, be careful when summing accounts across time frames as INC/EXP accounts do aggregate, but LEQ/AST accounts do not.

## Summary

In summary, when the above performance tips are properly utilized, EVDRE performance increases dramatically. In one customer's instance, these tips provided over 200% better performance than how it was originally designed.

	<i>Time (Original Report)</i>	<i>Cells (Original Report)</i>	<i>Time (New Report)</i>	<i>Cells (New Report)</i>
<i>Intersection 1</i>	<i>12:38</i>	<i>39,593</i>	<i>2:00</i>	<i>21,513</i>
<i>Intersection 2</i>	<i>Timed out at 19:19</i>	<i>344,953 *** did not finish</i>	<i>8:45</i>	<i>189,599</i>
<i>Intersection 3</i>	<i>Timed out at 20:03</i>	<i>344,332 *** did not finish</i>	<i>10:30</i>	<i>187,239</i>
<i>Intersection 4</i>	<i>Timed out at 35:40</i>	<i>857,783 *** did not finish</i>	<i>22:39</i>	<i>469,628</i>

As a best practice when developing a report or input template, make sure you completely understand the requirements, where the data is coming from, and how the data needs to be rendered. Then, spend ample time creating spec's and using UJSTAT to get the most performance out of the report as you can. In the end, the development of these reports/input templates may take longer, but the extra time will be well spent as they will perform much better than having one EVDRE with a lot of MDX.

---

## Related Content

For more information, visit the [Enterprise Performance Management homepage](#).

---

## Copyright

© Copyright 2011 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 Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, 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.