

Guidelines & Proven Practices for
IBM Planning Analytics & IBM Cognos TM1
Integration & Performance Optimization
with Cognos Analytics

Hong Huang
Technical Architect
IBM Cognos Analytics
IBM Data and AI Expert Labs
Email: huangho@us.ibm.com

Andreas Kugelmeier
Executive Consultant, FOPM
Planning Analytics Architect
IBM Data and AI Expert Labs
Mobile Phone: +1-215-384-7302
Email: kugelmeier@us.ibm.com

Michael Spataro
IBM Senior Consultant
IBM Cognos BI

Notices & Disclaimers

Copyright © 2018 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations and papers (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS document is distributed "AS IS" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Document Version History

Date	Version	Author	Description
2014-2016	1.0	Andreas Kugelmeier	Version 1.0
11/2016	1.1	Hong Huang & Michael Spataro	Add section on BI performance tuning/troubleshooting/logging
12/27/2016	1.15	Andreas Kugelmeier	Incorporate 1.1 changes, adjust layout, rename document, misc. additions & edits
01/09/2017	1.2	Hong Huang	Comments & edits
01/09/2017	1.3	Andreas Kugelmeier	First release version
03/27/2017	1.4	Andreas Kugelmeier	Misc. additions
06/20/2017	1.41	Andreas Kugelmeier	Add links in section 2.2.4
07/05/2017	1.42	Andreas Kugelmeier	Add section 4
11/26/2018	2.0	Andreas Kugelmeier	Misc. additions / new sections pertaining to Cognos Analytics
11/26/2018	2.1	Andreas Kugelmeier	Misc. edits
03/22/2019	2.2	Andreas Kugelmeier	Addtl. Information on caveats when logging into different namespaces (section 'Before You Start')
06/07/2019	2.3	Andreas Kugelmeier	Add link to Please see Cognos Analytics on Cloud Dedicated and Planning Analytics on Cloud integration - Next steps
08/14/2019	2.4	Andreas Kugelmeier	Add recommendation to leverage PA V2.0 hierarchies when having multiple (alternate) rollups in a dimension and when using element security with CA 'filler' members enabled
05/06/2020	2.5	Andreas Kugelmeier	Updates to paragraph re Filler-member functionality, pertaining to CA 11.1

Table of Contents

1. About this Document	6
2. Preparation: specify a measures dimension	7
3. Guidelines for optimal performance and functionality of TM1 with Cognos BI	8
3.1 General Guidelines	8
3.2 Connecting to TM1 with Cognos Analytics, using the Rest API	8
3.2.1 Before you start	8
3.2.2 Creating a Data Server Connection	9
3.2.3 Creating a Data Module	12
3.2.4 Creating a Data Package	13
3.2.5 Create a new data server connection via the legacy Cognos BI Administration screen	14
3.3 Writing Efficient OLAP Queries	17
3.4 OLAP Query Modes & Configuration in Cognos BI	18
3.4.1 BI 10.2.* - TM1 Architecture	18
3.4.2 DQM & LOLAP	18
3.4.3 Native TM1 MDX Engine vs Bluenose Engine (Default)	19
3.4.4 Other Links	19
3.5 TM1 & Cognos BI Dimension Element/Member 'Hierarchies'	21
3.6 Filtering	23
3.7 Aggregations/Summations	24
3.8 Relative Time	24
3.9 Member Caption	25
3.10 Dimensions & Hierarchies in Report Studio (example)	26
3.11 TM1 Logging of Cognos BI-related queries	26
3.12 Considerations when using Element Security	27
3.12.1 Introduction to TM1 Element Security	27
3.12.2 TM1 Element Security Behavior	27
4. Troubleshooting Performance Issues in Cognos BI	29
4.1 Report Studio Query Properties	29
4.2 Viewing MDX/SQL	29
4.3 Tracing MDX/SQL	29
4.3.1 Native SQL Tracing when Using Dynamic Query Mode	29
4.3.2 Cognos Performance Monitoring and Analysis (PMA)	29
4.3.3 Whole-System Analysis of Idle Time (WAIT) Tool	29
4.3.4 Using Dynamic Query Analyzer	30
4.4 BI Server Tuning and Monitoring	32

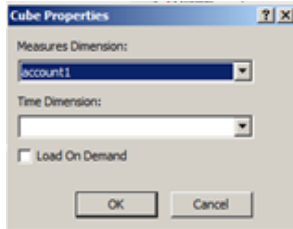
5. Logging of Cognos BI queries in TM1	33
5.1 Cognos BI and Cognos Analytics: MDX Logging	33
5.2 Cognos Analytics: Rest API Logging	33

1. About this Document

This document is geared at introducing key design practices and guidelines for integrating IBM Planning Analytics (and IBM Cognos TM1) with Cognos BI (including Cognos Analytics). Furthermore, the paper outlines corresponding performance tuning, optimization and logging practices in Cognos BI.

2. Preparation: specify a measures dimension

- You **need to specify a Measures Dimension for each cube that you are using in Cognos BI**. BI will not allow you to create a package against TM1 unless the cube(s) have a measures dimension defined. To specify a measures dimension, right mouse click on TM1 cube, select a cube dimension as the measures dimension (you do not need to define a time period dimension).



- Measures Dimensions in Cognos BI do not allow a hierarchical view or drill-down within a report. Consequently, only sufficiently 'flat' dimensions should be used as measures dimensions for Cognos BI. For example, only use an 'account' dimension as a measures dimension if the dimension contains sufficiently few elements and few (if any) rollups.
- Create a separate measures dimension (with 'Value', 'Currency', 'Units'... for example) ONLY if you cannot use any other dimension as a measures dimension OR if your cube is sufficiently small and fast and the TM1 Server has and will have ample memory reserves.** Reason: an additional dimension can increase memory requirements of a cube by up to 40-50%.
- => look for suitable measures dimensions first: A flat dimension like 'Currency' can be a good candidate for a measures dimension because currency dimensions typically do not contain hierarchical rollups that are needed for analysis/reporting purposes, i.e. currency dimensions tend to be 'flat'. Flat dimensions can be a good candidate for a measures dimension if only a few of its elements are selected in the context of a report. A flat dimension like 'invoice number' for example would not be a good measures dimension because it will be very likely that one would have to report against many invoices.
- Do not use Aliases in your measures dimension. You will not be able to use Measures Dimension Alias names in BI (Aliases are supported for non-Measures dimensions).
- When building the TM1 models, attempt to have either the same or same type of measures dimension for cubes that are closely related. Example: you would want the same measures dimension in a summary and detail cube because this will lead to much easier report development, i.e. do not 'switch' the measures dimension within one 'family'/sub-model/group of cubes. Now, let's assume you have a summary cube with a currency dimension. Your detail cube is USD only and hence does not need a currency dimension. But it does need a measures dimension. Other existing dimensions are not a good candidate for a measures dimension. Options:
 - create a measures dim for both summary and detail cube OR
 - include the currency dimension in the detail cube (and process to USD only) and use it as the measures dim in both cubes.
- In General: If your TM1 model already has a measures dimension, use it and don't worry about the above (unless you find that the cubes are very large and/or you can significantly speed up performance by eliminating the extra dimension)

3. Guidelines for optimal performance and functionality of TM1 with Cognos BI

3.1 General Guidelines

- Build fast TM1 models and leverage those in Cognos BI / Cognos Analytics.
- Do not just build a transformer-like (= PowerCube-like) cube with TM1, because that would equate to downgrading TM1.
- Use the TM1 Rest-API to connect Cognos Analytics to TM1 (vs. the C-API):

3.2 Connecting to TM1 with Cognos Analytics, using the Rest API

Also, please review [Cognos Analytics on Cloud Dedicated and Planning Analytics on Cloud integration - Next steps](#)

3.2.1 Before you start

Please ensure that you have sufficient PA access credentials to connect to PA and view the cubes/metadata that you want to leverage in CA.

Note that even if/when using the same ID/email and authentication mechanism,

- when logging in to CA, you will enter as a CAM/BI/CA Namespace user (unless you enter CA via the PA Workspace Portal), and
- when logging into PA, you will enter as a Planning Analytics Namespace user.

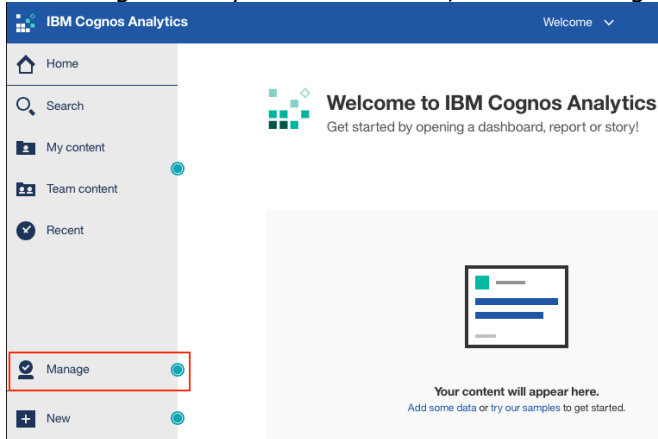
Now, when you log into CA and then establish a connection to PA (as described in the following sections) your CAM namespace user/ID will be added to the PA instance (if it does not exist there yet). The user whose credentials you will be using may hence not yet be part of a CAM group that was imported into PA, i.e. your user may not have any access credentials for PA yet. When you create a data source or data server connection and test it, the test just ensures connectivity can be established, i.e. when you test a connection, the connection will succeed even if your user is not allowed to see any TM1/PA data/metadata.

If you encounter problems with creating a data module or data package (see below), where no metadata can be retrieved, then this is an indication that your user is not part of any TM1/PA user group. => Log into PA and in PA security, look up your CAM user, and then assign a user group to this user, and then go back to CA to re-attempt creating the package and/or data module.

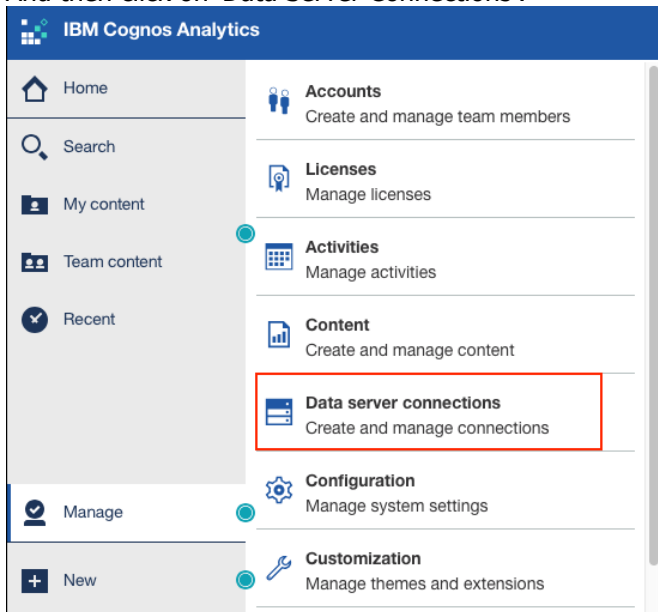
Note that the above step should only be necessary for the user who first configures connectivity to PA from within CA, and also only if this is done via a direct CA login vs a login into CA via PA.

3.2.2 Creating a Data Server Connection

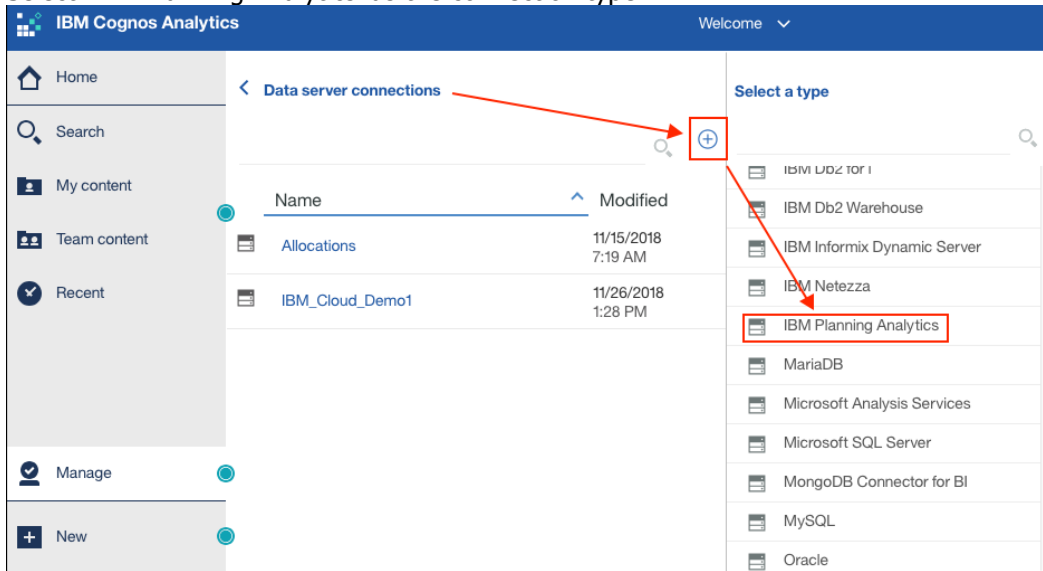
In the Cognos Analytics Home Screen, click on 'Manage':



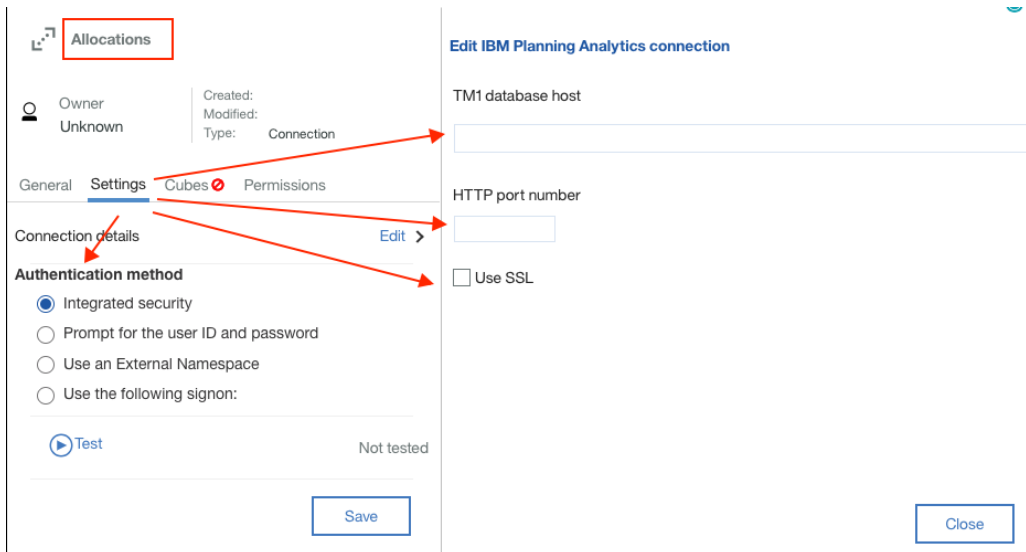
And then click on 'Data Server Connections':



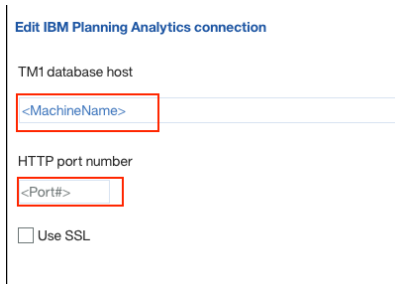
Select 'IBM Planning Analytics' as the connection type:



Name the connection, and configure (i) Authentication Method, (ii) TM1 Database Host (= Machine Name), (iii) HTTP Port Number (the Rest API Port of the TM1 DB you want to connect to), (iv) SSL:



The Port Number is the RestAPI Http Port Number for the TM1 database that you want to connect to (click [here](#) for instructions on how to configure a TM database for access via RestAPI). On cloud, enter 'data' (without parenthesis) as the machine name.



In the menu to the left, click save:

Allocations

Owner: Unknown
Created: [blank]
Modified: [blank]
Type: Connection

General **Settings** Cubes Permissions

Connection details [Edit >](#)

Authentication method

- Integrated security
- Prompt for the user ID and password
- Use an External Namespace
- Use the following signon:

[Test](#) Not tested

[Save](#)

And then test the connection:

Allocations

Owner: ANDR ... LMEIER
Created: 11/15/2018 7:19 AM
Modified: 11/15/2018 7:23 AM
Type: Connection

General **Settings** Cubes Permissions

Connection details [Edit >](#)

Authentication method

- Integrated security
- Prompt for the user ID and password
- Use an External Namespace
- Use the following signon:

[Test](#) [Success](#) [i >](#)

[Save](#)

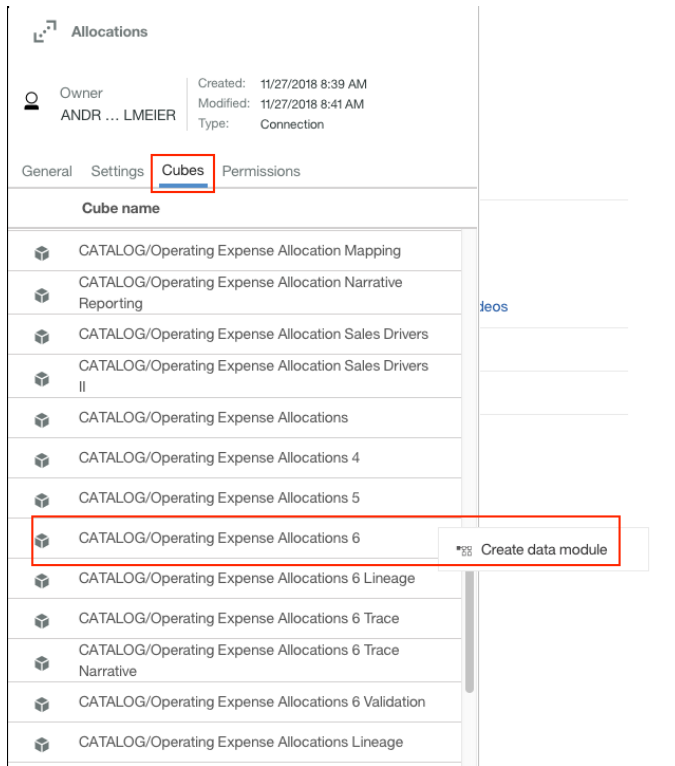
Test results

Messages

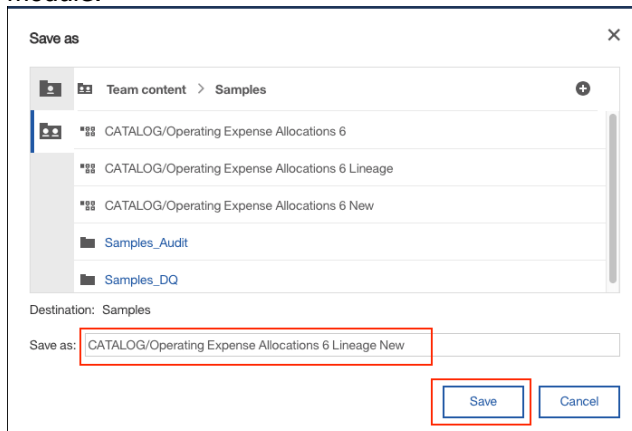
XQE-DS-0015 TM1 Server Name: Allocations: "11.2.00000.27".
XQE-DS-0015 TM1 Admin Host: : "11.2.00000.27".
XQE-DS-0016 Time to connect: 1ms.

In order to use the data server connection for a report or dashboard (except for Cognos BI/CA Workspace), you need to create a data package (BI and CA) or a data module (CA):

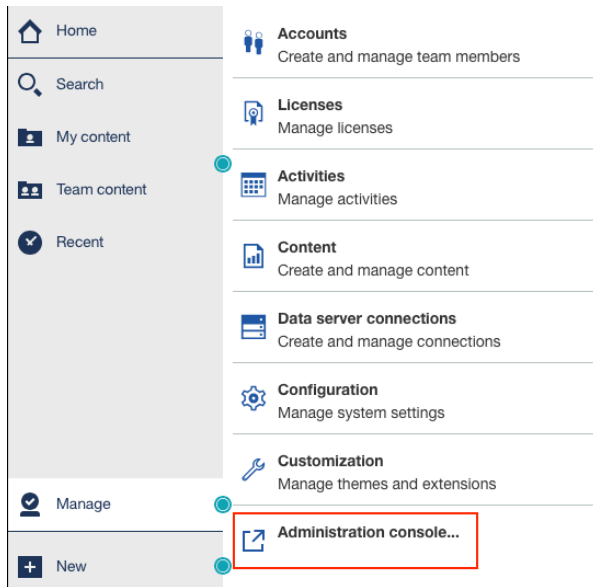
3.2.3 Creating a Data Module



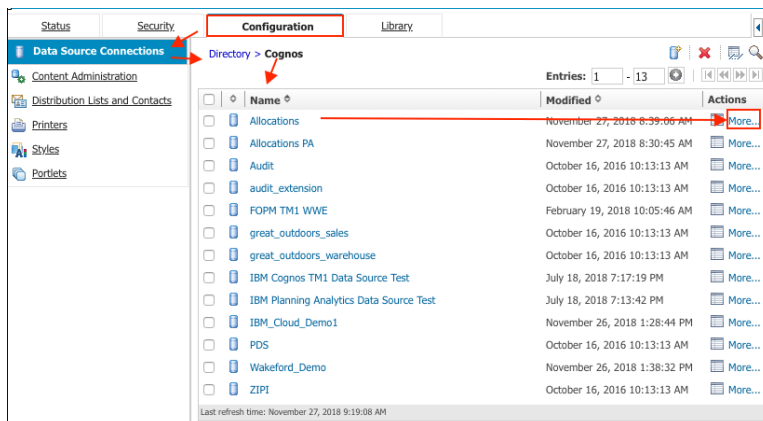
Click, 'Create a data module', then name the data module, and select a folder/location for the data module:



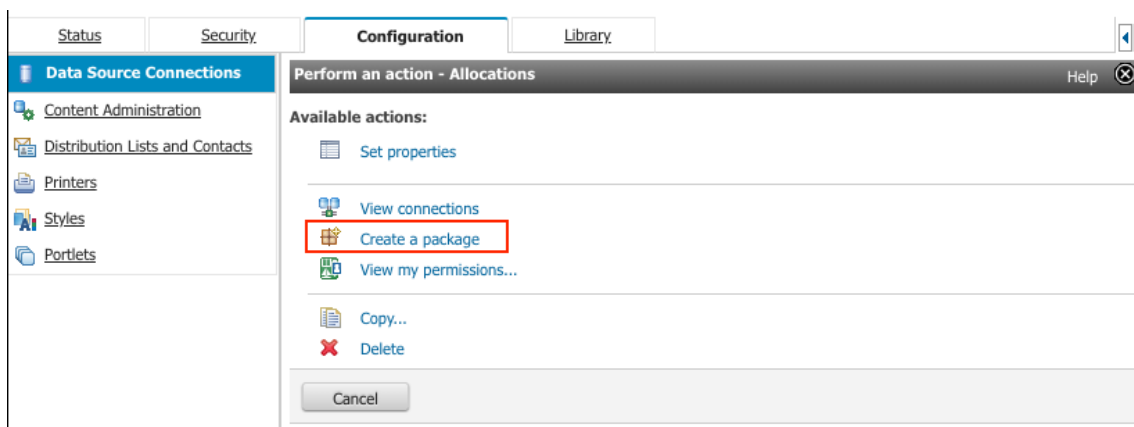
3.2.4 Creating a Data Package



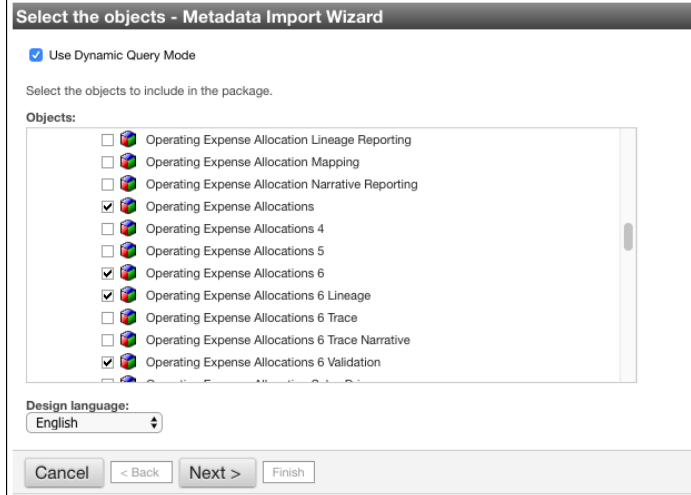
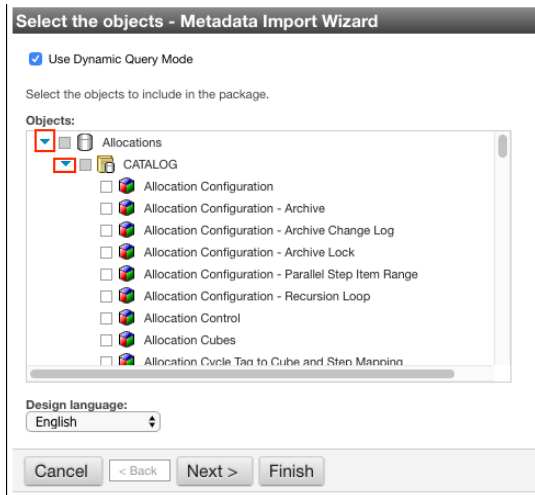
Go to the data server connection you created, and click on 'More':



Click 'Create a package':



Expand the Database and select the cubes you want to use in the package:

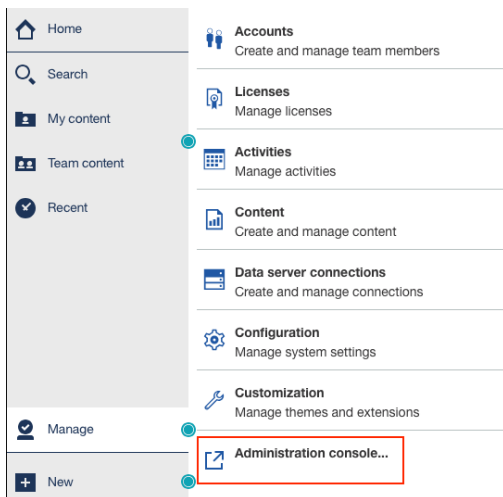


Click 'Next', select a language:

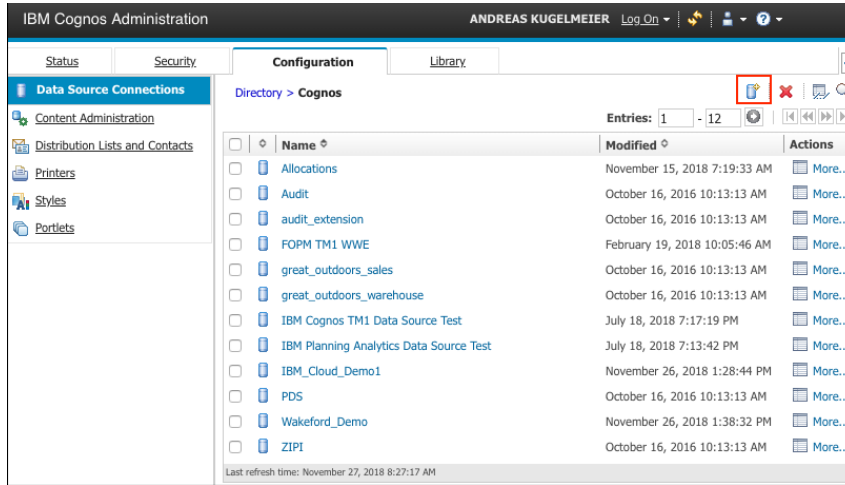


Then click 'Finish' to create the package.

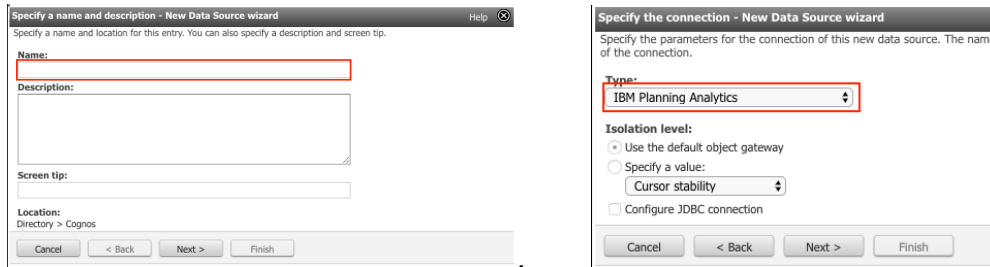
3.2.5 Create a new data server connection via the legacy Cognos BI Administration screen



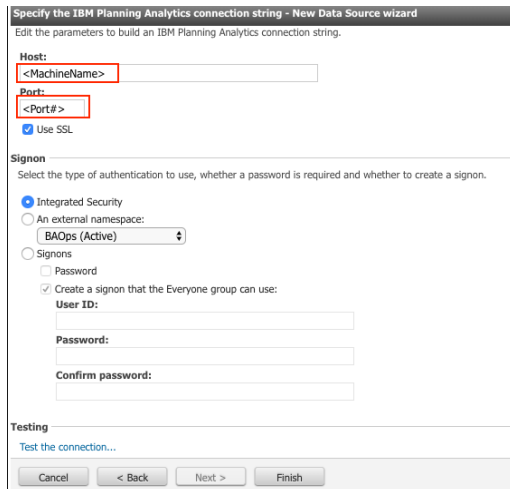
Create a new data source connection:



Name the Connection, click 'Next', and select 'IBM Planning Analytics' as the data source type:



Configure (i) Authentication Method, (ii) TM1 Database Host (= Machine Name), (iii) HTTP Port Number (the Rest API Port of the TM1 DB you want to connect to), (iv) SSL:



Select 'Create a Package' (you can also do this later: see previous chapter):



Select the objects - Metadata Import Wizard

Use Dynamic Query Mode

Select the objects to include in the package.

Objects:

- Allocations PA
 - CATALOG
 - Allocation Configuration
 - Allocation Configuration - Archive
 - Allocation Configuration - Archive Change Log
 - Allocation Configuration - Archive Lock
 - Allocation Configuration - Parallel Step Item Range
 - Allocation Configuration - Recursion Loop
 - Allocation Control
 - Allocation Cubes
 - Allocation Cycle Tag to Cube and Step Mapping

Design language:
English

Cancel < Back Next > Finish

Select the objects - Metadata Import Wizard

Use Dynamic Query Mode

Select the objects to include in the package.

Objects:

- Operating Expense Allocation Lineage Reporting
- Operating Expense Allocation Mapping
- Operating Expense Allocation Narrative Reporting
- Operating Expense Allocations
- Operating Expense Allocations 4
- Operating Expense Allocations 5
- Operating Expense Allocations 6
- Operating Expense Allocations 6 Lineage
- Operating Expense Allocations 6 Trace
- Operating Expense Allocations 6 Trace Narrative
- Operating Expense Allocations 6 Validation

Design language:
English

Cancel < Back Next > Finish

Click 'Next', select a language:

Select languages - Metadata Import Wizard

Select the languages to include in the package.

Languages:

- Catalan
- Chinese
- Croatian
- Czech
- Danish
- Dutch
- English
- Estonian
- Faroese
- Finnish

Cancel < Back Next > Finish

and then 'Finish' to create the package.

3.3 Writing Efficient OLAP Queries

A good paper on writing efficient OLAP queries in Cognos BI can be found at <http://www.ibm.com/developerworks/data/library/cognos/page128.html>

developerWorks > Technical topics > Information Management >

Writing Efficient OLAP Queries

Discussion on some of the common pitfalls that result in poorly performing reports. These are resolved using more appropriate filter expressions and by reducing the potential size of intermediate result sets through the use of filter functions. 01 January 2009

[→ Try IBM Cognos Business Intelligence](#)

[↓ Get the download](#)

Just as we have developed good habits for writing relational queries through understanding SQL and databases and databases optimizers, we also have to understand the nature of OLAP queries and OLAP providers and ensure we develop good habits for the creation of OLAP queries. This document discusses several OLAP query subjects dealing with OLAP query performance and optimization. The application of the habits ingrained in this document should help you build better, faster, more efficient OLAP queries.

In this article

Topics covered include:

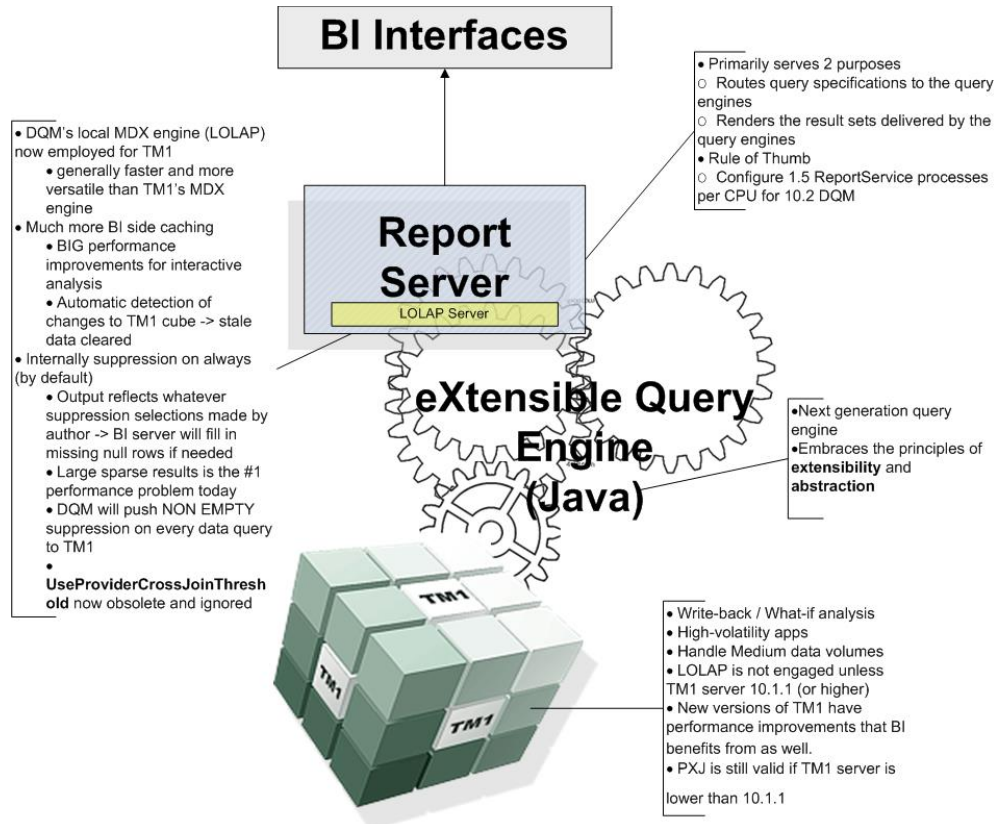
- INTRODUCTION
- OPTIMIZING QUERIES FOR OLAP CROSS JOIN BEHAVIOUR
- STEP-BY-STEP QUERY CREATION
- SCALAR FUNCTIONS IN OLAP EXPRESSIONS
- EXPLICIT AGGREGATE USAGE
- CONCLUSIONS

Download

Description	Name	Size
Article in PDF format	writing_efficient_olap_queries.pdf	454KB

3.4 OLAP Query Modes & Configuration in Cognos BI

3.4.1 BI 10.2.* - TM1 Architecture



3.4.2 DQM & LOLAP

- Performance is best when using [DQM](#) in Cognos BI 10.2.1 or higher with TM1 10.1.1 or higher. Performance is typically best through this architecture because the Cognos BI server will employ DQM's Java MDX engine (to query TM1) which is also known as LOLAP (Local OLAP engine).
- LOLAP is generally faster than the MDX engine in TM1 and can perform certain calculations that the TM1 MDX engine cannot.
- LOLAP allows for considerably more BI side caching. Avoiding round trips to the TM1 server may allow for reductions in user wait times during interactive analysis.¹ Note that there is no need to manage the LOLAP cache for TM1 data sources because LOLAP will automatically detect changes to TM1 cubes and will in turn automatically clear any stale data from its memory.
- LOLAP for TM1 makes use of the TM1 Java API which enables faster loading of members than previously used interfaces.
- By default, LOLAP for TM1 always applies NULL suppression in its queries to TM1 (NON EMPTY clauses are submitted for all data queries). This enables only the minimized result sets to be transferred from the TM1 server to the Cognos BI server. Note that the Cognos BI report outputs will

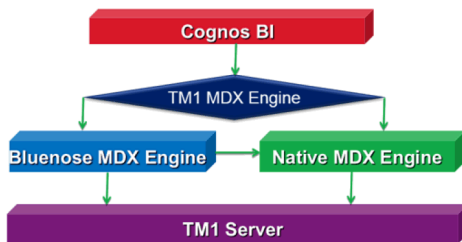
¹ as of TM1 MTQ, BI-side caching may no longer render advantages as the TM1 queries are likely to be much faster than the cache-based BI LOLAP queries

correctly reflect suppression options specified in the Cognos BI authoring interfaces - the Cognos BI server will fill in any missing null value intersections that are needed to produce the requested output.

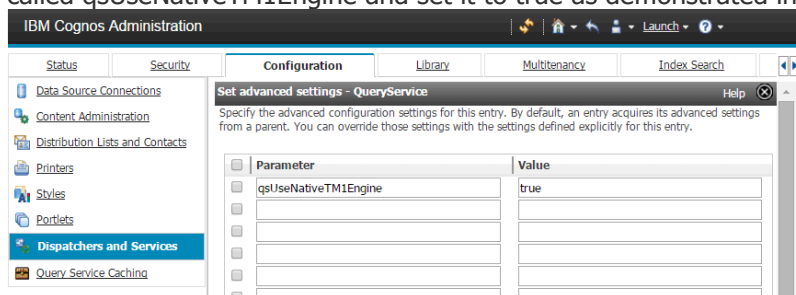
Memory Guidelines for Dynamic Query Mode: In production environments where Cognos Dynamic Cubes are not employed, it is recommended to set the query service JVM initial heap size to 4 GB and the limit to 8 GB. See <https://www.ibm.com/support/docview.wss?uid=swg21587457> for details.

3.4.3 Native TM1 MDX Engine vs Bluenose Engine (Default)

- [If you are using Cognos Business Intelligence 10.2.2 or higher along with TM1 10.2.1 FP3 and higher, then you can improve performance in some reports by removing a layer of processing in the Bluenose MDX engine such that the BI server calls upon TM1's Native MDX engine directly.](#) There is no longer a need to use Bluenose for the complex MDX queries since the MDX engine in BI (a.k.a. LOLAP) resolves those, and TM1's native MDX engine is better at the simpler queries, therefore performance can be improved by avoiding the overhead of having to go through Bluenose: In Cognos BI 10.2.2, by default the Bluenose MDX Engine is still always used and will delegate certain requests to the Native MDX Engine. The reason for this default is to avoid the risk of undesired changes to saved BI content that might occur in switching MDX engines. Most reports won't require any changes but some might².



To bypass the Bluenose MDX Engine, set an advanced setting on the BI query service called `qsUseNativeTM1Engine` and set it to true as demonstrated in the following screenshot:



3.4.4 Other Links

- [Setting up connectivity for IBM Cognos TM1 data sources](#)
- [Creating a project, connection, and package for an IBM Cognos TM1 data source](#)
- [Enable packages in Framework Manager to use dynamic query mode](#)

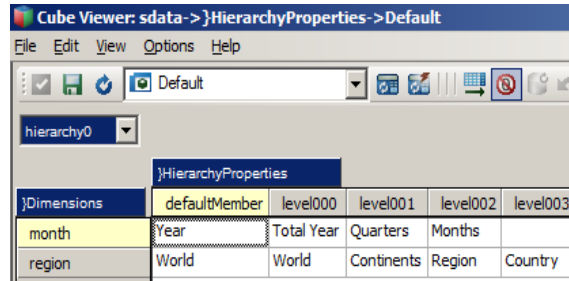
²² The most likely cause of seeing different results in a report after bypassing the Bluenose MDX engine is caused by under fed cubes - please see http://www.ibm.com/developerworks/data/library/cognos/financial_management/analytics/page620.html for more information on under fed cubes.



[Overriding the query mode for packages or query testing](#)
[Troubleshooting TM1/BI Data Source Connections](#)

3.5 TM1 & Cognos BI Dimension Element/Member 'Hierarchies'

- In TM1, set TM1 **}HierarchyProperties for balanced dimensions** in order to use hierarchy metadata in Cognos BI.³

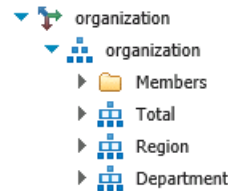


Use meaningful level names for hierarchies that represent consistent hierarchical structure of a dimension for ease of use. Example:

In TM1:

}Dimensions	
}HierarchyProperties	Organization Reporting
defaultMember	Total Company
level000	Total
level001	Region
level002	Department

In Cognos BI:



Defining level names in TM1 provides BI report authors the flexibility to effectively use MDX level or set expressions to retrieve data.

- If you have a dimension with a balanced hierarchy, it does makes sense to define the hierarchy in }HierarchyProperties.cub in TM1 as this eases report development
- Do not define hierarchy level names for dimensions with unbalanced hierarchies**
- Do not define hierarchy level names for dimension with multiple & alternate hierarchies, unless all alternate hierarchies are balanced and the same level names apply to the alternate levels.**
- For dimensions with ragged/unbalanced hierarchies it is a good practice to create a TM1 dimension attribute like 'TM1 Level' and to populate this attribute with TM1 Level values (Level 0 starting at Leaf Level) using TM1 Function ELLEV (<DimensionName>, <ElementName>). You can then use this attribute to filter by TM1 Level if needed.
- Use common hierarchy level names (like Level1, Level2,...) for
 - balanced hierarchies where the depth and/or meaning of the hierarchy levels change
 - for dimensions with alternate hierarchies & where the levels (level numbers) hence may have different meanings. For example, if you have a Time Period Dimension with multiple hierarchies for Monthly (MTD), YTD, CY vs P, naming the numeric levels will not work as level 1 in one hierarchy may be 'Quarters' but in another hierarchy, it may be '<YYYY MM> YTD'.

³ [Using Hierarchy Named levels with TM1 Dimensions](#)

- Use the TM1 Function **RefreshMDXHierarchy()** to refresh **TM1 hierarchy level names in BI after changes to }HierarchyProperties**
- When building Reports, **always keep in mind that Cognos BI uses Member Unique Names as element qualifiers!**. Every MUN in a TM1 cube is defined with the full path of a member in a hierarchy:
`[Cube].[Dimension].[Hierarchy].[Level] ->:[TM].[Dimension].[Hierarchy].[@MEMBER].[Element]`

This is the way each member in the cube is uniquely identified not only by its principal name but also by its location within the dimension hierarchy. Subsequently, if 'hard-coded' member references are used within a report and if the hierarchy changes above the level of the referenced member, the element reference in the report will be invalidated.

=> for dimensions with multiple hierarchies, Cognos BI will assign an individual Member Unique Name (MUN) for each occurrence of the element(s), specific to the element parent. A member can occur as many times as it is used in as many hierarchies within the dimension. If an element is part of two hierarchies, it will occur as two members with two distinct MUNs. A leaf-level list of all elements will include all leaf-level MUNs and therefore, TM leaf elements may occur more than once.

The screenshot illustrates how Cognos BI assigns unique Member Unique Names (MUNs) to the same element based on its parent in the hierarchy. Two instances of 'L Series 1.8 L Wagon 4WD' are shown with different MUNs: one under 'L Series 2.0 L Wagon 4WD' and another under 'L Series 1.8 L Wagon 4WD'. The IBM Cognos Viewer table shows data for these members, with the same element name appearing multiple times due to the different hierarchies.

Dimension	Sales	Budget	Actual	Variance
L Series 1.8 L Wagon 4WD				
L Series 2.0 L Wagon 4WD				
L Series 1.6 L Sedan	37230.42	36604.98	433.44	
L Series 1.6 L Sedan				
L Series 2.0 L Sedan	40645.02	40380.59	-264.43	
L Series 1.8 L Wagon				
L Series 2.0 L Wagon				
L Series 1.8 L Wagon 4WD				
L Series 2.0 L Wagon 4WD				
L Series 1.6 L Convertible	233.79	233.79		
L Series 1.8 L Convertible				
L Series 2.0 L Convertible				

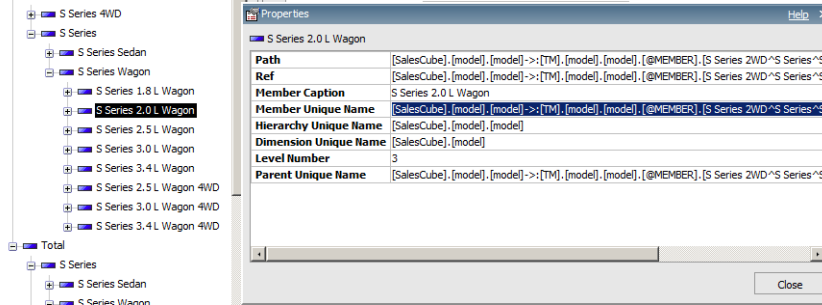
=> if the Cognos BI report is to contain a distinct list of elements, or a calculation such as a summary is to be applied, it is a common practice to apply an MDX-based filter (as an MDX-based set expression for example) to the report that will filter by elements of a particular hierarchy. In the below example, a report is filtered by the 'Total' Hierarchy (`DESCENDANTS ([Total],1,after)`) and by attribute 'TM1 Level' = 0 (`[SalesCube].[model].[model].[TM1 Level] = '0'`):

The screenshot shows the 'Set Expression' editor with the following expression definition:
`FILTER (DESCENDANTS ([Total],1,after), [SalesCube].[model].[model].[TM1 Level] = '0')`

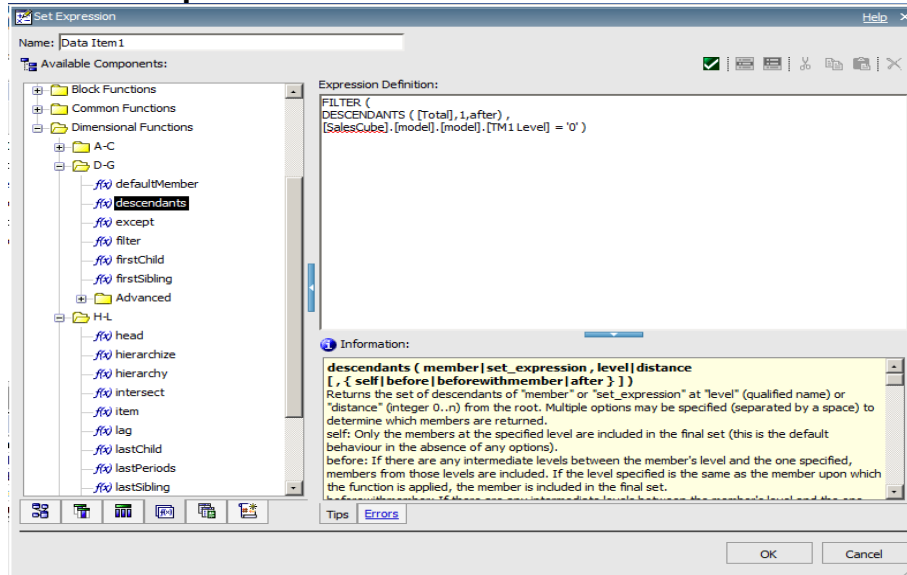
Information:
descendants (member)set_expression [level] distance
 [, { set|before|beforewithmember|after }]
 Returns the set of descendants of 'member' or 'set_expression' at 'level' (qualified name) or 'distance' (integer 0..n) from the root. Multiple options may be specified (separated by a space) to determine which members are returned.
 before: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. If the level specified is the same as the member upon which the function is applied, the member is included in the final set.

3.6 Filtering

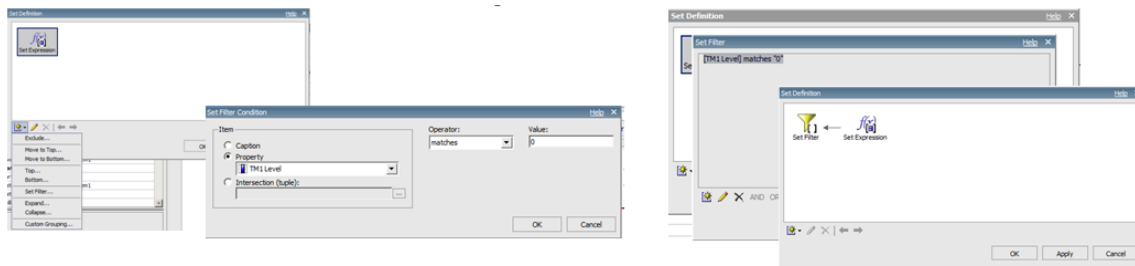
- Filtering with a MUN is fastest



- Filter data using multidimensional functions/expressions (MDX), as in the aforementioned example in section <3.5>.



Using such MDX expressions for filtering typically results in faster queries than filtering via the member property (=Attribute) using the UI. In other words, the above filter (using MDX) will result in better performance than adding a filter using the filter wizard like:



- If you need a filter that is not available in TM1, add the attribute (calculation) in TM1 and leverage the attribute in Cognos BI for MDX-based filtering. For example: in the above screenshot, 'TM1 Level' is such an attribute that was added in TM1 to filter base on TM1 level #.

3.7 Aggregations/Summations

- In most dimensions, every non-leaf member is the roll-up (i.e., the natural summary) of its children. As opposed to having summary values computed in real-time, if you use the parent member this value will come directly from the data source (the TM1 engine), resulting in significantly faster performance compared to aggregations performed in either the BI reporting layer or BI MDX engine.
- When using the Summary toolbar buttons, requesting an automatic summary instead of an explicit summary (such as Total) allows the summaries, when possible, to be delegated to and optimized by the data source system, resulting in better performance. This optimization is especially useful when detail summaries are required, such as in a list report.
- When using the expression editor, the function that computes automatic summaries is `Aggregate()`. You can't use this technique if you need a summary set of members that is not a complete set of children (e.g., a TopCount set). In this case, the summaries must be computed on demand.

3.8 Relative Time

- Use a TM1 continuous time dimension for Time-Based analysis. For proven Time Dimension designs in TM1, please refer to the document 'Proven TM1 Practices for Continuous Time Period Dimension Design and Time-related Analysis incl Design Template'.
- In such a time dimension, there are specific hierarchy rollups and elements for 'YTD', 'Current Month/Qtr/...', 'Prior Month/QTR/...' etc. and associated variances. Example:

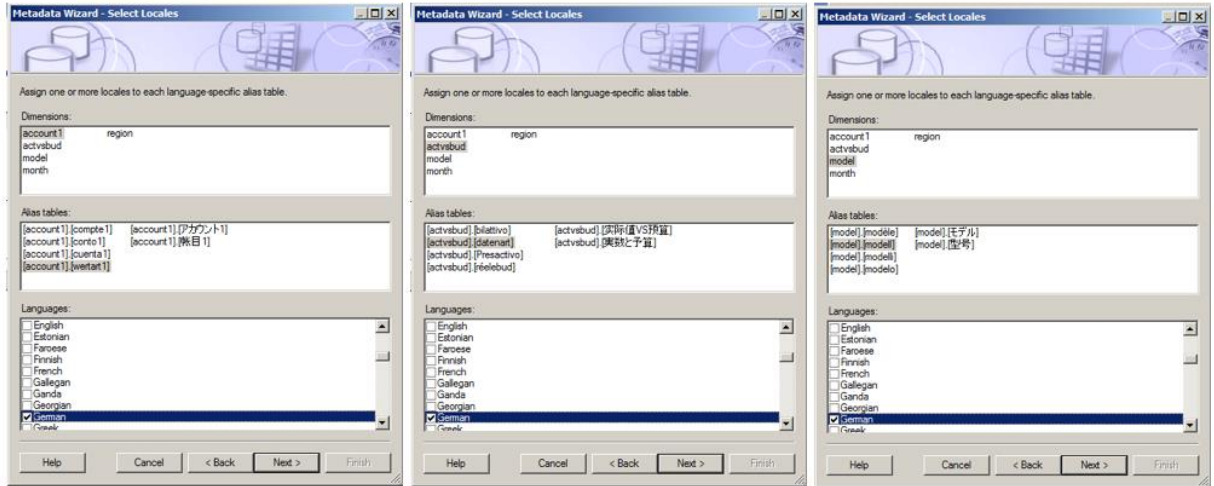
- ☐ Months Grouped
 - ☐ Current Month
 - ☐ Prior Month
 - ☐ Prior Month Variance
 - ☐ Prior Month Variance (%)
 - ☐ Same Month Prior Year
 - ☐ Prior Year Variance
 - ☐ Prior Year Variance (%)
- ☐ Quarters Grouped
 - ☐ QTD
 - ☐ Prior QTD
 - ☐ QTD Variance
 - ☐ QTD Variance (%)
 - ☐ Prior Year QTD Variance
 - ☐ Prior Year QTD Variance (%)
 - ☐ Same QTD Prior Year
- ☐ Years Grouped
 - ☐ YTD
 - ☐ Prior YTD
 - ☐ YTD Variance
 - ☐ YTD Variance (%)

- As stated above, such a time dimension shall not be exposed using named hierarchy levels in `}HierarchyProperties`. Named levels will not work as level 1 in one hierarchy may be 'Quarters' but in another hierarchy it may be '<YYYY MM> YTD' or 'Current Month'. => for time period dimension with relative time and/or alternate time-related rollups:

		}HierarchyProperties			
}Dimensions	defaultMember	level000	level001	level002	level003
Month	Year	Total Year	Quarters	Months	

3.9 Member Caption

You may want to use alias names as member captions and keep element value as business keys for efficiency and minimize impact of code changes on reports. You define it in Framework Manager when you import the cube in Metadata wizard. Example for selecting alias for Plant dimension:



3.10 Dimensions & Hierarchies in Report Studio (example)

The screenshot displays the Cognos Report Studio interface. On the left, a tree view shows a cube model structure with nodes for 'region', 'model', and 'accounts'. Three callout boxes highlight specific areas:

- Elements:** A callout box around the 'region' node, listing 'World', 'Europe', and 'Americas'.
- Hierarchies as defined in TM1 and HierardchProperties.cub:** A callout box around the 'model' node, pointing to the 'HierarchyProperties' table in the 'Cube Viewer' window.
- Attributes:** A callout box around the 'model' node, pointing to the 'ElementAttributes_model' window, which lists attributes like 'Engine Size', 'Hier Level', 'TM1 Level', and various language labels for 'model'.
- Measures as per Cube Properties:** A callout box around the 'accounts' node, pointing to the 'Subset Editor: sdata->account1' window, which shows measures like 'Gross Margin%', 'Price', 'Units', 'Gross Margin', 'Sales', and 'Variable Costs'.

The 'Cube Viewer' window shows a table with dimensions and their levels:

Dimensions	defaultMember	level000	level001	level002	level003
month	Year	Total Year	Quarters	Months	
region	World	World	Continents	Region	Country

3.11 TM1 Logging of Cognos BI-related queries

Use MDX and Rowset logging to temporarily debug/analyze Cognos BI queries against TM1, by adding the following rows to the `tm1s-log.properties` file.

```
log4j.logger.TM1.Mdx=DEBUG
log4j.logger.TM1.Mdx.Interface=DEBUG
log4j.logger.TM1.MdxViewCreate=DEBUG
log4j.logger.TM1.Rowset=DEBUG
log4j.logger.TM1.MdxLogger=DEBUG
```

3.12 Considerations when using Element Security

3.12.1 Introduction to TM1 Element Security

PowerCube / Transformer models do not support cell/element security. A Transformer cube will show the aggregations for the lower level children that the user has access to. For example, if a user is only allowed to see CA data, then the USA node will show only CA data for that user. As Transformer cubes support visual totals, a user can see all dimension & hierarchy elements (while not being allowed to view all corresponding data). If TM1 Element Security is in place, the member trees in Cognos BI / CA will only show the Elements for which the user has been granted READ access (or higher). Consequently, reports built against a TM1 cube with Element Security will not behave the same way as using PowerCubes.

3.12.2 TM1 Element Security Behavior

- **Rollup values remain unchanged:** In TM1, the value of a non-leaf member does not change if some or all of its descendants are inaccessible. While this feature results in highest data-consistency & integrity (a parent node's data cannot change due to different security profiles), the report development approach may have to be adjusted if Transformer Models are to be replaced by TM1.
- **Inaccessible root members – parent-child hierarchies:** In a parent-child hierarchy, if a root element is inaccessible, the highest level accessible descendants effectively become root elements, with level ordinal 0, and the level ordinals of their descendants are adjusted accordingly. This may result in the hierarchy becoming unbalanced.
- **BUT: As of TM1 10.2 FP1 and BI 10.2.1 FP1 updaters one can have Cognos BI use so-called 'Filler Members' to 'fill' in for the TM1 members that the user does not have access to.** Hierarchy Level Ordinals for the visible members will be the same as if the user were allowed to see all elements. Click [here](#) for details on this functionality. **From a usability point of view, the use of this functionality is highly recommended for dimensions secured with TM1 ElementSecurity. As of CA 11.1, the filler-member functionality is enabled per default.**⁴
- **With the [Cognos Analytics 'filler-member' functionality](#) on dimensions with ElementSecurity and that have multiple (alternate) rollups, it is highly recommended to implement the alternate rollups (the different hierarchies) as different PA V2.0 Hierarchies.** This is because the filler members are not 'rollup-specific'. If there are two hierarchies in one dimension, and a user does not have access to the root element of either, there will be not two, but only one filler member. As a result, the hierarchies for a restricted user appear 'nested'. If however the different hierarchies are in separate hierarchy containers, the filler members are applied to the one hierarchy (per hierarchy container) only and will appear as expected.
- **Inaccessible non-root members:** If a member is not accessible, but its parent is accessible, then all of the descendants will effectively become inaccessible.
- **Members with no accessible children:** If a member is accessible, but none of its children are accessible, then the hierarchy is in effect unbalanced.
- If converting from Power Cubes (Transformer) to TM1, and if 'Transformer-like' access to all elements is to stay in place (but with TM1 restricting access to the associated data): Use a combination of Element Security and Cell Security to (a) Give All Users Read Access to all [Elements](#)⁵ via Element

⁴ filler-members functionality can be disabled as of CA 11.1.3

⁵ Note that as of TM1 10.2, Cell Security performance can perform much faster because a cell security cube only needs to contain the dimensions needed for defining cell security. Example: if your securing a cube against Cost Center and Version only, your cell security cube only needs to include those two dimensions plus the }Groups dimension




Security against a specific 'read all' group, while (b) restricting Cell Level data (using TM1 CellSecurity) according to Element Security settings for all other groups.

4. Troubleshooting Performance Issues in Cognos BI

4.1 Report Studio Query Properties

If applicable, consider setting/enabling performance-related [Query properties](#) such as [Use for Parameter Info](#) & [Query Prioritization](#).

4.2 Viewing MDX/SQL

- To view the SQL or MDX for the entire Report Studio report, from the Tools menu, click Show Generated SQL/MDX. This option shows the SQL or MDX that will be executed in the data source. The SQL⁶ or MDX is organized by query and by query result. If a query is used in more than one data container, a query result is generated for each data container.
- To view the SQL or MDX for a specific query, (i) Pause the pointer over the query explorer button  and click the query and then - in the Properties pane, (ii) double-click the Generated SQL/MDX property. The Generated SQL/MDX property shows the SQL or MDX that is executed when you view tabular data (from the Run menu, click View Tabular Data). Tabular data shows the data that is produced by the query in the form of a list. You can use this property to help you build advanced queries.

A first performance optimization should focus on the generate MDX and how the report could be changed/written to generate a more performant MDX. Keep in mind that the performance of a report can often be improved by consolidating multiple sub-queries into fewer queries. If a report for example consists of 10 main sections (each section may be characterized by one or more sets), evaluate if the set expressions could be combined into one expression. Similarly, if a report were to use multiple, custom sets (and resulting in multiple queries for report assembly), an MDX set filter may be used instead to combine the custom sets into one set (and query). One could also resort to creating report-filter specific TM1 attribute(s) and leverage those for the MDX set filter.

For further optimization and particularly for large, performance-intensive reports that may spawn many MDX query requests against the TM1 database, MDX/SQL tracing may be needed:

4.3 Tracing MDX/SQL

4.3.1 Native SQL Tracing when Using Dynamic Query Mode

www.ibm.com/support/docview.wss?uid=swg21579065

4.3.2 Cognos Performance Monitoring and Analysis (PMA)

<https://ibm.box.com/s/8hvr4zsv7voia9entt2g6zsvx6avyjq>

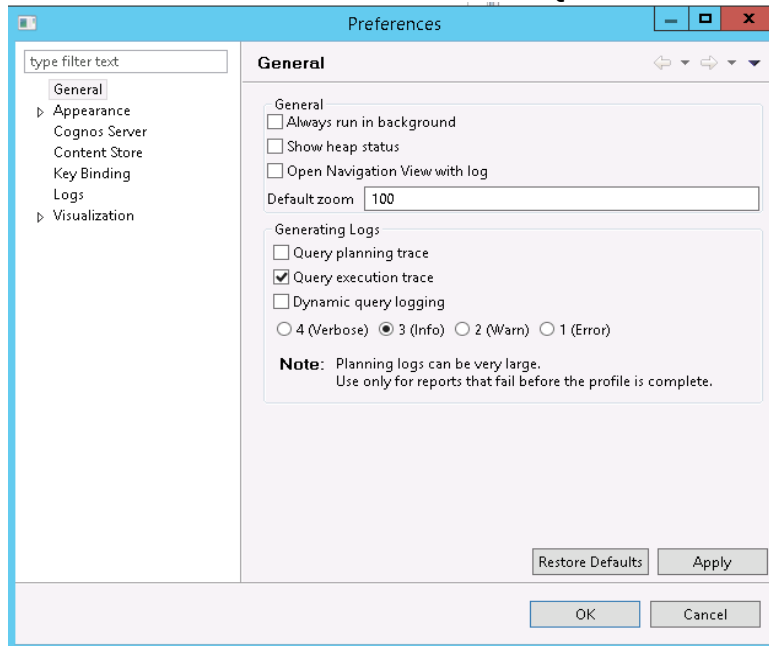
4.3.3 Whole-System Analysis of Idle Time (WAIT) Tool

<https://wait.ibm.com/>

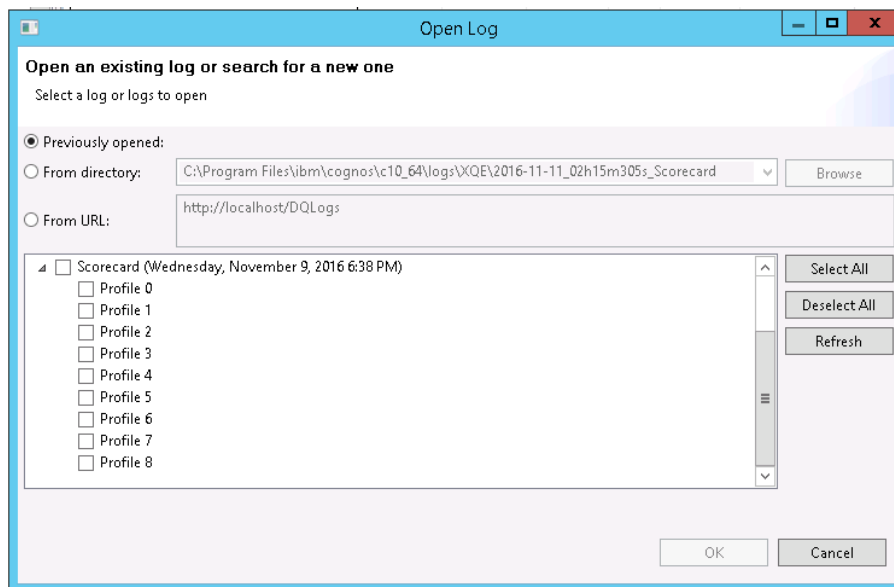
⁶ For SQL, you can choose to view native SQL, which is the SQL that is passed to the database when you execute the query, or Cognos® SQL, which is a generic form of SQL that IBM® Cognos Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

4.3.4 Using Dynamic Query Analyzer

To trouble-shoot issues with a specific report, you can use Dynamic Query Analyzer (DQA) to generate the logs, which captures the native MDX queries executed when the report is run as well as execution time and waiting time for query constructs. You'll need to install and configure DQA on your PC as per the [Dynamic Query Analyzer Installation and Configuration Guide](#). To generate the logs, you'll need to enable the Query Execution Trace in the Preferences window in DQA:



The query log files are generated in the <Cognos BI path>/logs/xqe/<time-stamp>_<report_name> folders:



For each MDX query, a profile log is generated. As you can see from the above screenshot, the Scorecard report generated 9 MDX queries.

The logs provide the execution time, wait time:

Summary

Name: Scorecard
Package: Smartco
Time stamp: Wednesday, November 9, 2016 6:38:28 PM
 OLAP (Online analytical processing)

Timing

Total Time: 3.468288 ms

Working Times:

Wait Times:

550 XMdxSelect (3.205488 ms)

673 XOLAPComputeCellOrdinalDecoration (0.060944 ms)

380 XOLAPEdgeMapping (0.059776 ms)

381 XOLAPEdgeMapping (0.047456 ms)

382 XOLAPEdgeMapping (0.045888 ms)

and the executed MDX queries. An example of one of the queries for Scorecard:

```

Query
WITH
  MEMBER
    [Version].[XQE_V5M_CM0] AS '
    (([Version].[Version].[@MEMBER].[DashboardCA]) - ([Version].[Version].[@MEMBER].[Prior Year Actual])),
    SOLVE_ORDER = 4,
    MEMBER_CAPTION = 'Actual - Prior Year Actual'
SELECT
  {[Version].[Version].[@MEMBER].[DashboardCA],
  [Version].[Version].[@MEMBER].[Variance],
  [Version].[Version].[@MEMBER].[Variance%],
  [Version].[XQE_V5M_CM0]}
  DIMENSION PROPERTIES PARENT_LEVEL,
  PARENT_UNIQUE_NAME
  ON AXIS(0),
  {[Account].[Account].[@MEMBER].[4999]}
  DIMENSION PROPERTIES PARENT_LEVEL,
  PARENT_UNIQUE_NAME
  ON AXIS(1),
  {[Currency Calc].[Currency Calc].[@MEMBER].[Local]}
  DIMENSION PROPERTIES PARENT_LEVEL,
  PARENT_UNIQUE_NAME
  ON AXIS(2)
FROM
  [Income Statement Reporting]
WHERE
  ([Year].[Year].[@MEMBER].[Y2],
  [Month].[Month].[@MEMBER].[Year],
  [Organization Reporting].[Organization Reporting].[@MEMBER].[Total Company])
CELL PROPERTIES CELL_ORDINAL,
  FORMAT_STRING,
  VALUE
  
```

Please refer to [Dynamic Query Analyzer User Guide](#) for details on how to analyse the logs.

4.4 BI Server Tuning and Monitoring

The recommended best practice for tuning and monitoring BI servers are also applicable for using TM1 cubes as data source for BI reports.

Please refer to [Tuning IBM Cognos BI Dispatchers](#) and [Tuning the Report Service, Batch Report Service, and Report Data Service](#) for guidelines on BI server tuning and [Performance Metrics](#) for guidelines on metrics to monitor on BI servers.

The dispatcher JVM heap size is indicative of how much memory is available for report processing. When the current heap size reaches 90% of the maximum JVM heap size, report requests may start to fail, and you may need to look into what's causing it and consider adjusting the heap size.

Metrics - <http://ibmdemo.demos.ibm.com:9300/p2pd>

0 0 0 No metric score

Name	Value
JVM	
Committed heap size (bytes)	-- 1,073,741,824
Current heap size (bytes)	-- 652,340,232
Initially requested heap size (bytes)	-- 1,073,741,824
Maximum heap size (bytes)	-- 1,073,741,824
Up time	-- 2 04:52:58.000

The high water marks for report queue and requests for query service, report service, batch service are also indicative whether there are performance issues with reports.

Queue - Report service	
Latency	00:00:00.000
Number of queue requests	0
Queue length	-- 0
Queue length high watermark	0
Queue length low watermark	0
Time in queue	00:00:00.000
Time in queue high watermark	00:00:00.000
Time in queue low watermark	00:00:00.000
Request - Report service	
Last response time	00:00:00.256
Number of failed requests	1
Number of processed requests	206
Number of successful requests	205
Percentage of failed requests	0.49%
Percentage of successful requests	99.51%
Response time high watermark	00:01:48.234
Response time low watermark	00:00:00.015
Seconds per successful request	00:00:01.099
Service time	00:03:45.508
Service time failed requests	00:00:00.047
Service time successful requests	00:03:45.461

5. Logging of Cognos BI queries in TM1

5.1 Cognos BI and Cognos Analytics: MDX Logging

Use MDX* and Rowset logging to temporarily debug/analyze Cognos BI queries against TM1, by adding the following rows to the tm1s-log.properties file.

```
log4j.logger.TM1.Mdx=DEBUG  
  
log4j.logger.TM1.Mdx.Interface=DEBUG  
  
log4j.logger.TM1.MdxViewCreate=DEBUG  
log4j.logger.TM1.Rowset=DEBUG  
log4j.logger.TM1.MdxLogger=DEBUG
```

5.2 Cognos Analytics: Rest API Logging

Use API loggers to temporarily debug and analyze TM1 (Rest) API communication between Cognos Analytics and TM1

```
log4j.logger.TM1.API=DEBUG  
  
log4j.logger.TM1.API.Parameters=DEBUG
```