Guideline

IBM Cookbook for IBM Cognos 8.4

for use with

SAP NetWeaver Business Warehouse™

Product(s): IBM Cognos 8.4

Area of Interest: Infrastructure

Copyright and Trademarks

Licensed Materials - Property of IBM.

© Copyright IBM Corp. 2009

IBM, the IBM logo, and Cognos are trademarks or registered trademarks of International Business Machines Corp., 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 http://www.ibm.com/legal/copytrade.shtml

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. IBM does not accept responsibility for any kind of loss resulting from the use of information contained in this document. The information contained in this document is subject to change without notice.

This document is maintained by the Best Practices, Product and Technology team. You can send comments, suggestions, and additions to cscogpp@ca.ibm.com.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

SAP, SAP NetWeaver, & SAP R/3 are registered trademarks of SAP AG in Germany and in several other countries.

Contents

1 Introduction

1.1 Purpose

This document is intended to provide a single point of reference for techniques and product behaviours when dealing with SAP Business Warehouse as a data source.

1.2 Applicability

The techniques and product behaviours outlined in this document apply to:

- IBM Cognos 8 version 8.4 27.28

1.3 Exclusions and Exceptions

The techniques and product behaviours outlined in this document may not be applicable to future releases.

2 SAP NetWeaver Business Warehouse Objects and BEx Queries

IBM Cognos 8 leverages the SAP Business Warehouse investment within your organization by providing access to the Business Warehouse metadata and business content. The integration with SAP Business Warehouse provides access to existing SAP Business Warehouse structures within the Business Warehouse environment:

- InfoQueries
- InfoCubes
- MultiProviders
- Virtual Cubes
- Remote Cubes
- DSO or ODS
- InfoSets
- InfoObjects
- Master data and attributes

The following section further outlines the SAP Business Warehouse objects that can be leveraged by IBM Cognos 8, and illustrates proven methods within SAP Business Warehouse that will help to ensure a successful implementation of IBM Cognos 8 to embrace, enhance and extend your investment in the SAP Business Warehouse.

2.1 Overall Guideline

When modelling against SAP Business Warehouse, there are some general guidelines that you should follow to ensure optimal performance within your IBM Cognos 8 environment with SAP Business Warehouse.

1. It is strongly recommended to use an InfoQuery as the source for the Framework Manager metadata model. The InfoQuery acts as a database view which can help reduce the volume of data for better performance. The use of the InfoQuery will provide additional access to query objects such as:

    - SAP Business Warehouse Variables
    - Filters, Restrictions
    - Calculated Key Figures
    - Restricted Key Figures
    - Characteristic and Key Figure Structures
    - Currency Conversions
    - Global Filters
    - Cell Definitions
    - Navigational Attributes

2. When using a SAP Business Warehouse InfoQuery, utilize SAP Business Warehouse variables, filters, and restrictions where appropriate. This will force data processing to be performed by the SAP Business Warehouse server database and will often provide better performance. It will also be beneficial to use Calculated Key Figures and Restricted Key Figures where possible to leverage the processing power of the SAP Business Warehouse server.

3. Carefully consider how many IBM Cognos 8 reports you decide to build from a SAP Business Warehouse InfoQuery. There should not be a 1-to-1, IBM Cognos 8 report-to-SAP Business Warehouse query ratio.
    - Consider having a SAP Business Warehouse query that sources several IBM Cognos 8 reports.
    - But do not create one master SAP Business Warehouse query that sources all of your IBM Cognos 8 reports.
    - Start by addressing a particular subject area, and then extend what you have learned in your environment to other subject areas. Apply this approach rather than starting big by implementing several SAP Business Warehouse queries (or one master SAP Business Warehouse query) for all subject areas, and potentially having to pull back due to poor performance.

4. Review the overall design of SAP Business Warehouse structures and data volumes to identify potential bottlenecks. A SAP Business Warehouse design that is inefficient in SAP Business Warehouse will filter up and deliver unsatisfactory performance to IBM Cognos 8 users. For example, it is commonly known that an InfoCube typically performs faster than a Data Store Object (DSO). In that case, it is suggested to use a SAP Business Warehouse query against a cube rather than against objects such as DSO's or InfoSets.

5. Use of a BEx InfoQuery is required to gain access to Data Store Objects (DSO), InfoSets, and Master Data as an InfoProvider within SAP Business Warehouse. Use this method sparingly if you are dealing with large data sets as you are likely to experience data latency from SAP

Business Warehouse. Reading data from these providers is typically slower than reading data from an InfoCube, and is considered normal SAP Business Warehouse performance amongst SAP Business Warehouse administrators. The same rule holds true when referencing these providers indirectly via a MultiProvider.

6. Furthermore, when utilizing a SAP Business Warehouse query against a MultiProvider, consider carefully the underlying InfoProviders in which data is read. Data that is read from a DSO, InfoSet, or InfoCube via a MultiProvider, with high volumes of data may inhibit SAP Business Warehouse performance, thereby passing this slow performance to IBM Cognos 8. Ensure that proper filters, variables, restrictions, and calculations are used within the SAP Business Warehouse query to force SAP Business Warehouse to pass only a subset of the data, instead of passing much more data with no filters and restrictions.

7. The use of the SAP Business Warehouse Accelerator appliance, BWA (formerly BIA), may be used in conjunction with IBM Cognos 8 for improved SAP Business Warehouse performance. These performance benefits can be realized from within the IBM Cognos 8 environment as the hardware appliance is completely transparent to IBM Cognos 8 and the user. However it is strongly advised to follow all the proven practices as noted within this document to ensure optimal performance for the overall IBM Cognos 8 and SAP Business Warehouse solution.

8. In some instances, you may find that you are creating an IBM Cognos 8 report that is reading high volumes of data from SAP Business Warehouse, which may overstep the limitations of the SAP Business Warehouse temp space. In the event SAP Business Warehouse cannot handle a data request from IBM Cognos 8, and all of these SAP Business Warehouse guidelines have already been applied, then consider adding more temp space to the SAP Business Warehouse server.

2.2 Definition of SAP NetWeaver Business Warehouse & BEx Terminology

The following table defines common SAP Business Warehouse objects that can be leveraged through IBM Cognos 8.

SAP Business Warehouse Object	Definition
InfoProvider	Analysis-relevant view of a SAP Business Warehouse object for which queries in SAP Business Warehouse can be created or executed. There are two types of InfoProviders. One type includes objects that contain physical data. These are known as data targets, such as InfoCubes, ODS objects, and InfoObjects (characteristics with attributes, texts, or hierarchies). The other type includes objects that display no physical data storage, such as InfoSets, RemoteCubes, and MultiProviders.
InfoCube	An InfoCube is an InfoProvider that describes a self-contained dataset. An InfoCube is a collection of relational tables that are created according to the star schema: a large fact table in the

center, with several dimension tables surrounding it.

MultiProviders

Type of InfoProvider that combines data from several InfoProviders and makes it available for reporting. The MultiProvider itself contains no data as its data comes exclusively from the InfoProviders on which it is based (collated using a union operation). A MultiProvider is assembled from different combinations of InfoProviders.

Virtual Cubes

InfoProvider with transaction data that is not stored in the object itself, but which is read directly for analysis and reporting purposes. The relevant data can be from the BI system or from other SAP or non - SAP systems.
VirtualProviders only allow read access to data.

Remote Cubes

Infocube that is managed in a remote system, but can be accessed in SAP Business Warehouse.

DSO or ODS

Data Store Objects or Operational Data Stores are objects that stores consolidated and cleansed transaction data on a detail level.

InfoSets

A semantic view of ODS objects and InfoObjects (characteristics with master data) that allows you to create reports on these objects, particularly on the joins between these objects.

InfoObjects

Business evaluation objects (for example, customers or sales) are called InfoObjects in SAP Business Warehouse. InfoObjects are subdivided into characteristics, key figures, units, time characteristics, and technical characteristics (such as request numbers).

Master Data

Static data (data that rarely changes) describing characteristics. An example may be an employee's birth date, home address, social security number, etc.

Aggregates

Subsets of data from an InfoCube. Aggregates make it possible to access InfoCube data quickly in Reporting. Aggregates serve to improve performance much like data base indexes.

InfoQuery

A reporting structure in SAP Business Warehouse that serves as a request for information from SAP Business Warehouse InfoProviders.

Key Figures

Type of InfoObject. Numeric fields; like number of items on hand, account credit amount & account debit amount.

Characteristic

A type of InfoObject that is a non-numeric field such as company code, account number, course, birth date, etc.

Dimension

The grouping of characteristics whose content is logically connected, under a single generic term, such as Organization, Region, Customers, Products, etc. Dimensions are defined to enable characteristics to be stored in a star schema table (dimension table).

Hierarchy

The organization of the values of a characteristic into a tree structure.

Characteristic Level

A characteristic that acts as a level within a hierarchy.

Attributes

Data which is not stored in an InfoCube, but in master data tables.

Navigational Attributes
InfoObjects that are logically assigned attributes which you can filter or drill on in the query.

Display Attributes
Used for display only as this kind of attribute is subordinated to a characteristic and cannot be filtered or drilled on in a query.

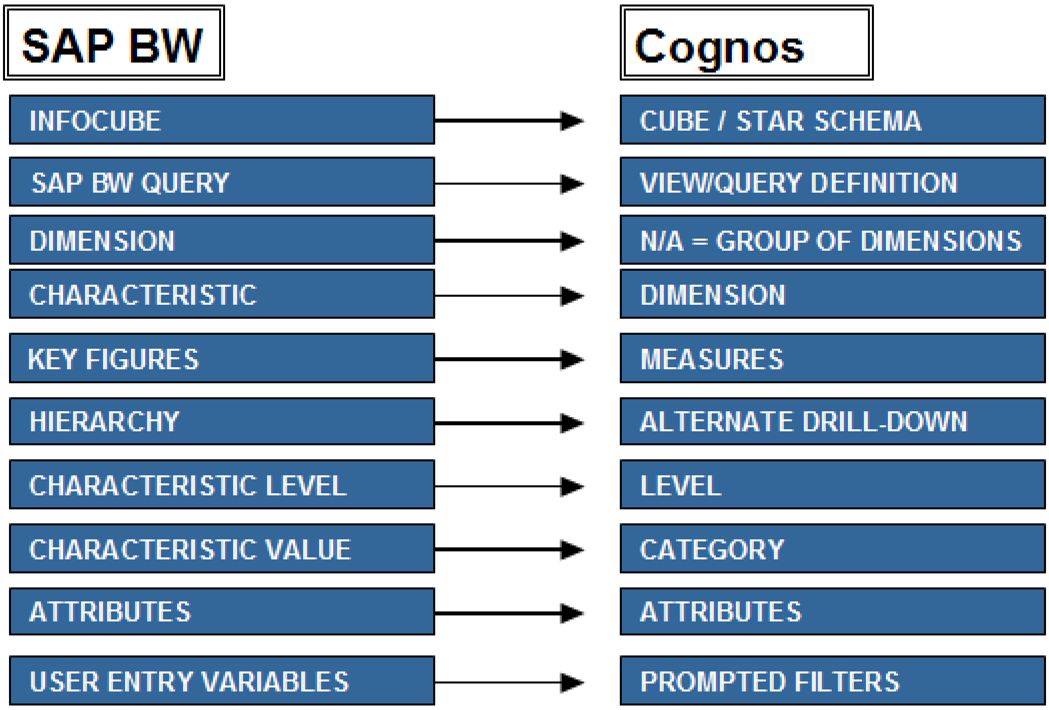
Filter
A filter has a limiting effect on the data selected. When defining the filter, you select characteristic values from one or more characteristics or from a key figure. All of the InfoProvider data is restricted using the filter selection of the query.

Free Characteristic
Characteristics not displayed on the initial report, but available to be included in the report. Free Characteristics have drill-down or drill-across capabilities.

Calculated Key Figure
Consist of formula definitions containing basic key figures, restricted key figures or pre calculated key figures.

Restricted Key Figure
You can restrict the key figures of an InfoProvider by selecting one or more characteristics. The key figures that are restricted by one or more characteristic selections by selecting a basic key figure, calculated key figure or by other restricted key figures.

The following outlines some of these common SAP Business Warehouse and BEx terms, and maps them to common IBM Cognos 8 terms.



SAP BW	Cognos
INFOCUBE	CUBE / STAR SCHEMA
SAP BW QUERY	VIEW/QUERY DEFINITION
DIMENSION	N/A = GROUP OF DIMENSIONS
CHARACTERISTIC	DIMENSION
KEY FIGURES	MEASURES
HIERARCHY	ALTERNATE DRILL-DOWN
CHARACTERISTIC LEVEL	LEVEL
CHARACTERISTIC VALUE	CATEGORY
ATTRIBUTES	ATTRIBUTES
USER ENTRY VARIABLES	PROMPTED FILTERS

2.3 InfoQuery Naming Convention

For easy maintenance it is recommended that InfoQueries which are to be used with IBM Cognos 8 adhere to the following naming convention. Note that this is only a suggested approach, and may be superseded by your existing naming conventions where differences exist.

1. Initial letter Z followed by the InfoProvider name. For example, *ZCPA_M04* denotes custom InfoQuery for the InfoProvider CPA_M04

2. Following the InfoProvider Name should be Q and a sequence of numbers:

   ZCPA_M04_Q0001 , where the Q denotes Query, and 0001 is the query number

3. All InfoQueries developed for use with IBM Cognos 8 should then be followed with the letters COG:

   ZCPA_M04_Q0001_COG

4. Finally, all InfoQueries should end with the version number:

   ZCPA_M04_Q0001_COG_001

2.4 When Creating an InfoQuery for IBM Cognos 8

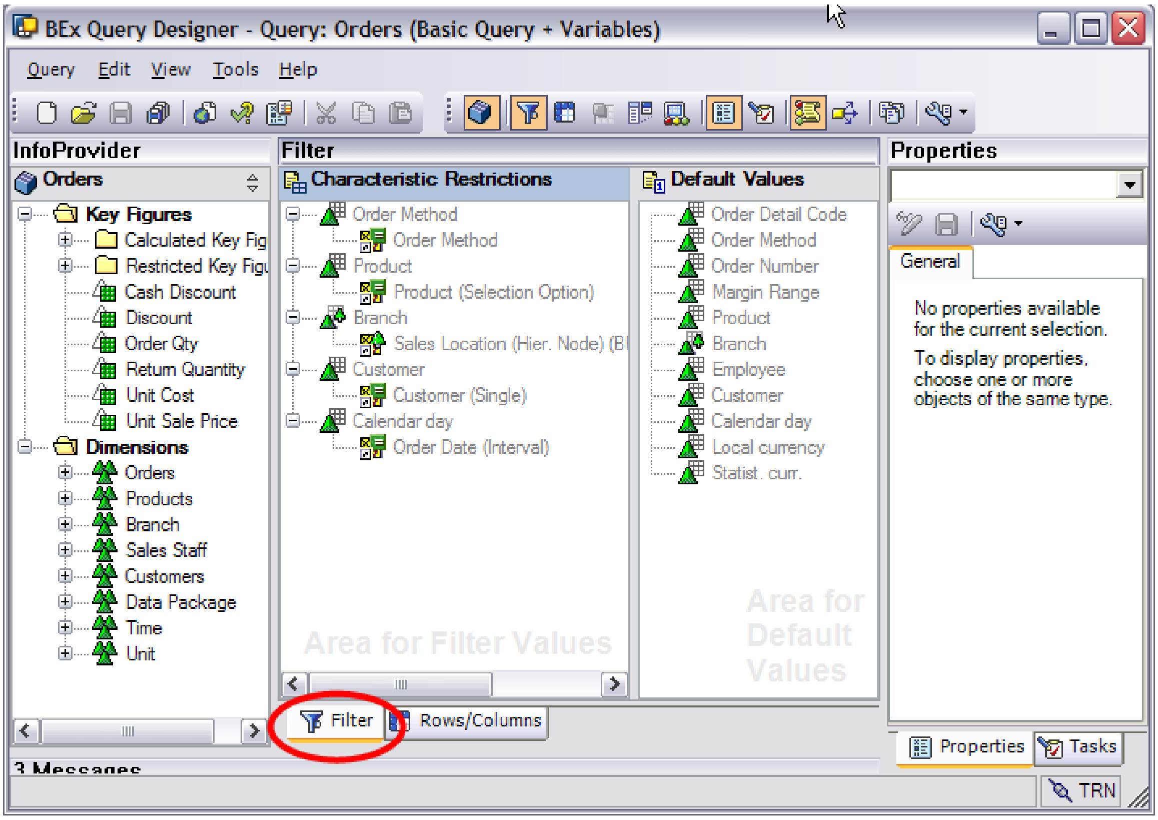
The following points are general guidelines, tips & techniques which should first be considered before developing an InfoQuery in the BEx Query Designer that is built specifically for IBM Cognos 8 use.

1. Start small. Do not create a 'master query' that brings in all data from a cube or multiple cubes.

2. Instead restrict the data set to a certain subject area, and apply filters, restrictions, and variables which make sense for the users. Use of calculate or restricted key figures is encouraged as well. This approach will enable you to develop many IBM Cognos 8 reports from one InfoQuery, but will ensure much of the data processing will be performed by the SAP Business Warehouse server.

3. Use of existing or new SAP Business Warehouse variables is highly encouraged. To help ease the management of the variables in IBM Cognos 8, try to keep the technical name of the variable to a maximum of eight characters. This will ensure the proper technical name is referenced in IBM Cognos 8 each time the variable is re-imported.

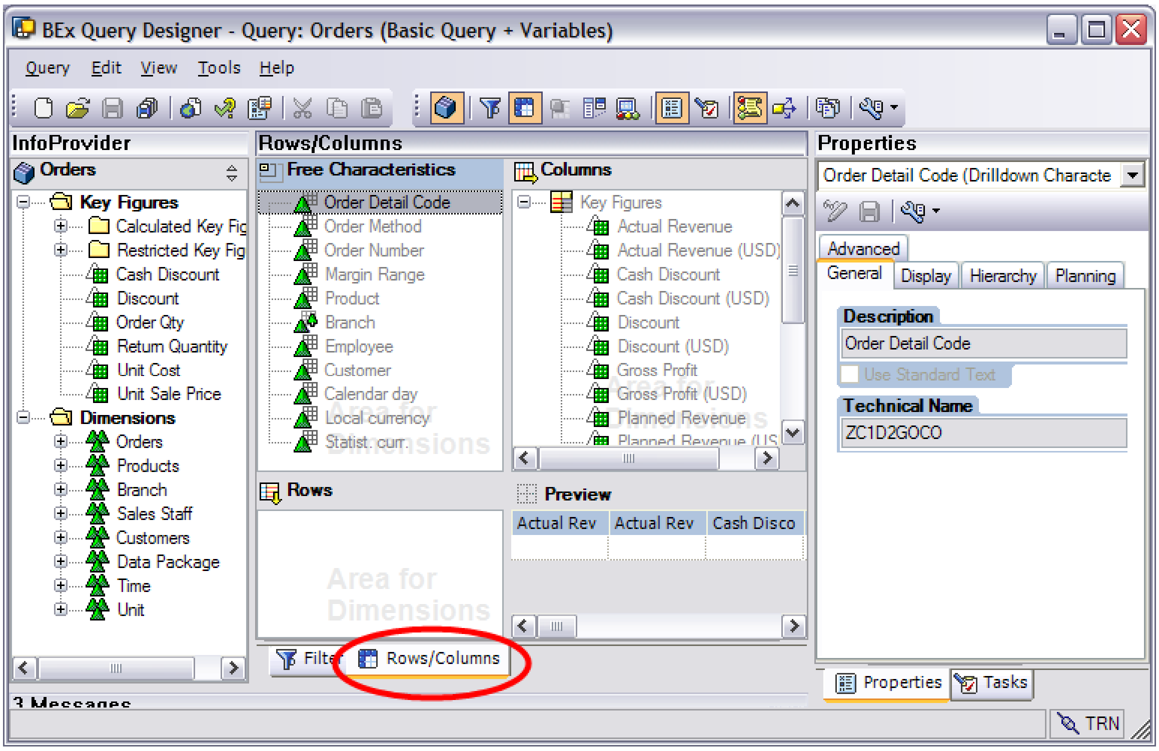
4. For the 'filter' tab in Query Designer, place the filters, restrictions, or variables in the 'Characteristic Restrictions' window rather than the 'Default Values' window. This will ensure all restrictions are captured within IBM Cognos 8.



No properties available for the current selection.

To display properties, choose one or more objects of the same type.

5. Note that all objects placed into the ‘Rows & Columns’ tab will be read into IBM Cognos 8, even Free Characteristics. You will notice from the screenshot below that rows are not included in this query, which is perfectly fine. All of the metadata is captured by IBM Cognos 8 from the Key Figures columns and the Free Characteristics.



Description: Order Detail Code

Technical Name: ZC1D2GOCO

2.4.1 Releasing an InfoQuery for IBM Cognos 8

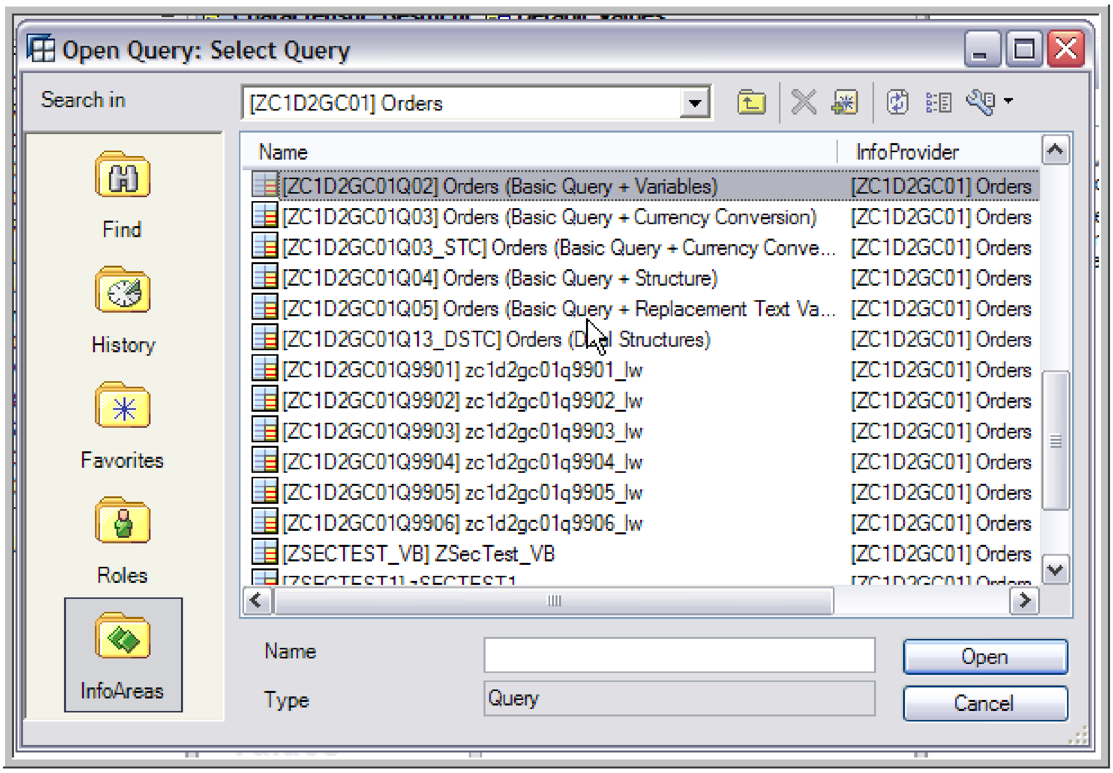
In order for an InfoQuery to be visible to IBM Cognos 8 Framework Manager for import, it should be released for third party applications. This is done within the Business Explorer Query Designer.

1. Launch the Query Designer from the All Programs\Business Explorer\Query Designer.

2. Choose the SAP Business Warehouse system which contains the InfoQuery that needs to be released.

3. Logon providing a valid user ID, password and client number.

4. Locate the desired InfoQuery that needs to be released and press OK.



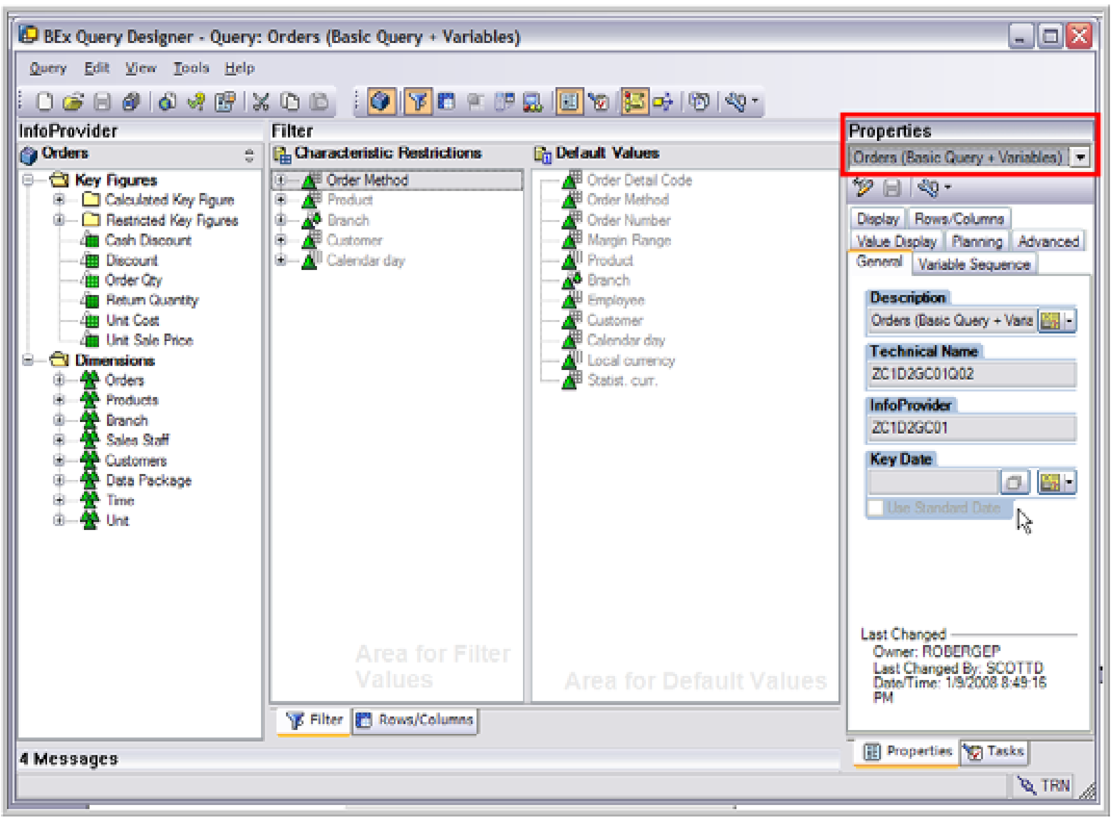
Search in: [ZC1D2GC01] Orders

Name	InfoProvider
[ZC1D2GC01Q02] Orders (Basic Query + Variables)	[ZC1D2GC01] Orders
[ZC1D2GC01Q03] Orders (Basic Query + Currency Conversion)	[ZC1D2GC01] Orders
[ZC1D2GC01Q03_STC] Orders (Basic Query + Currency Conve...	[ZC1D2GC01] Orders
[ZC1D2GC01Q04] Orders (Basic Query + Structure)	[ZC1D2GC01] Orders
[ZC1D2GC01Q05] Orders (Basic Query + Replacement Text Va...	[ZC1D2GC01] Orders
[ZC1D2GC01Q13_DSTC] Orders (Dual Structures)	[ZC1D2GC01] Orders
[ZC1D2GC01Q9901] zc1d2gc01q9901_lw	[ZC1D2GC01] Orders
[ZC1D2GC01Q9902] zc1d2gc01q9902_lw	[ZC1D2GC01] Orders
[ZC1D2GC01Q9903] zc1d2gc01q9903_lw	[ZC1D2GC01] Orders
[ZC1D2GC01Q9904] zc1d2gc01q9904_lw	[ZC1D2GC01] Orders
[ZC1D2GC01Q9905] zc1d2gc01q9905_lw	[ZC1D2GC01] Orders
[ZC1D2GC01Q9906] zc1d2gc01q9906_lw	[ZC1D2GC01] Orders
[ZSECTEST_VB] ZSecTest_VB	[ZC1D2GC01] Orders
[ZSECTEST1] ZSECTEST1	[ZC1D2GC01] Orders

Name:

Type: Query

5. Once the query opens, select the Properties of the InfoQuery from the drop-down menu in the properties pane located to the right of the screen.



Description: Orders (Basic Query + Varia

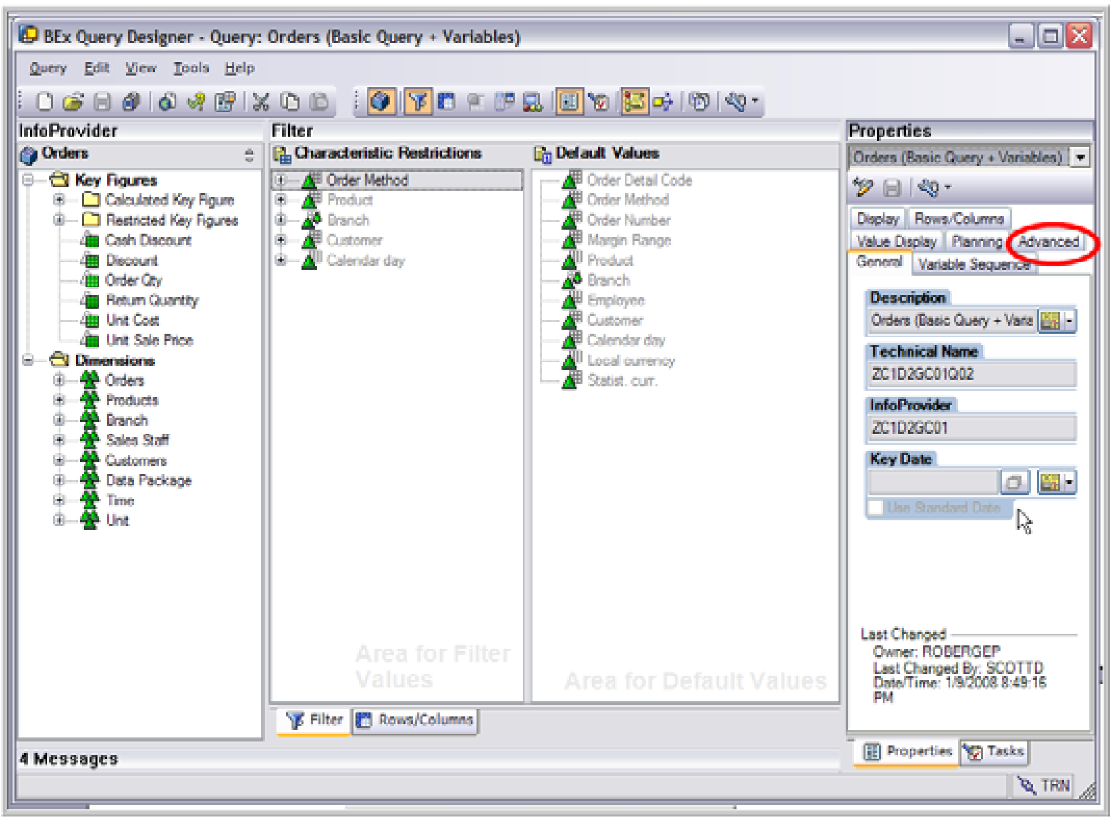
Technical Name: ZC1D2GC01Q02

InfoProvider: ZC1D2GC01

Key Date:

Last Changed
Owner: ROBERGEP
Last Changed By: SCOTTD
Date/Time: 1/9/2008 8:49:16 PM

6. Within the Query Properties, select the Advanced Tab



Description: Orders (Basic Query + Varia

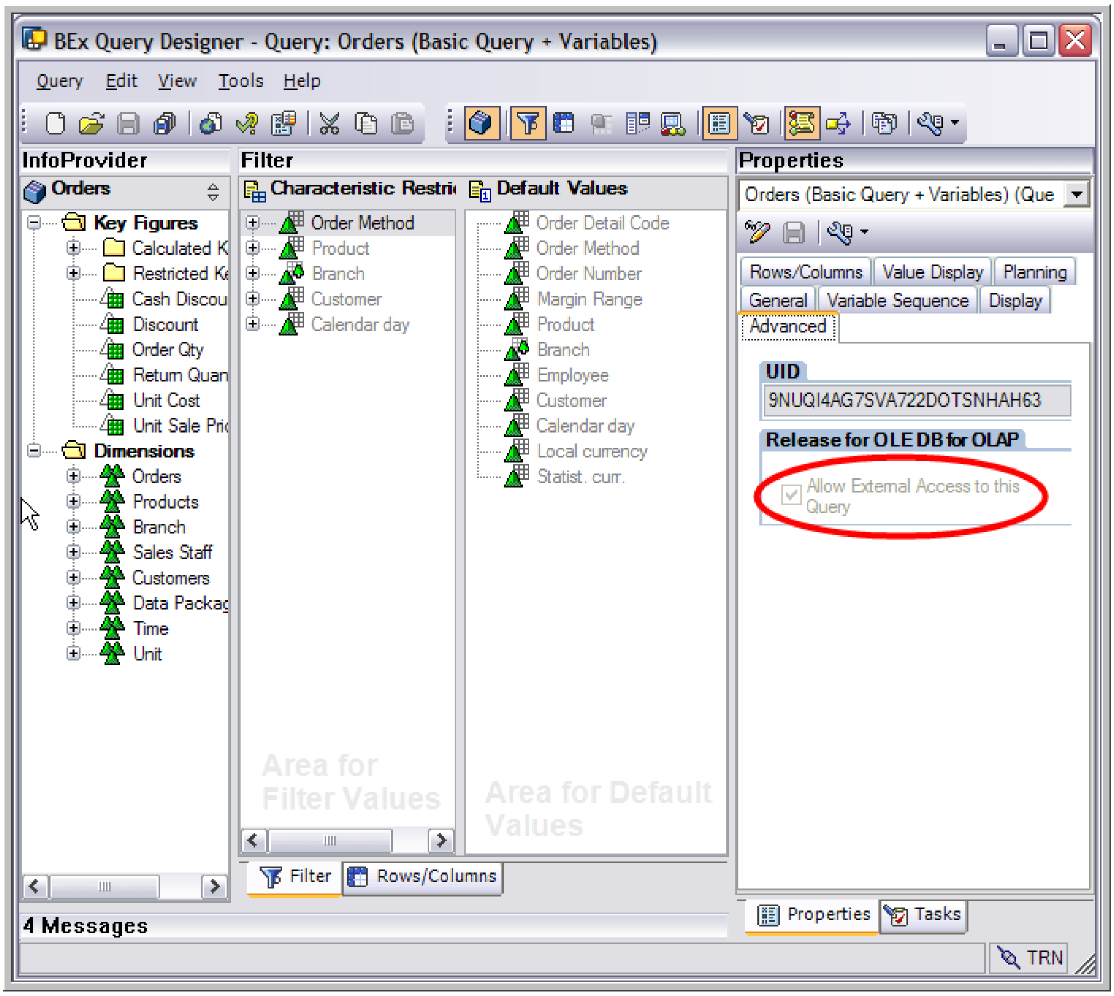
Technical Name: ZC1D2GC01Q02

InfoProvider: ZC1D2GC01

Key Date:

Last Changed
Owner: ROBERGEP
Last Changed By: SCOTTD
Date/Time: 1/9/2008 8:49:16 PM

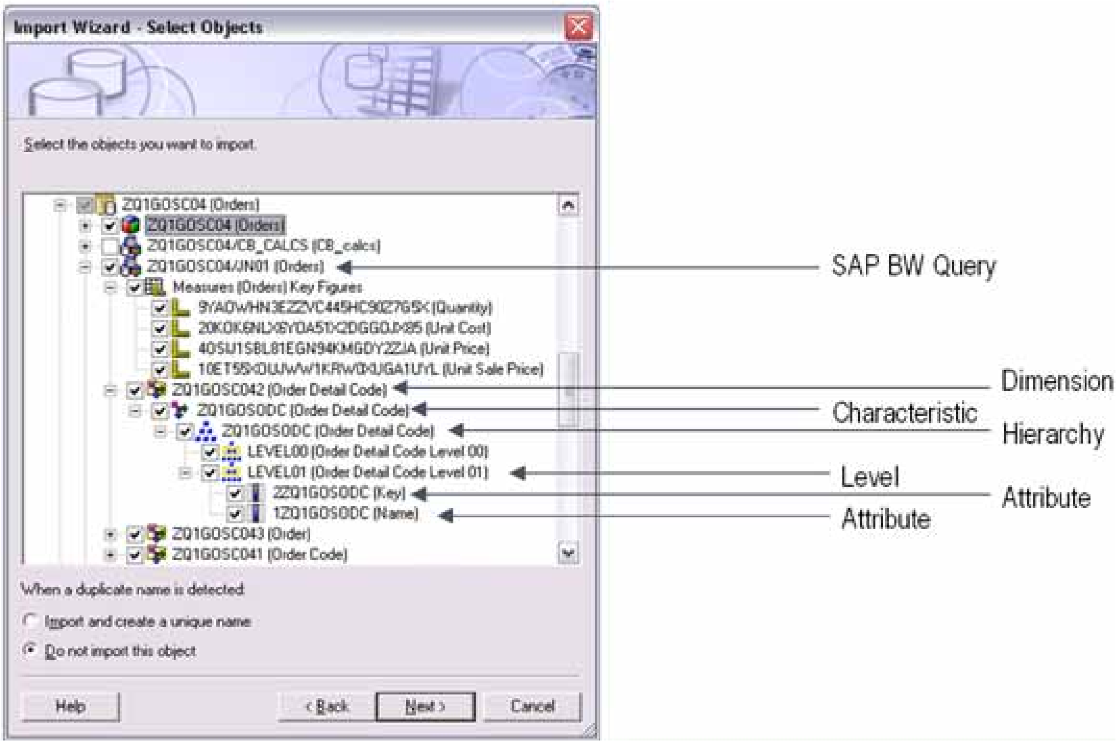
7. Check the “Allow External Access to this Query” check box and select OK



UID
9NUQI4AG7SVA722DOTSNHAH63

Release for OLE DB for OLAP
☑ Allow External Access to this Query

8. The InfoQuery should now be visible to IBM Cognos 8 Framework Manager (FM) for import. The screenshot below shows the FM Import Wizard, and the individual elements within a SAP Business Warehouse query.



Import Wizard - Select Objects

Select the objects you want to import.

☑ ZQ1GOSC04 (Orders)
☑ ZQ1GOSC04 (Orders)
□ ZQ1GOSC04/CB_CALCS (CB_calcs)
☑ ZQ1GOSC04/JN01 (Orders)
☑ Measures (Orders) Key Figures
☑ 9YAOWHN3EZZVC445HC9027G5K (Quantity)
☑ 20K0K6NLX8YDA51X2DGG0JX85 (Unit Cost)
☑ 40SU1SBL81EGN94KMGDY2ZJA (Unit Price)
☑ 10ET55KOUJWW1KRW0KUGA1UYL (Unit Sale Price)
☑ ZQ1GOSC042 (Order Detail Code)
☑ ZQ1GOSODC (Order Detail Code)
☑ ZQ1GOSODC (Order Detail Code)
☑ LEVEL00 (Order Detail Code Level 00)
☑ LEVEL01 (Order Detail Code Level 01)
☑ 2ZQ1GOSODC (Key)
☑ 1ZQ1GOSODC (Name)
☑ ZQ1GOSC043 (Order)
☑ ZQ1GOSC041 (Order Code)

When a duplicate name is detected

○ Import and create a unique name

● Do not import this object

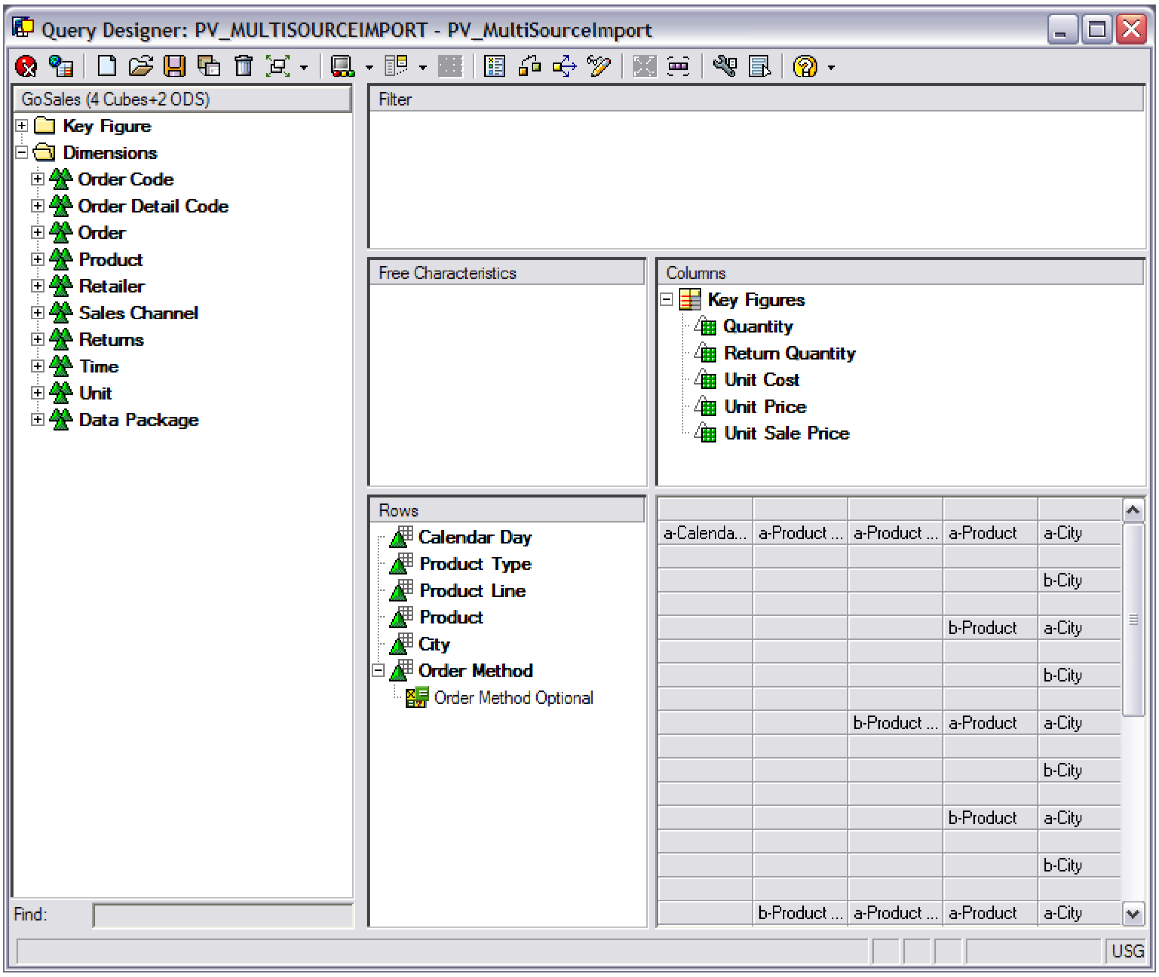
Due to the way that metadata is extracted and stored in the IBM Cognos 8 Framework Manager model, it will be necessary to include IBM Cognos 8 Framework Manager in your SAP Business Warehouse change management plans. The following are common scenarios where changes in SAP Business Warehouse do not usually alter the metadata and do not require an update of the IBM Cognos 8 Framework Manager model:

- **Filter**: You may change an existing filter inside the InfoQuery, or create a new static filter and the data will be filtered based on the filter criteria. Note this does not apply to creation of a new user entry variable.
- **Restrictions**: You can apply a new restriction on the filter tab, and the restriction will be applied without a re-import so long as the characteristic was already part of the InfoQuery.
- **Calculated & Restricted Key Figures**: You can change the formula/restriction inside an existing Calculated or Restricted Key figures, and the new formula will show up inside of IBM Cognos 8. Note you cannot delete the Calculated Key Figure and re-create using the same name and expect the reports to run. The Key Figures are referenced by the technical name.

2.5 Considerations for BEx queries intended for IBM Cognos 8 PowerPlay Tranformer

In the event you intend to read data from SAP Business Warehouse into an IBM Cognos 8 PowerPlay cube, you may utilize IBM Cognos 8 Transformer with SAP Business Warehouse InfoQueries. The following provides the proven methods to build a SAP Business Warehouse InfoQuery within SAP's Query Designer.

1. Create a SAP Business Warehouse InfoQuery with only the dimensions you wish to import. The more dimensions you include the more detail records you will get back. When IBM Cognos 8 extracts factual data from the INFOQUERY, a “select * from cube\INFOQUERY” approach is used for performance considerations. Therefore only include the dimensions you wish to utilize in transformer. Each dimensions added will grow the number of detailed records being returned – so if you do not need a particular dimension, do not include it.



Query Designer: PV_MULTISOURCEIMPORT - PV_MultiSourceImport

GoSales (4 Cubes+2 ODS)

- Key Figure
- Dimensions
  - Order Code
  - Order Detail Code
  - Order
  - Product
  - Retailer
  - Sales Channel
  - Returns
  - Time
  - Unit
  - Data Package

Find:

Filter

Free Characteristics

Columns

- Key Figures
  - Quantity
  - Return Quantity
  - Unit Cost
  - Unit Price
  - Unit Sale Price

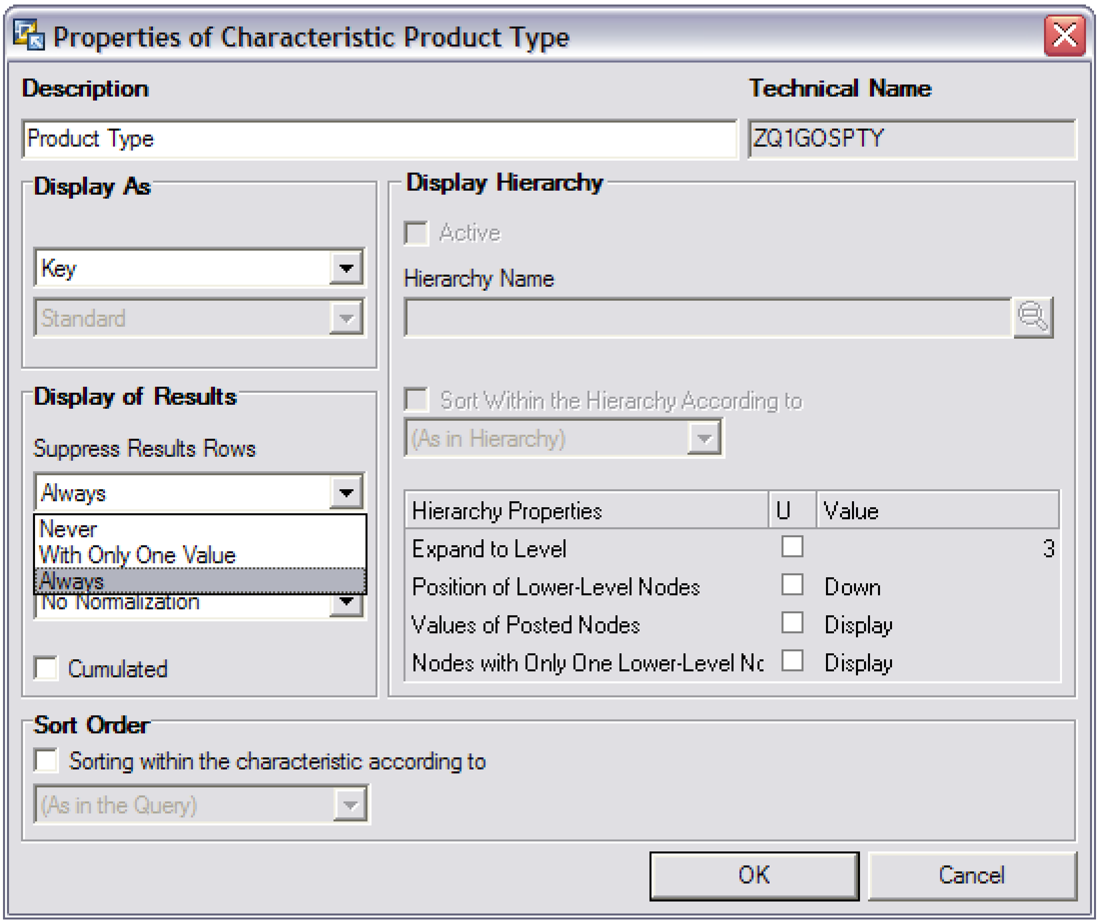
Rows

- Calendar Day
- Product Type
- Product Line
- Product
- City
- Order Method
  - Order Method Optional

a-Calenda...	a-Product ...	a-Product ...	a-Product	a-City
				b-City
			b-Product	a-City
				b-City
		b-Product ...	a-Product	a-City
				b-City
			b-Product	a-City
				b-City
	b-Product ...	a-Product ...	a-Product	a-City

USG

2. With SAP’s Query Designer, remember to set “Supress Results Rows” to “Always” and “Display As” to “Key” for each characteristic as per the documentation.



Properties of Characteristic Product Type

Description	Technical Name
Product Type	ZQ1GOSPTY

Display As

Key

Standard

Display of Results

Suppress Results Rows

Always

Never
With Only One Value
Always
No Normalization

□ Cumulated

Display Hierarchy

□ Active

Hierarchy Name

□ Sort Within the Hierarchy According to

(As in Hierarchy)

Hierarchy Properties	U	Value
Expand to Level	□	3
Position of Lower-Level Nodes	□	Down
Values of Posted Nodes	□	Display
Nodes with Only One Lower-Level Nc	□	Display

Sort Order

□ Sorting within the characteristic according to

(As in the Query)

3. Make sure to create an **optional** variable on one characteristic. The optimal characteristic to use is one that slices the data evenly across all its unique members. Otherwise, in the absence of an optional variable, IBM Cognos 8 will attempt to extract all the data with one query utilizing the regular reporting approach to SAP Business Warehouse. While very flexible approach for reporting, for extractions it will not allow you to retrieve more then 1 million cells due to SAP Business Warehouse limitations, and will perform poorly.

For performance considerations, the IBM Cognos 8 process to extract fact data will initially make a function call to retrieve all possible values for this created variable. It will then send one MDX in the following format:

Select * from [*CUBE/INFOQUERY*]
SAP VARIABLES [*VAR_NAME*] INCLUDING [*CHARAC*].[*VALUE*]

For each variable value received, IBM Cognos 8 will send down an MDX statement passing the variable values.

For example, if an optional variable was defined on [PROFIT_CTR] and 100 distinct profit centers exist are returned through a function call, IBM Cognos 8 will sequentially send 100 MDX statements, one for each profit center.

This approach both greatly improves performance as well as allowing the IBM Cognos 8 engine to retrieve detailed records beyond the 1 million cell limit imposed by SAP Business Warehouse in most environments.

To learn more about reading data from SAP Business Warehouse into a IBM Cognos 8 PowerPlay cube, you may reference the IBM Cognos 8 Framework Manager and IBM Cognos 8 Transformer sections within this document.

The IBM Cognos 8 Transformer User Guide also contains useful information regarding the extraction of data from SAP Business Warehouse using Transformer. These details can be found within section Appendix E of the help document entitled *Guidelines for Working with SAP BW Data for Use in Transformer*. This guide can be found within the IBM Cognos 8 home directory, c8\webcontent\documentation\.

2.6 Variables within BEx queries

When developing a SAP Business Warehouse InfoQuery for IBM Cognos 8, SAP Business Warehouse optional or mandatory variables can be used to restrict data where necessary. The following outlines the SAP Business Warehouse variable types, processing types, and variable functions which can be leveraged in IBM Cognos 8.

SAP NetWeaver Business Warehouse Variable Types, Processing Types & Variable Functions

Characteristic Value	Characteristic value variables represent characteristic values and can be used wherever characteristic values are used. These can be utilized to filter data based on individual, multiple, or a range of characteristic values
Text	Text represent a text and can be used in descriptions of queries, calculated key figures and structural components
Hierarchy Node	Hierarchy node variables represent a node in a hierarchy and can be used wherever hierarchy nodes are used. Use these to prompt users to select a node within a SAP Business Warehouse hierarchy.
Formula	Formula variables represent numerical values and can be used in formulas. These may be used to perform calculations within the query for key figures, or as an example could be used to perform unit conversions, currency translations, etc.
SAP User Exits	Gives you the option of setting up a processing type for variables, tailor-made to your specific needs. These exits are preconceived enhancement (routines) from SAP that you can configure with customer-specific logic without affecting the standard SAP programs.
Replacement Path	Characteristic value variables, hierarchy variables, text variables, and formula variables with the *Replacement Path* processing type can be replaced by the value from a different variable. The value is either determined from the key, the external attribute of the key, the description, or the attribute value. You can also define an offset start and offset length.
Authorization	If within characteristic or hierarchy node variables you choose

	Process with Authorization, the variable is automatically filled with the values of the user's authorization. When the user opens a query, the data is selected automatically according to the user’s authorizations.
Single Value	A variable that represents one value only.
Multiple Single Values	Variable represents a number of single values. This setting is useful in hierarchy nodes, for example, to allow you to select multiple single nodes.
Intervals	Variable represents a specific ‘from’ value and a specific ‘to’ value, thus an interval.
Optional	The variable does not have to be filled with a value at runtime.
Mandatory	At least one value has to be specified for the variable at runtime. The initial value (#) is permitted explicitly. The initial value # means “unassigned”, that is, you can use it to explicitly select all data records in which this characteristic has no instances.
Variable Offsets	Provide the ability to dynamically show data for a time interval by specifying only a single input value within a variable.

Although most SAP Business Warehouse variables integrate with IBM Cognos 8 out of the box, there are some exceptions to the rule. The following describes the exceptions accompanied with alternate solutions you may deploy within IBM Cognos 8 to meet the functionality desired.

Hierarchy Selection Variable: Hierarchy selection variables represent a set of hierarchies and are used in SAP Business Warehouse wherever hierarchies should be selected. These are used to prompt the user to select a hierarchy from a list of multiple hierarchies.

Alternate Solution: Rather than utilizing a hierarchy selection variable within the SAP Business Warehouse InfoQuery, build an IBM Cognos 8 prompt in Report Studio that will enable a user to select from a list of SAP Business Warehouse hierarchies.

Selection Option: This SAP Business Warehouse variable represents any combination of single values and intervals. In addition, when you select values for variables, you can use operators (>,<,=, and so on), use the contains pattern option (for example, search for all values that begin with A: A*) and exclude certain values (by specifying the values for which you do not want to search).

Alternate Solution: This SAP Business Warehouse variable is read into IBM Cognos 8 as a single select drop-down prompt. If you desire similar functionality in IBM Cognos 8 as a Selection Option variable, consider using a Search & Select prompt or a combination of IBM Cognos 8 prompts in Report Studio which will provide the appropriate prompt functions desired for the user. Further information regarding this method can be found within Section 5-IBM Cognos 8 Report Studio.

2.7 Considerations for BEx Queries that Draw Large Volumes of Data

There are several SAP Business Warehouse design and query concepts you should consider when using BEx queries that read large volumes of data from theBusiness Warehouse. The following statements identify many of those SAP Business Warehouse concepts. Many of these recommendations are basic SAP Business Warehouse and data warehousing concepts which will ensure the SAP Business Warehouse system works for you and not against you. In the end, following these steps will provide a more stable and better performing Business Intelligence solution.

1. Adhere to SAP NetWeaver Business Warehouse proven practices for InfoProvider design. A solid design of an InfoProvider will provide for much more consistent query and report performance.

2. If you have a choice of building a query from an InfoCube or MultiProvider, always choose the InfoCube. Building a query from a MultiProvider will result in SAP Business Warehouse creating unions in order to satisfy the query.

3. Similarly, if you have a choice of building a query from an InfoCube or an ODS/DSO and/or InfoSet, always choose the InfoCube. Typical ODS or DSOs have detailed levels of data, and are not optimized for reporting purposes. And building a query from an InfoSet will require additional table joins from DSOs and InfoObjects in order to satisfy the query.

4. If using a query from a MultiProvider, limit as much as possible the number of InfoProviders from which data is read. So for example, if the MultiProvider consists of 5 InfoCubes, attempt to restrict data-reading to one cube rather than reading data from all or most of those cubes. This can be accomplished by applying filters and restrictions on the query within the SAP Query Designer

5. Avoid using remote InfoCubes except in the case where there is a small amount of data and a limited number of users. These generally do not perform as well as your basic cube.

6. Minimize the amount of information returned by the query. The less information returned from the query, generally the faster it can be transferred to the end user.

7. Design queries to prevent a large number of rows from being returned. This may be accomplished by using mandatory (required) variables to filter data or create exception reports to return only the data of interest.

8. Avoid complex queries that offer everything for every user. Create multiple queries to meet the requirements and/or use the IBM Cognos 8 drill-through capabilities to link reports together at different levels of detail.

9. If hierarchies are to be used, limit the number of nodes in the query results by using filters and variables for the hierarchy. The objective should be to make the entry hierarchy level as deep as possible upon executing the report. This limits the levels of the hierarchy that must be processed at execution time.

10. Use filters & variables as much as possible, especially for large characteristics, to reduce the size of the query result.

11. Apply the zero suppression property within the SAP Business Warehouse InfoQuery. This method can reduce the number of records which are sent from SAP Business Warehouse to the IBM Cognos 8 environment in the event many zero values exist within the SAP Business Warehouse InfoProvider for the combination of dimensions and key figures requested.

12. Apply general system performance techniques where applicable such as enabling SAP Business Warehouse caching, implementing database indexes, build InfoCube aggregates, or load data to a SAP Business Warehouse Accelerator appliance. These are all performance techniques that will be leveraged by the IBM Cognos 8 engine.

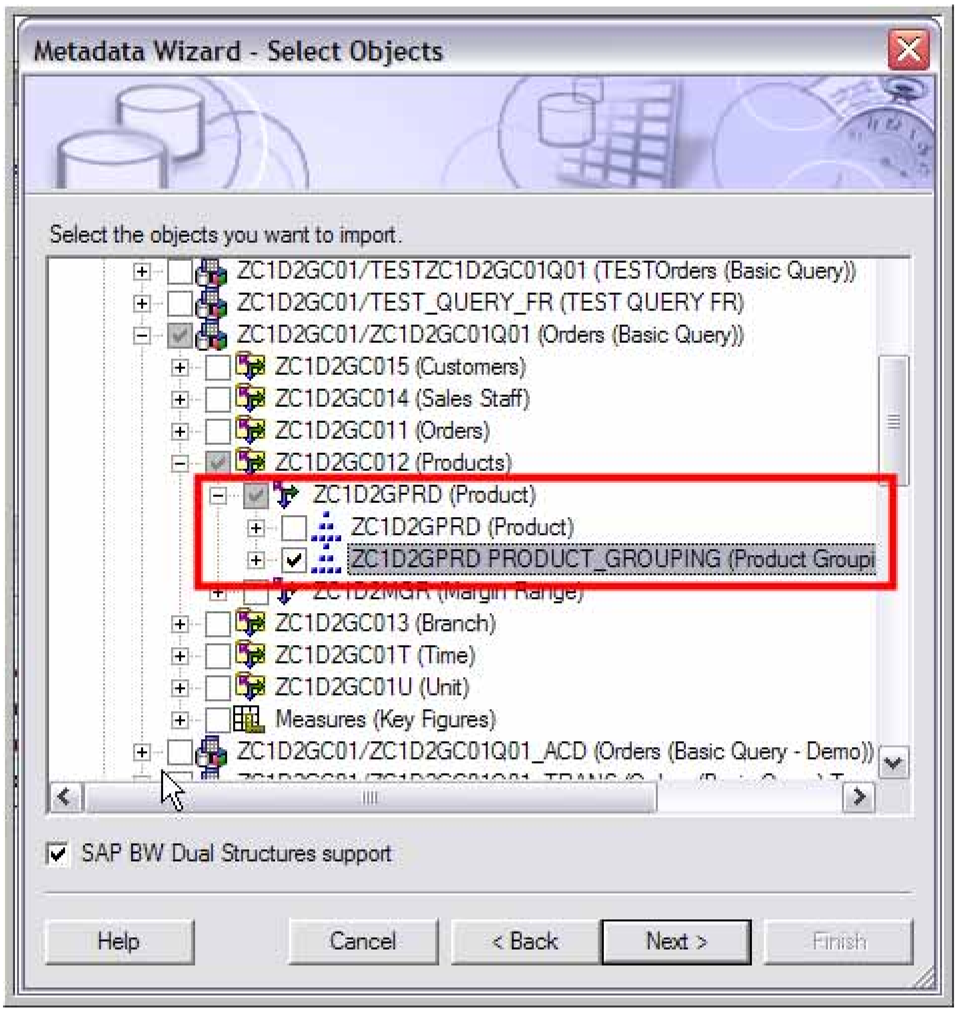
13. In the event all of these concepts are applied and better performance is desired, other options may include reading SAP Business Warehouse data into a high-performance caching IBM Cognos 8PowerPlay cube via IBM Cognos 8 Transformer, or load data to a IBM Cognos 8 TM1 cube which uses in-memory technology.

2.8 BEx Queries using Hierarchies

SAP Business Warehouse hierarchies are the organization of the characteristic values of a characteristic into a tree structure. They are typically created in SAP’s transactional R/3 or ECC system and then loaded into SAP Business Warehouse, or built directly within SAP Business Warehouse for a characteristic. Hierarchies can be used across InfoProviders, and are used to provide an organized view for reporting needs.

2.8.1 SAP Business Warehouse Hierarchies in IBM Cognos 8

All hierarchies that are loaded to, or built in SAP Business Warehouse for an individual characteristic are easily made available in your IBM Cognos 8 environment by importing metadata from SAP Business Warehouse via IBM Cognos 8 Framework Manager. You have the ability include all alternate hierarchies for a characteristic, whether there is only one hierarchy or several. Finally, you have the ability to omit specific hierarchies of a characteristic within the Metadata Wizard via IBM Cognos 8 Framework Manager.



Metadata Wizard - Select Objects

Select the objects you want to import.

- ☐ ZC1D2GC01/TESTZC1D2GC01Q01 (TESTOrders (Basic Query))
- ☐ ZC1D2GC01/TEST_QUERY_FR (TEST QUERY FR)
- ☑ ZC1D2GC01/ZC1D2GC01Q01 (Orders (Basic Query))
  - ☐ ZC1D2GC015 (Customers)
  - ☐ ZC1D2GC014 (Sales Staff)
  - ☐ ZC1D2GC011 (Orders)
  - ☑ ZC1D2GC012 (Products)
    - ☑ ZC1D2GPRD (Product)
      - ☐ ZC1D2GPRD (Product)
      - ☑ ZC1D2GPRD PRODUCT_GROUPING (Product Groupi
    - ☐ ZC1D2MGR (Margin Range)
  - ☐ ZC1D2GC013 (Branch)
  - ☐ ZC1D2GC01T (Time)
  - ☐ ZC1D2GC01U (Unit)
  - ☐ Measures (Key Figures)
- ☐ ZC1D2GC01/ZC1D2GC01Q01_ACD (Orders (Basic Query - Demo))

☑ SAP BW Dual Structures support

2.8.2 Considerations when Using SAP Netweaver Business Warehouse Hierarchies

1. If the level structure of a hierarchy changes in the Business Warehouse, you will need to re-import the hierarchy into the metadata model to acquire the latest version. However you do not need to re-import the hierarchy in the event characteristic values are assigned to different tree nodes so long as the tree structure & levels stay intact.

2. Two or more hierarchies may be used within a IBM Cognos 8 query so long as they are assigned to different Business Warehouse characteristics.

3. As is true within SAP Business Warehouse InfoQueries, only one hierarchy can be used per characteristic per IBM Cognos 8 query. But you can use more than one hierarchy of a characteristic in an IBM Cognos 8 report if multiple IBM Cognos 8 queries are used in the report. In short, the rule is one hierarchy per characteristic per query.

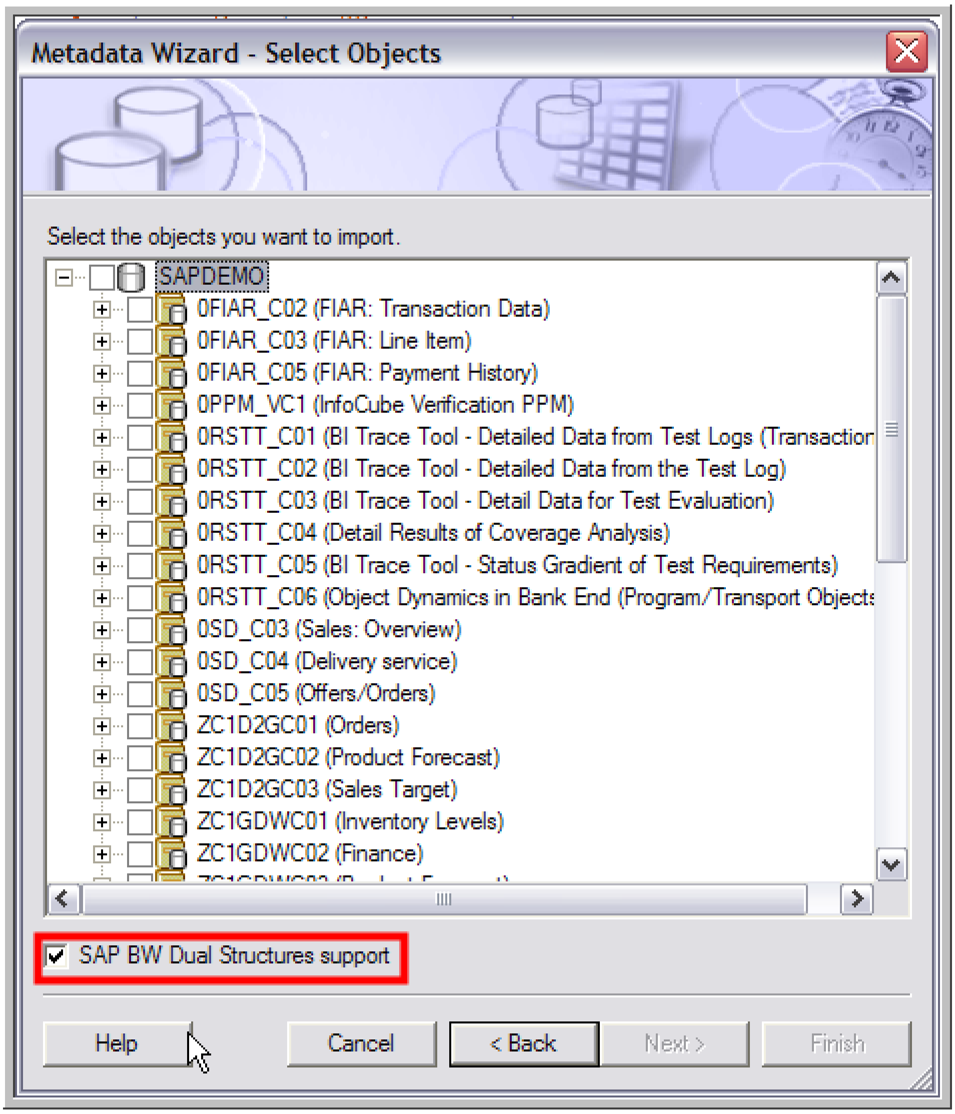
2.9 BEx Queries using Structures

Structures within a BEx query are groupings of characteristics and key figures in an InfoProvider, and are typically used to group and restrict data. There are two types of structures you may encounter within your SAP Business Warehouse environment: Local or Reusable structures.

Local structures are structures built within the SAP Business Warehouse query and can only be used in one query whereas a *reusable structure* is saved at the InfoProvider level and can be used in several queries for the same InfoProvider.

2.9.1 Structures in IBM Cognos 8

The structures (local or reusable) which are referenced within the SAP Business Warehouse InfoQuery can be leveraged and used by IBM Cognos 8. In the event the SAP Business Warehouse query contains two structures (one in rows & one in columns), IBM Cognos 8 will leverage dual structures as well. You just have to ensure the dual structures support box is checked at time of the metadata import via IBM Cognos 8 Framework Manager.



Select the objects you want to import.

- SAPDEMO
  - 0FIAR_C02 (FIAR: Transaction Data)
  - 0FIAR_C03 (FIAR: Line Item)
  - 0FIAR_C05 (FIAR: Payment History)
  - 0PPM_VC1 (InfoCube Verification PPM)
  - 0RSTT_C01 (BI Trace Tool - Detailed Data from Test Logs (Transaction
  - 0RSTT_C02 (BI Trace Tool - Detailed Data from the Test Log)
  - 0RSTT_C03 (BI Trace Tool - Detail Data for Test Evaluation)
  - 0RSTT_C04 (Detail Results of Coverage Analysis)
  - 0RSTT_C05 (BI Trace Tool - Status Gradient of Test Requirements)
  - 0RSTT_C06 (Object Dynamics in Bank End (Program/Transport Objects
  - 0SD_C03 (Sales: Overview)
  - 0SD_C04 (Delivery service)
  - 0SD_C05 (Offers/Orders)
  - ZC1D2GC01 (Orders)
  - ZC1D2GC02 (Product Forecast)
  - ZC1D2GC03 (Sales Target)
  - ZC1GDWC01 (Inventory Levels)
  - ZC1GDWC02 (Finance)

☑ SAP BW Dual Structures support

2.9.2 Considerations When Deciding Whether to Use Structures

- **When to Use:**

  If you have reports with existing structures, you may use these rather than rebuilding within IBM Cognos 8 Report Studio reports. Structures can be helpful within your IBM Cognos 8 reports as they will push data processing back to the SAP Business Warehouse server, much like SAP Business Warehouse filters, restrictions, and variables. Hence when you encounter a SAP Business Warehouse InfoProvider with large volumes of data, you may consider the use of structures to perform certain business rules, manipulations, and restrictions which are desired. This will ensure the data processing is performed at the SAP Business Warehouse server and encourages better overall performance of your Business Intelligence & Performance Management solution.

- **When Not to Use:**

  Structures may limit the audience in that they are typically specific to a business purpose or focused report, which may not be relevant to many IBM Cognos 8 on SAP Business Warehouse users. Carefully consider the application of these structures as they could become a maintenance issue within SAP Business Warehouse if they are used too often. Note that some changes to these structures in SAP Business Warehouse may affect every IBM Cognos 8 report that is using them. A regression test of the IBM Cognos 8 reports will be necessary if changes occur within the structures. In the event you find that structures are getting in the way of the SAP Business Warehouse and IBM Cognos 8 development, consider utilizing more SAP Business Warehouse variables, filters, and restrictions within the InfoQuery rather than using structures.

2.10 SAP Business Warehouse Performance Measures that Compliment IBM Cognos 8

There are several SAP Business Warehouse Reporting Performance factors that you can implement within SAP Business Warehouse which will positively influence the overall performance of your IBM Cognos 8 on SAP Business Warehouse solution. The most common of these SAP Business Warehouse performance measures are noted below, and work seamlessly with IBM Cognos 8 out of the box and are completely transparent to users. You should also refer to your Business Warehouse performance and tuning documentation for SAP performance-tuning recommendations for SAP Business Warehouse.

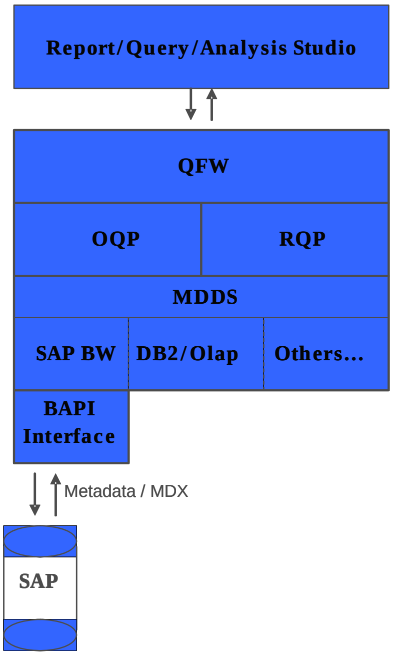
1. **SAP Business Warehouse OLAP Cache:** Accelerates response for similar queries by caching query result sets, and reading from cache instead of the SAP Business Warehouse database. Although this method is the simplest to implement, this is the least flexible of the performance measures and requires exact matching of data subsets.

2. **Cube Aggregates:** Summarized (or subset) materialized views of InfoCube data which are built to enhance data read performance from SAP Business Warehouse cubes. Aggregates are more flexible than OLAP cache, but time to build delays the users' ability to access data.

3. **Database Optimizations:** The application of database indexes, custom statistics, etc, which requires DBA analysis and administration.

4. **SAP Business Warehouse Accelerator:** Installation of a hardware appliance that utilizes in-memory technology to optimize query performance and flexibility when analyzing large amounts of data from SAP Business Warehouse cubes. BWA provides the ability to perform aggregation of the results on the fly and merge results for display.

2.11 Overview of How IBM Cognos 8 Interoperates with the SAP NetWeaver Business Warehouse



The IBM Cognos 8 BI component stack contains multiple layers tasked with processing a request from the studios (Report/Query/Analysis Studio) and returning the result to the end user. The layers start with generalized data source agnostic components (the studios) to refine themselves to very specific components (data source specific providers). In the diagram provided, we can identify the main components related to SAP Business Warehouse (some components have been removed for simplicity).

The Query Framework Coordination Planner (QFW) receives a request from the studios and dissects it into either a relational or an OLAP request. From there QFW then prepares the V5 request following a set of rules for further processing. Some of these rules can be controlled through configuration changes using the qfs_config.xml file.

For SAP Business Warehouse specific requests, the requests are then sent to the OLAP Query Planner (OQP). OQP is tasked with breaking down the OLAP query and constructing data source agnostic MDX statements – also known as "Incoming MDX".

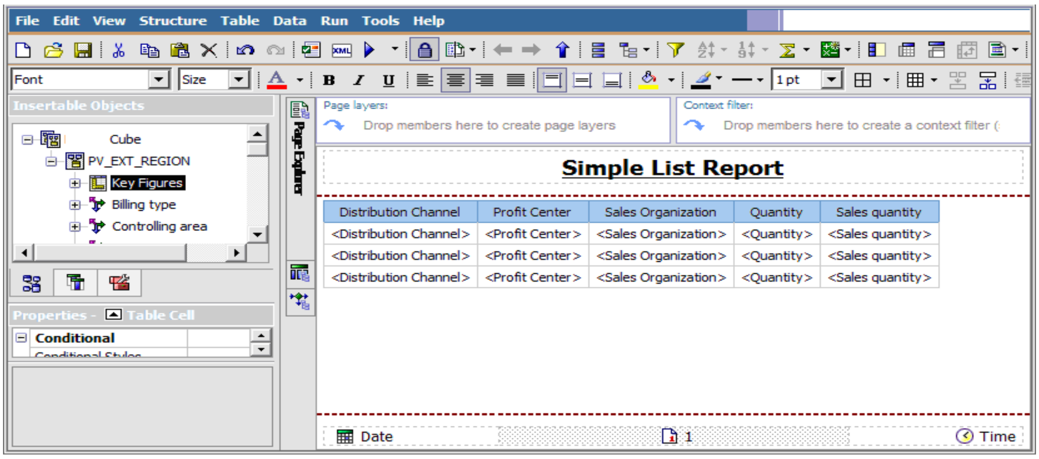
OQP then forwards the "Incoming MDX" to the Multi Dimensional Database Services component (MDDS). It is then recognized as an SAP Business Warehouse query at which time multiple transformations occur to break down the generic MDX and send a series of SAP Business Warehouse BAPI function or MDX statements to SAP Business Warehouse. Configuration changes within the sapbw_config.xml file can alter how and which calls are made by MDDS to SAP Business Warehouse. Once the results are received from Business Warehouse, they are sent back up the component stack for report rendering.

IBM Cognos 8 BI Reporting

Generally speaking the methodology used by IBM Cognos 8 follows a process of decomposing a request to its basic form. All reports eventually resolve themselves to the following steps through a combination of MDX and BAPI_MDPROVIDER_* function calls:

1. Identify the projected dimensions/members/hierarchies.
2. Metadata request to gather all members of the projected dimensions/member/hierarchies.
3. Data Request to retrieve the factual data.

As an example, take the following simple report.



Simple List Report

Distribution Channel	Profit Center	Sales Organization	Quantity	Sales quantity
<Distribution Channel>	<Profit Center>	<Sales Organization>	<Quantity>	<Sales quantity>
<Distribution Channel>	<Profit Center>	<Sales Organization>	<Quantity>	<Sales quantity>
<Distribution Channel>	<Profit Center>	<Sales Organization>	<Quantity>	<Sales quantity>

The report above is comprised of
Three dimensions:

- Distribution Channel
- Profit Center
- Sales Organization

Two Key Figures:

- Quantity (QTY)
- Sales Quantity (Sales QTY)

This simple report will be translated into:

- Metadata Request to gather the enumerated set of members of dimension Distribution Channel
- Metadata Request to gather the enumerated set of members of dimension Profit Center.
- Metadata Request to gather the enumerated set of members of dimension Sales Organization.
- Using the collected enumerated members, an MDX data request is sent to gather factual data using the key figures projected (QTY and Sales QTY)

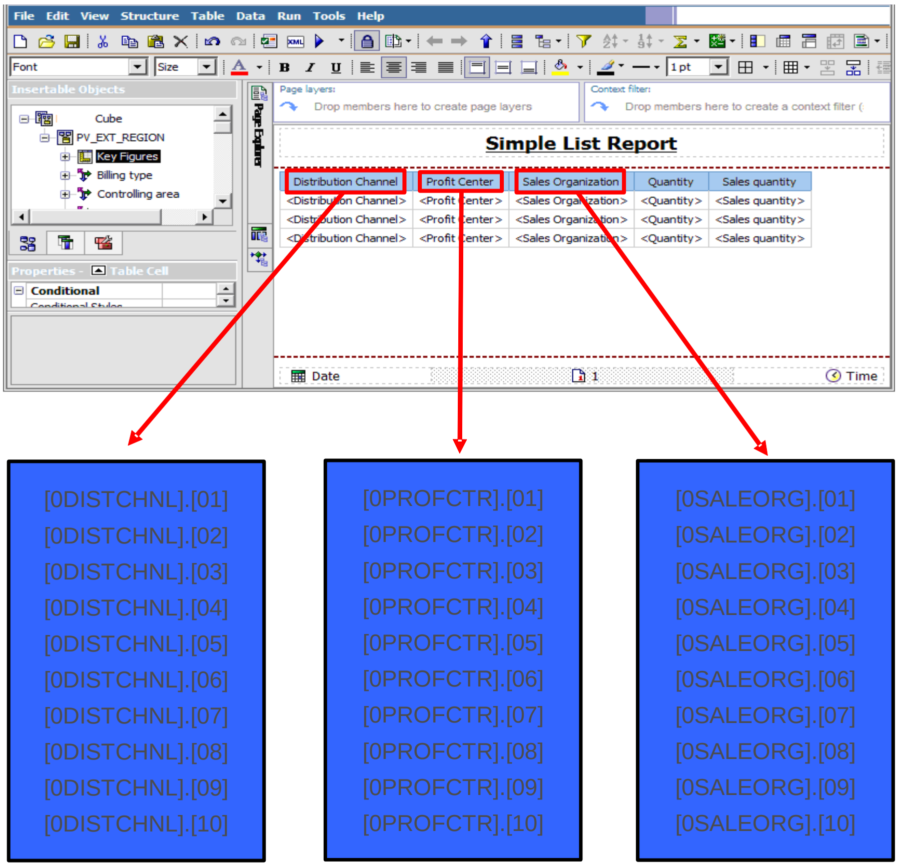
While greatly simplified, with any reports, metadata requests are performed to gather the enumerated set of members, and then followed by data requests to retrieve the factual data.

If we were to demonstrate this graphically, it would follow this process:

Once the “Incoming MDX” is processed, IBM Cognos 8 begins to construct the MDX to send to SAP Business Warehouse to retrieve the results. It takes the following skeleton MDX:

```
SELECT
   {[0DISTCHNL].[LEVEL01].MEMBERS}
   ON AXIS(0),
   {[0PROFCTR].[LEVEL01].MEMBERS}
   ON AXIS(1),
   {[0SALEORG].[LEVEL01].MEMBERS}
   ON AXIS(2),
   {[Measures].[QUANTITY],
    [Measures].[SALESQTY]}
   ON AXIS(3)
FROM
   [ZCOPA/ ZCPA_M04_QKW_COG_001]
   CELL PROPERTIES FORMAT_STRING
```

IBM Cognos 8 BI then proceeds to fetch all enumerated set of members for each represented dimension.



Simple List Report

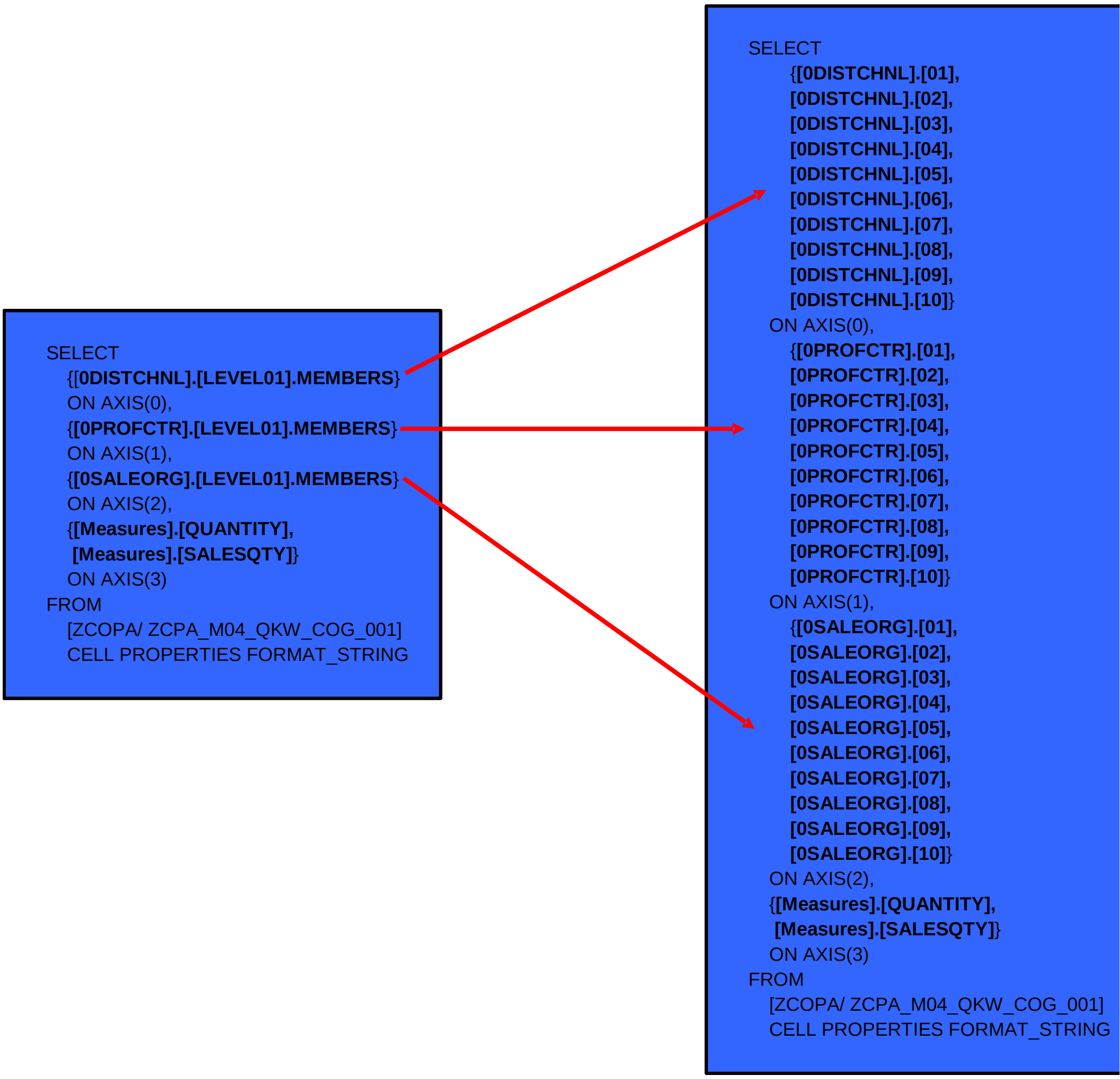
Distribution Channel	Profit Center	Sales Organization	Quantity	Sales quantity
<Distribution Channel>	<Profit Center>	<Sales Organization>	<Quantity>	<Sales quantity>
<Distribution Channel>	<Profit Center>	<Sales Organization>	<Quantity>	<Sales quantity>
<Distribution Channel>	<Profit Center>	<Sales Organization>	<Quantity>	<Sales quantity>

[0DISTCHNL].[01]
[0DISTCHNL].[02]
[0DISTCHNL].[03]
[0DISTCHNL].[04]
[0DISTCHNL].[05]
[0DISTCHNL].[06]
[0DISTCHNL].[07]
[0DISTCHNL].[08]
[0DISTCHNL].[09]
[0DISTCHNL].[10]

[0PROFCTR].[01]
[0PROFCTR].[02]
[0PROFCTR].[03]
[0PROFCTR].[04]
[0PROFCTR].[05]
[0PROFCTR].[06]
[0PROFCTR].[07]
[0PROFCTR].[08]
[0PROFCTR].[09]
[0PROFCTR].[10]

[0SALEORG].[01]
[0SALEORG].[02]
[0SALEORG].[03]
[0SALEORG].[04]
[0SALEORG].[05]
[0SALEORG].[06]
[0SALEORG].[07]
[0SALEORG].[08]
[0SALEORG].[09]
[0SALEORG].[10]

Once retrieved, the final MDX is constructed:



```
SELECT
   {[0DISTCHNL].[LEVEL01].MEMBERS}
   ON AXIS(0),
   {[0PROFCTR].[LEVEL01].MEMBERS}
   ON AXIS(1),
   {[0SALEORG].[LEVEL01].MEMBERS}
   ON AXIS(2),
   {[Measures].[QUANTITY],
    [Measures].[SALESQTY]}
   ON AXIS(3)
FROM
   [ZCOPA/ ZCPA_M04_QKW_COG_001]
   CELL PROPERTIES FORMAT_STRING
```

```
SELECT
      {[0DISTCHNL].[01],
      [0DISTCHNL].[02],
      [0DISTCHNL].[03],
      [0DISTCHNL].[04],
      [0DISTCHNL].[05],
      [0DISTCHNL].[06],
      [0DISTCHNL].[07],
      [0DISTCHNL].[08],
      [0DISTCHNL].[09],
      [0DISTCHNL].[10]}
   ON AXIS(0),
      {[0PROFCTR].[01],
      [0PROFCTR].[02],
      [0PROFCTR].[03],
      [0PROFCTR].[04],
      [0PROFCTR].[05],
      [0PROFCTR].[06],
      [0PROFCTR].[07],
      [0PROFCTR].[08],
      [0PROFCTR].[09],
      [0PROFCTR].[10]}
   ON AXIS(1),
      {[0SALEORG].[01],
      [0SALEORG].[02],
      [0SALEORG].[03],
      [0SALEORG].[04],
      [0SALEORG].[05],
      [0SALEORG].[06],
      [0SALEORG].[07],
      [0SALEORG].[08],
      [0SALEORG].[09],
      [0SALEORG].[10]}
   ON AXIS(2),
   {[Measures].[QUANTITY],
    [Measures].[SALESQTY]}
   ON AXIS(3)
FROM
   [ZCOPA/ ZCPA_M04_QKW_COG_001]
   CELL PROPERTIES FORMAT_STRING
```

2.11.1 Performance Considerations: Metadata Requests

There are two different methods used to retrieve metadata while constructing the final MDX.

1. Through a BAPI function call - BAPI_MDPROVIDER_GET_MEMBERS. This function call returns all members for the requested dimension from the Master Data tables, regardless of whether these members have posted data values recorded or whether they meet the filtered criteria as per defined in variables.

2. Through a MDX statement. This MDX statement is issued with a NON EMPTY clause and as such only records with posted data values are returned. As well, any variable selection will be included therefore also potentially reducing the number of members returned.

BAPI_MDPROVIDER_GET_MEMBERS return more/all members and in most cases returns the members quicker then MDX statements as no fact tables are involved. The adverse effect of using this approach is the potential to bloat the final MDX data request which can have an adverse effect on performance.

The results of the MDX statement restrict the members returned to only those members with fact data associated for the variables defined, meaning the potential exists for fewer members to be included in the final MDX. The adverse effect of using this approach is the possibility that the MDX sent to retrieve the members will take a much longer time than a BAPI_MDPROVIDER_GET_MEMBERS. The advantage being a reduction in the size of the final MDX statement, and therefore, superior performance for the final MDX data request.

There are two distinct configuration settings used to dictate and control which one of the two methods (BAPI vs. MDX) will be utilized to retrieve metadata. This is something IBM Cognos 8 administrators will work with to balance performance (metadata fetch performance vs. data fetch performance). The specifics of these settings will be discussed later in this document.

2.12 The qfs_config.xml Configuration File

The Query Framework component behaviour can be affected by the provider specific files located in the <c8_install>/configuration directory. These are:

- qfs_config.xml This is the file that is read by the Query Framework component to obtain any configuration settings.
- qfs_config.xsd This file is the schema file for the qfs_config.xml file.

2.12.1 The qfs_config.xml Configuration Settings

The following section is a list of the most commonly used settings available for the qfs_config.xml file.

UseSAPMUNAsBusinessKey

In some specific drill through situations, the application would truncate the MUN. Applying this setting may resolve the issue.

```
  <provider name="OlapQueryProvider" libraryName="oqp">
…
<parameter name="UseSAPMUNAsBusinessKey" value="true"/>
```

DetectSAPVariableUniqueness

When importing multiple queries within one package which have identically named variables from multiple InfoQueries – and especially when including multiple InfoQuery within one report, this setting will uniquely identify each variable by prefixing them with the InfoQuery name. Applying this setting will affect all SAP Business Warehouse content.

```
<provider name="OlapQueryProvider" libraryName="oqp">
…
<parameter name="DetectSAPVariableUniqueness" value="true"/>
```

2.13 The sapbw_config.xml Configuration File

As mentioned in the architectural overview. The IBM Cognos 8 SAP provider behaviour can be affected by the provider specific files located in the <c8_install>/configuration directory. These are:

- Sapbw_config.xml: This is the file that is read by the Cognos 8 SAP provider to obtain any configuration settings.
- Sapbw_config.xml.sample: This is a sample file which contains examples of the most used provider configurations.
- Sapbw_config.xsd: This file is the schema file for the Sapbw_config.xml: This file will contain a list of all the currently available settings.

2.13.1 The sapbw_config.xml Configuration Settings

The following section is a list of the mostly used settings available for the sapbw_config.xml file.

DisplayProperty

Impacts: Member metadata fetches for reports, IBM Cognos 8 Report Studio authored prompts.

Usage: This parameter allows a user to change the text display from the default behaviour of caption to other attributes such as medium name, name or key. This configuration applies system wide and will affect all reports.

Interoperability with other parameters: None.

OutputDebugMDX

Impacts: Debugging.

Usage: This parameter controls whether to log/display internal MDX statements. The MDX statements produced by this setting are only interim statements later decomposed prior to sending final MDX statements to the underlying data sources. This parameter is used exclusively for debugging purposes and should be set to false in any other circumstances.

Interoperability with other parameters: While BAPI tracing is activated, the output is created within the BAPI trace.

useMDXToRetrieveMembersLimit

Impacts: Member metadata fetches for reports, IBM Cognos 8 Report Studio authored prompts.

Usage: This parameter controls the decision of when to use MDX or to use the BAPI_MDPROVIDER_GET_MEMBERS BAPI call to retrieve member metadata. MDX will be used if the size of the Hierarchy being queried exceeds the value of this parameter. The default value is 100,000. This parameter affects the queries used to load the Global Cache (if active) as well as regular queries from the provider. Setting the value to 500 as a starting point would be recommended.

Interoperability with other parameters: The parameter UseMDXToRetrieve-MembersFor also affects whether MDX or BAPI calls will be used to retrieve Member data.

Side effect: When using MDX to retrieve metadata from the SAP Business Warehouse InfoQuery/Cube, IBM Cognos 8 will retrieve only the members which have corresponding fact data. When using BAPI_MDPROVIDER_GET_MEMBERS, IBM Cognos 8 will retrieve all members from the master data table regardless of whether the members have corresponding fact data.

UseMDXToRetrieveMembersFor

Impacts: Member metadata fetches for reports, IBM Cognos 8 Report Studio authored prompts.

Usage: In some cases, it can be more efficient to use MDX queries instead of SAP BAPI Remote Function Calls to retrieve Member Information from a Cube on an SAP Server. This setting contains a list of the Unique Names of Dimensions, Hierarchies and/or Members for whom MDX, rather than BAPI Remote Function Calls, should be used to retrieve Member information. This allows customers fine-grained control over when this optimization is used. If this setting contains an empty list or is missing, it has no effect. The default value is an empty list. The parameter useMDXToRetrieveMembersLimit also affects whether MDX or BAPI calls are used to retrieve Member data. This parameter also affects the queries used to load the Global Cache and Cache Server (if either is configured) as well as regular queries from the provider. With crosstabs, in order for this parameter to take effect, the query property "Suppress" MUST be set to something other than "default". If set to default, IBM Cognos 8 will revert to using BAPI_MDPROVIDER_GET_MEMBERS for all metadata fetches.

When using MDX to retrieve metadata from the SAP Business Warehouse InfoQuery/Cube, IBM Cognos 8 will retrieve only the members which have corresponding fact data.

When using BAPI_MDPROVIDER_GET_MEMBERS, IBM Cognos 8 will retrieve all members from the master data table regardless of whether the members have corresponding fact data. Determining which dimensions to include in this section requires a close examination of the dimensions included in the InfoQuery. As a general rule, the dimensions which have a large delta between the number of all members existing in the dimension and the members with fact data associated to them are good candidates. For example, the [0MATERIAL] dimension may contain a total of 10,000 members of which maybe only 100 of them with actual fact data associated – this would be a good candidate for UseMDXToRetrieveMembersFor.

Interoperability with other parameters: The parameter useMDXToRetrieveMembersLimit also affects whether MDX or BAPI calls are used to retrieve Member data.

UseXJoinForLargeSet

Impacts: MDX data fetch approach during reporting.

Usage: Controls the use of Provider Cross Join for large sets - Provider Cross Join will be used whenever the size of the set exceeds the value of this parameter. The default value is 1,000,000. This parameter applies to the SAPBWODP2 (sbwodp) adapter only.

By default, the Business Warehouse cannot return more then 1,000,000 cells, and we need to ensure we do not exceed this limit. In order to do so, when the potential output exceeds this limit, we will be using a provider cross join in order to retrieve the data. This is a system-wide setting and should not be modified unless specifically instructed by IBM Cognos Support. Some benefits could potentially be seen by reducing this number; however it is applied to the entire environment and may not have a positive effect on all reports.

Interoperability with other parameters: None. However, we will hit this 1M threshold quicker if the number of members included in the MDX isn't reduced by the UseMDXToRetrieveMembersFor setting.

UseSAPOrdinalsForMembers

Impacts: Metadata fetch while slicing/filtering on single members.

Usage: When performing filtering/slicing on single members, in order to get the parentage of the specific member, two approaches can be taken: top-down or bottom-up. By this we refer to how metadata is fetched in order to find the lineage (parent to children, or start with specific member and find its parent). By default, IBM Cognos 8.2 uses the top-down approach (false) while IBM Cognos 8.4 uses the bottom-up approach (true). The bottom up approach has proven to provide some performance benefits as fewer metadata calls are performed to obtain the lineage.

Interoperability with other parameters: None.

EnableModelMetadataLookups

Impacts: General metadata queries during connection initialization.

Usage: When establishing a connection to an SAP Business Warehouse environment, releases prior to IBM Cognos 8.4 would perform various function calls to the underlying cube/InfoQuery to obtain some information such as dimensions, hierarchies etc. IBM Cognos 8 now performs these requests against the runtime model, greatly reducing initial connection time. By default this is set to true in IBM Cognos 8.4 and should only be set to false for troubleshooting purposes.

Interoperability with other parameters: None.

UseGetHierarchiesForRootMembers

Impacts: Root member metadata queries during connection initialization.

Usage: Upon establishing a connection to a cube/InfoQuery, in order to improve performance we can retrieve root members for single root hierarchies using information from the GET HIERARCHIES BAPI instead of making a separate GET_MEMBERS call to SAP. This works since the single root member for single root hierarchies is the "All" member and the information required to load it is already available via the GET_HIERARCHIES. By default this is set to false (all releases) causing the application to make BAPI_MDPROVIDER_GET_MEMBERS in order to fetch root members. Alternatively, we can fetch this from the model by setting this parameter to true. Setting the value to true can greatly improve performance should the query contain many dimensions.

Interoperability with other parameters: None.

UseFastGetMembers

Impacts: Metadata queries performed to populate members for SAP Business Warehouse variables for prompts.

Usage: There are two methods of fetching members to satisfy SAP Business Warehouse variables for prompt usage, FastGetMember and BAPI_MDPROVIDER_GET_MEMBERS. Values returned with FastGetMember will only include members with fact data associated to them while BAPI_MDPROVIDER_GET_MEMBERS will return all members. The FastGetMember approach can be quick in most environments while sometimes proving slower when querying SAP Business Warehouse Multiproviders. By default, this parameter is set to true.

Interoperability with other parameters: Can be used in conjunction with UseFastGetMembersFor when set to true.

UseFastGetMembersFor

Impacts: Metadata queries performed to populate members for SAP Business Warehouse variables for prompts.

Usage: In order to offer more flexibility in the use of FastGetMember or GetMember while retrieving members to satisfy SAP Business Warehouse variables for prompt usage, the “UseFastGetMembersFor” allow customers to specify which dimensions they wish to perform FastGetMembers for. In order for this setting to take effect, “UseFastGetMembers” must be set to true. If any dimensions are specified for this parameter, then IBM Cognos 8 will only apply FastGetMember for the specified dimensions.

Interoperability with other parameters: Can be used in conjunction with UseFastGetMembers when UseFastGetMembers is set to true.

LoadMeasuresFromMDQuery

Impacts: Key figure metadata query approach.

Usage: Some reports were designed to use a large number of key figures (600) – more often going against SAP Business Warehouse dual structures. IBM Cognos 8 can either load all measure metadata at once through a BAPI call or load them individually. Earlier releases of the product will load all measures at once and is not configurable through this setting. IBM Cognos 8.4, by default, will load the measures one at a time (default value of false). The cost benefits of switching this setting to true would only apply for environments where we encounter reports with very large number of key figures included.

Interoperability with other parameters: None.

UseReplacementVariablesFor

Impacts: Exposes Replacement Text Variables.

Usage: When making use of Replacement Text Variables, administrators are required to include the Cube/InfoQuery name and dimension name of the targeted dimension within this configuration setting.

For example, if a replacement text variable is defined on a key figure, the following configuration setting would be required:

```
<provider name="SAPBWODP2">
        <parameter name="UseReplacementVariablesFor" value="[CUBE_NAME| INFOQUERY_NAME].[Measures]"/>
</provider>
```

Many entries can be included; they must be separated with commas.

Interoperability with other parameters: None.

2.14 Step by Step Example for Setting an Entry within the SAPBW_CONFIG.xml

The following section will step a user through setting the “UseMDXToRetrieveMembersLimit” feature within the sapbw_config.xml file. The “UseMDXToRetrieveMembersLimit” feature of the IBM Cognos 8 SAP Business Warehouse provider is a performance enhancement that uses MDX queries to replace BAPI calls to retrieve Member data. By default, this feature is turned on for SAP Business Warehouse Dimensions where the number of members exceeds a user-defined threshold. The default user-defined threshold is set to 1,000,00. Member data that is retrieved via MDX will only includes data that has facts. When the BAPI calls are used to retrieve member data, all members are returned whether they have facts or not.

1. Locate the <c8_install>\configuration\sapbw_config.xml file and make a backup copy.

2. Open the original file using a text editor.

3. Locate the string “"SAPBWODP2"”.

4. Add the following line under the <parameters> tag to activate the UseMDXToRetrieveMemberLimit configuration

5. Locate the UseMDXToRetrieveMembersLimit parameter. The completed entry should read as follows:

```
<provider name="SAPBWODP2">
            <parameters>
                 <!-- Output processing MDX statements into debug window - set to "true" if
                 required -->
                 <parameter name="useMDXToRetrieveMembersLimit"
                 value="100000"/>
                 ...
```

6. The value can be changed from 100000 to the desired number of members.

Save the changes and close the file. This configuration setting will be picked up by any new BiBusTKServerMain process that is launched

3 IBM Cognos 8 Framework Manager

IBM Cognos 8 Framework manager provides the metadata model development environment in IBM Cognos 8 and is the starting point for IBM Cognos 8 and SAP Business Warehouse integration. It is important to note that you cannot add or edit SAP Business Warehouse relationships. The Star layout is fact table centric and is the appropriate choice when viewing SAP Business Warehouse metadata. You can use any kind of InfoCube directly (InfoCube, VirtualCube, RemoteCube, MultiProvider) or any SAP Business Warehouse Query on top of them or other Business Warehouse InfoProviders (including ODS/DSO Data, Master Data, and InfoSets). Although options are available to you to use InfoProviders directly, it is highly recommended to use a SAP Business Warehouse query which acts as a database view and provides flexibility to restrict data from the SAP Business Warehouse.

As the general guidelines encourage users to leverage the SAP Business Warehouse query flexibility, the following should also be taken into consideration:

1. Perform as few changes as possible within the IBM Cognos 8 Framework Manager project to avoid re-work in the event a query needs to be re-imported.
2. Limit new calculations in IBM Cognos 8 Framework Manager, but rather build those in SAP Business Warehouse query designer or in the IBM Cognos 8 Studios.

It’s worth noting for faster metadata import you should only import hierarchies that will be used for reporting. This usually cuts down on the objects to import dramatically. In order to improve studios performance it is recommended to leverage the modelling performed in each SAP Business Warehouse query. If importing metadata for a Query/InfoCube is taking too long or an SAP Business Warehouse memory limitation is hit, use the recommendation above to try to streamline the import, or split the query in smaller chunks by using the option enhance and Framework Manager will perform even faster.

3.1 Mapping SAP Business Warehouse Objects to IBM Cognos 8 Framework Manager

SAP Business Warehouse objects are mapped to the following IBM Cognos 8 Framework Manager objects.

SAP Business Warehouse Object	**IBM Cognos 8 Framework Manager Object**
Query, InfoCube, RemoteCube, MultiCube	Namespace.
Characteristic	A folder that contains dimensions. You must select the **Enhance model for SAP BW organization of objects** option when importing metadata to have a folder for each characteristic for each dimension. **Note:** By default, Framework Manager imports SAP Business Warehouse Currency and Unit of Measure characteristics. You can remove these characteristics if you do not need them.
Dimension	Dimension. The dimension may contain hierarchies representing each presentation hierarchy. The default hierarchy contains two levels representing the aggregation of all characteristic values, also known as the All value aggregating across all characteristic values
Key figure	Query item that is part of a measure dimension called Key Figures.
Presentation hierarchy level	Level. **Note:** Level names must be defined in the Administrator Workbench to be meaningful.
Attribute	Query item associated with a level whose **Usage** property value is set to **Attribute**.
SAP Business Warehouse variable	Data source property. For information about the SAP Business Warehouse variables that Framework Manager supports, see SAP Business Warehouse Variables.

3.1.1 SAP Business Warehouse Variables

SAP Business Warehouse variables are parameters of an SAP Business Warehouse Query that are set up during query definition. When queries run, the SAP Business Warehouse variables are filled with values. These function as placeholders and can be processed in different ways. These are automatically exposed as prompts at run time.

SAP Business Warehouse variable information is included in a composite custom property named “***SAP BW Variables***” that exists only if a data source has one or more variables associated with it. The “***SAP BW Variables***” property contains one or more composite properties, each of which must be assigned a unique name. Each property represents a description of a single Business Warehouse variable. Because the variable information is specified in a custom property, Framework Manager does not validate these properties.

The SAP Business Warehouse variable information is obtained using the SAP BW BAPI *MDDataProviderBW::GetVariables*. IBM Cognos 8 Framework Manager supports the following types of SAP Business Warehouse variables:

characteristic

There are two kinds of characteristic variables, characteristic value and hierarchy node. Characteristic values variables select characteristic values. Hierarchy node variables select values from any position in a presentation hierarchy.

hierarchy

The user is not prompted for a value because IBM Cognos 8 automatically populates it at run time based on the selected hierarchy. Variables for hierarchies function act as placeholders for the hierarchy of a characteristic. All the values for hierarchy variables are read-only.

formula

The user types a numeric value at run time. Use formula variables if a formula component should be entered only when a query is run. For example, you can use a formula variable for a value-added tax rate to process the current rate at run time.

authorization

Authorization variables are like other variables, but IBM Cognos 8 automatically populates the variable values with the user’s credentials. The Business Warehouse uses these credentials to supply the information needed by an SAP Business Warehouse Query that has security applied to it.

3.1.2 Name Property

This property is a string value.

SAP Business Warehouse equivalent: VARIABLE_NAME

Restrictions: Read-only.

3.1.3 Caption Property

The string value for this property is a composite and locale-dependent. Each locale in the model should be represented by a custom property whose value is the locale name. For example, if the locales en-ca and fr-fr exist in the model, define two custom properties named en-ca and fr-fr.

The default value is obtained from SAP Business Warehouse.

3.1.4 Default Low Caption and Default High Caption Properties

The value for each of these properties is a composite, locale-dependent string value. Each locale in the model should be represented by a custom property whose value is the locale name. For example, if the locales en-ca and fr-fr exist in the model, define two custom properties named en-ca and fr-fr.

The default value is obtained from SAP Business Warehouse.

Restrictions: The **Default High Caption** properties are applicable only for variables with a **Selection Type** of **interval**.

3.1.5 Selection Type Property

The possible values are value, interval, complex, multiValued.

Value	SAP Business Warehouse Equivalent
value	SAP_VAR_SEL_TYPE_VALUE
interval	SAP_VAR_SEL_TYPE_INTERVAL
complex	SAP_VAR_SEL_TYPE_COMPLEX
multiValued	SAP_VAR_SEL_TYPE_COMPLEX

Restrictions: Read-only.

3.1.6 Entry Type Property

The default value is obtained from the SAP Business Warehouse.

Value	SAP Business Warehouse Equivalent
optional	SAP_VAR_INPUT_TYPE_OPTIONAL
mandatory	SAP_VAR_INPUT_TYPE_MANDATORY
mandatoryNotInitial	SAP_VAR_INPUT_TYPE_MANDATORY_NOT_INITIAL

Restrictions: Read-only.

3.1.7 Default Low Value and Default High Value Properties

Each of these properties specifies a range of values.

The default value is obtained from SAP Business Warehouse.

Restrictions: The **Default High Value** property is applicable only for variables with a **Selection Type** of **interval**.

3.1.8 Description Property

This property is a string value.

3.1.9 SAP Business Warehouse Variable Type Property

The possible values are numeric, characteristic, hierarchy, or hierarchicalNode.

The default is obtained from SAP Business Warehouse.

Restrictions: Read-only.

3.1.10 Prompt Type Property

The default value depends on the type of the variable. If the value of this property is not one of the predefined values, it is assumed to be hierarchyPickList.

Value	**Restrictions**
typeIn	Required for numeric variables and optional for characteristic values
pickList	Optional for characteristic variables
calendar	Only for characteristic variables based on 0CALDAY
hierarchyPickList	Optional for all presentation hierarchies
notApplicable	Required for hierarchy variables

Restrictions: Read-only for some types of variables such as characteristic and formula. typeIn can be changed to pickList as well as pickList to typeIn and apply to characteristic variables.

3.1.11 Level Restriction Property

This property is a numeric value.

The default value is **1**.

Restrictions: Applicable only for hierarchical node variables with a **Prompt Type** of **hierarchyPickList**.

3.1.12 Trim Levels Property

This property is a string value that reduces the number of members in a hierarchical picklist. If the property is set to zero (0), members from all levels of a hierarchy are included in the prompt. You can also specify a range such as 2:4 to include only the members from certain levels. If the starting and ending ranges are the same, such as 3:3, only members from that level will be included.

The default value is zero (0).

Restrictions: Applicable only for characteristic variables with a **Prompt Type** of **hierarchyPickList**.

3.1.13 Use Default Values Property

This property is a Boolean property that determines whether the default values will be used. If this property is set to **true**, your users are not prompted for the associated variable, and the default value is always applied.

The default value is **false**.

3.1.14 Show Key and Caption Property

To show keys and captions for the SAP Business Warehouse variables, set this property to **true**. This property is applicable only for pick list prompts and hierarchy node prompts.

The default value is **false**.

3.1.15 Initial Number of Pick List Values Property

A numeric value that specifies the initial number of values used to populate a picklist, hierarchical picklist, or prompt.

The default value is zero (0), which means all.

3.1.16 Numeric Variable Property Values

The following variable properties are applicable to numeric variables:

Property	**Default Value**
Type	Numeric
Caption	
Selection Type	Value
Entry Type	obtained from the SAP Business Warehouse
Default Low Value	
Default High Value	
Prompt Type	typeIn
Use Default Value	False

You can change the default values for a numeric variable except for the Prompt Type property, which is read-only.

3.1.17 Characteristic Variable Property Values

There are two kinds of characteristic variables, characteristic value and hierarchy node. Characteristic values variables select characteristic values. Hierarchy node variables select values from any position in a presentation hierarchy.

3.1.18 Characteristic Value Variable Property Values

The following variable properties are applicable to characteristic value variables:

Property	Default Value
Type	Characteristic
Caption	
Selection Type	obtained from the SAP Business Warehouse
Entry Type	obtained from the SAP Business Warehouse
Default Low Value	If the entry type is value or complex, the default property is shown. If the entry type is interval, the default low property is shown. This value is obtained from the SAP Business Warehouse.
Default High Value	If the entry type is value or complex, the default property is shown. If the entry type is interval, the default high property is shown. This value is obtained from the SAP Business Warehouse.
Prompt Type	typeIn or pickList This depends on the number of members in the referenced dimension. If the value is invalid, typeIn is used.
Use Default Value	False
Show Key and Caption	False
Initial Number of Pick List Values	zero (0)

A characteristic value variable for the 0CALDAY dimension is shown in the model as a date. The Data Type property is set to xsdDate and the Prompt Type property is set to calendar. The Prompt Type property is read-only for the 0CALDAY dimension.

3.1.19 Hierarchy Node Variable Property Values

The following variable properties are applicable to hierarchy node variables:

Property	**Default Value**
Type	characteristic
Caption	
Selection Type	obtained from the SAP Business Warehouse
Entry Type	obtained from the SAP Business Warehouse
Default Low Value	
Default High Value	
Prompt Type	hierarchy PickList You can change the **Prompt Type** property to **typeIn** or **pickList**.
Level Restriction	zero (0)
Use Default Value	False

3.2 Picklist Prompts

Each picklist prompt contains a pre-defined number of values. These values are determined by the **Maximum Number of Values** property.

If the number of actual values is less than or equal to the default number of values, the prompt is generated as a single picklist prompt. If the number of actual values exceeds the default number, two prompts are generated in this order:

- a bound range parameter with a starting value of **1** and an ending value determined by the **Maximum Number of Values** property. This parameter is of the type xsdUnsignedLong and is optional. The name of the parameter is the name of the original prompt followed by _range_prompt. The caption is locale-specific. If this is a multilingual model, you must store the template for the caption in a message file.
- a picklist prompt containing the default number of values

3.3 Dimensions

A dimension is a broad grouping of data about a major aspect of a business, such as products, dates, or markets.

The types of dimensions that you can work with in IBM Cognos 8 Framework Manager are regular dimensions and measure dimensions. In the SAP Business Warehouse, measure dimensions are called key figures.

For example, in a project for sales analysis, you include these dimensions:

Name	Type	Description
Time	Regular dimension	Dates of sales organized into years, quarters, months, weeks, and days when sales were made
Region	Regular dimension	Locations of sales grouped into sales regions, countries, and cities
Product	Regular dimension	Product details organized by product type, brand, model, color, and packaging
Customer	Regular dimension	Customer information
Sales	Key Figures dimension	Purchase details such as units sold, revenue, and profit

3.3.1 Modify a Regular Dimension

A regular dimension contains descriptive and business key information and organizes the information in a hierarchy, from the highest level of granularity to the lowest. It usually has multiple levels and each level requires a key and a caption. If you do not have a single key for your level, it is recommended that you create one in a calculation.

Model regular dimensions are based on data source or model query subjects that are already defined in the model. You must define a business key and a string type caption for each level. When you verify the model, the absence of business keys and caption information is detected. Instead of joining model regular dimensions to measure dimensions, create joins on the underlying query subjects and create a scope relationship between the regular dimension and the measure dimension.

When dimensions are based on SAP Business Warehouse metadata, you cannot edit the underlying query.

1. Click the regular dimension you want to modify.
2. From the **Actions** menu, click **Edit Definition**.
3. Choose the action that you want:
4. Embed calculations by selecting the level, clicking **Add**, and then defining the expression

5. Embed filters
6. Test the dimension
7. Click **OK**.

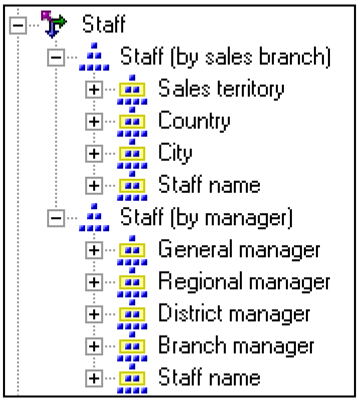
3.3.2 Hierarchies for a Regular Dimension

A hierarchy is an ordered list of levels or a collection of items. Each query item in a hierarchy must have a unique name.

You can specify multiple hierarchies on regular dimensions in Framework Manager. Multiple hierarchies for a regular dimension behave as views of the same query. The first hierarchy is the primary or default hierarchy.

You can use only one hierarchy at a time in a query. For example, you cannot use one hierarchy in the rows of a crosstab report and another hierarchy from the same dimension in the columns. If you need both hierarchies in the same report, you must create two dimensions, one for each hierarchy. For more information, see Modeling Dimensions with Multiple Hierarchies in the IBM Cognos 8 Framework Manager User Guide

For example, sales staff can be viewed by manager or by geography and can be modeled as a single dimension with two hierarchies.

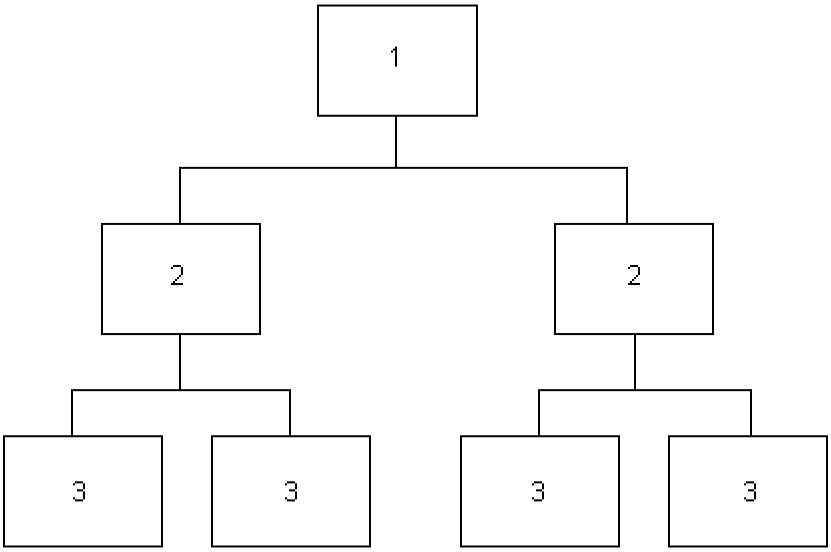


- Staff
  - Staff (by sales branch)
    - Sales territory
    - Country
    - City
    - Staff name
  - Staff (by manager)
    - General manager
    - Regional manager
    - District manager
    - Branch manager
    - Staff name

3.3.3 Balanced Hierarchy

Each path in a balanced hierarchy descends to the same depth.

For example, in the following diagram, the highest level is Product Line. Level 2 is Product Type. Level 3 is Products.



In SAP Business Warehouse, all leaf nodes of a hierarchy are values of the characteristic, but each path does not have to descend to the lowest level of the hierarchy.

3.3.4 Levels for a Regular Dimension

A level is a collection of attributes, typically of a common granularity. Each level needs an item that is defined as a key and another item that is defined as a caption. For the SAP Business Warehouse data sources, levels contain members.

The first level of the hierarchy is automatically defined as the All level. It contains a single root member, which represents the top level of the hierarchy. For example, the All level for the Time dimension is named Time (All). You cannot delete or move the All level. You can change its name, description, and screen tip.

If you do not specify the levels of the hierarchy correctly, incorrect aggregation could occur.

3.4 Member Unique Names

The member unique name (MUN) is how the member is found in the data source, much like using business keys to find records in a table.

The member unique name is used in the expression for a member data item that is used in a report, a reference to members in filters and expressions, and used in drill-through between OLAP data sources. The member keys in the MUN for the different OLAP data sources must match.

3.5 Roles

Roles define what appears in the member tree in the IBM Cognos 8 studios. Use roles to organize and manage metadata and to determine how to present data to your users.

You can also create expressions that refer to roles instead of query items. You must use the `roleValue` function to refer to a particular role. For example, you want to query against a specific role in a hierarchy but the query item playing that role is different at each level of the hierarchy. A single query can span the different query items at each level. You can also use the `roleValue` function when you know the role but not the underlying query item.

You can assign multiple roles to one query item, but the same role cannot be assigned to different query items in the same level.

Default roles are pre-defined for all parent-child hierarchies and for all levels in level-based hierarchies. Most of these roles are not visible in the IBM Cognos 8 studios.

The roles that are reserved by IBM Cognos 8 start with an underscore. The name for a custom role cannot start with an underscore.

3.6 SAP Business Warehouse Hierarchies

When importing metadata, IBM Cognos 8 Framework Manager generates a dimension in each SAP Business Warehouse characteristic.

Only one hierarchy associated with a given characteristic can be used in a report. Therefore, you should group dimensions that represent the hierarchies of a single characteristic into a folder or model query subject to make reporting easier for your users.

If there are multiple hierarchies in an SAP Business Warehouse data source, the first hierarchy that is imported becomes the default hierarchy.

IBM Cognos 8 Framework Manager supports the following types of hierarchies:

characteristic

This is a list of all the characteristic values.

text node

Non-leaf nodes contain only text and do not reference any other data source object.

characteristic value

The nodes of each level of a presentation hierarchy are values from another characteristic.

recursive

The nodes of the entire presentation hierarchy are from the characteristic itself.

If a characteristic is not in a time dimension but it is a date and is treated as a date in the SAP Business Warehouse, the characteristic is imported with the date data type.

IBM Cognos 8 Framework Manager does not support hierarchies that contain two or more types of nodes. These hierarchies are imported but are hidden in the Framework Manager model.

Because hierarchical metadata is automatically generated for the SAP Business Warehouse, you cannot change it within the IBM Cognos 8 Framework Manager.

3.6.1 Versioned Hierarchies

You can import the following types of versioned hierarchies from an SAP Business Warehouse data source:

Version dependent hierarchy

A hierarchy can have multiple versions. Each version of a hierarchy can have a different structure, such as Sales by Region and Sales by Manager. During metadata import, Framework Manager identifies each version as a separate hierarchy and creates a dimension for each.

Entire hierarchy time dependent

Each version has an associated time period that does not overlap with any other version of the same hierarchy. The structure of each version can be different. During metadata import, IBM Cognos 8 Framework Manager identifies each version as a hierarchy and includes the applicable time period as part of the dimension name.

3.7 Importing SAP Metadata into IBM Cognos 8 Framework Manager

IBM Cognos 8 Framework Manager is a Windows-based client application as opposed to the IBM Cognos 8 Studios in IBM Cognos 8 Connection which are web-based applications. When you work in IBM Cognos 8 Framework Manager you work in a project. At the highest level objects in a project include a model, namespaces or folders to organize and provide uniqueness to objects with the same name, data sources to define your connection to the underlying data sources, parameter maps to allow for the substitution of parameter values at runtime to match items found in your data, and packages to define the presentation of reporting objects made available to authors.

The IBM Cognos 8 Framework Manager workflow is an iterative process where the model is typically revisited to make adjustment for performance, security or business requirements. Initially modellers create a project. This phase requires diligence in gathering reporting requirements and understanding your data and its structure, and ensuring that you have the appropriate data access plan. In other words is the data source you intend to use the best structure for your reporting and analysis needs? Once you are ready to import metadata ensure you have an appropriate project structure and import only the required metadata to keep your modelling environment manageable.

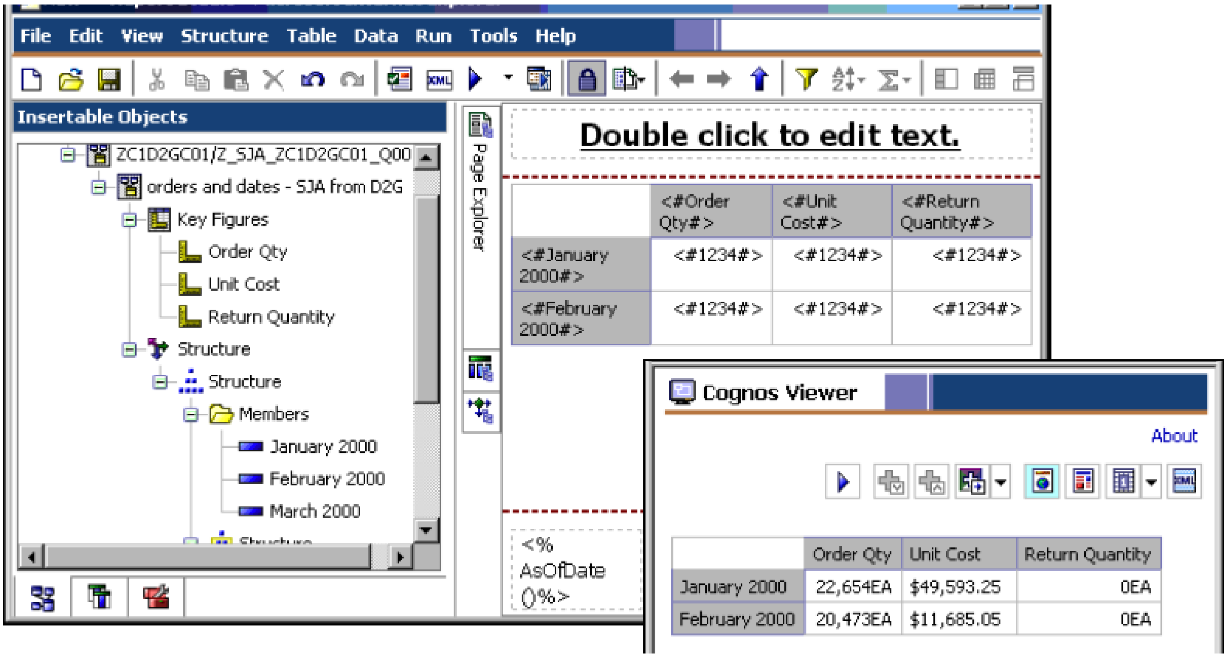
3.8 SAP Business Warehouse Structures

Many existing SAP Business Warehouse queries that will be used in an IBM Cognos 8 application contain structures, and you may plan to use them in queries prepared for an IBM Cognos 8 application to control the amount and the order of information that your report users see.

The SAP Query Designer automatically creates a key figure structure when you add key figures to a query. Although the SAP Business Warehouse does not require key figures in a query, you must have at least one key figure to import the query metadata into Framework Manager. This is true even when you do not use the key figure in the IBM Cognos 8 reports. Therefore, you will always have a key figure structure.

A characteristic structure is a collection of characteristic values (members) from one or more dimensions. You create a characteristic structure within the Business Warehouse by adding a structure to the query, and then adding the required members to the structure. In IBM Cognos 8.4, the structure appears as an additional dimension that has only one, multi-root level. If you use a characteristic structure as well as a key figure structure (dual structures) in your SAP Business Warehouse query, then you create a table with fixed cell definitions for reporting. You can set filters and so on as you would for any other SAP Business Warehouse query.

When you import an SAP Business Warehouse query into IBM Cognos 8 Framework Manager and publish a package to the reporting studios, the key figure structure appears in the measures dimension, and the characteristic structure appears as any other dimension. That is, you can drag it onto a report, and select individual and multiple members.



	<#Order Qty#>	<#Unit Cost#>	<#Return Quantity#>
<#January 2000#>	<#1234#>	<#1234#>	<#1234#>
<#February 2000#>	<#1234#>	<#1234#>	<#1234#>

	Order Qty	Unit Cost	Return Quantity
January 2000	22,654EA	$49,593.25	0EA
February 2000	20,473EA	$11,685.05	0EA

3.9 Step by Step Data Source Connection and Import for SAP Business Warehouse

3.9.1 Creating a Data Source Connection

A data source defines the physical connection to a database. A data source connection specifies the parameters needed to connect to a database, such as the location of the database and the timeout duration. These parameters form a connection string for the data source.

You can create data sources in the portal or in IBM Cognos 8 Framework Manager. Because they are stored on the server, data sources appear in both places, regardless of where they were created. Existing data source connections can be edited only in the portal.

If you are an administrator, you can set up all required data sources before models are created in Framework Manager so that all connections are available in the Framework Manager Metadata wizard.

Data sources are stored in the IBM Cognos 8 namespace and must have unique names. For example, you cannot use the same name for a data source and a group.

You can include authentication information for the database in the data source connection by creating a signon. Users need not enter database authentication information each time the connection is used because the authentication information is encrypted and stored on the server. The signon produced when you create a data source is available to the Everyone group. Later, you can modify who can use the signon or create more signons. For more information about data source security, see the *Administration and Security Guide*.

Before creating data sources, you must have write permissions to the folder where you want to save the data source and to the IBM Cognos 8 namespace. You must also have execute permissions for the **Data Source Connections** secured feature. To create a data source connection within IBM Cognos 8 Framework Manager.

1. If you are not already using Run Metadata Import Wizard, click the namespace folder, or segment you want to import into, and from the **Actions** menu, click **Run Metadata Wizard.**
2. In the Select Metadata Source window, click Data Sources.
3. Click Next.
4. In the Select Data Source window, click New.
5. In the name and description page, type a unique name for the connection and, if desired, a description and screen tip, and then click **Next.**
6. In the New data source page, enter the Application server name (SAP Business Warehouse server), System number, Client number and also select the type of Authentication (Provide Userid and Password).

3.9.2 Steps to Import from an SAP Business Warehouse Data Source

1. Ensure there is a connection to the data source
2. Click the namespace, folder, or segment you want to import into and from the **Actions** menu, click **Run Metadata Wizard**.
3. Select a data source connection and click **Next**.
4. Select the objects you want to import.
5. If you are re-importing, the existing object gets updated.
6. If you are importing new objects and an object with the same name exists, the new object is imported and a number is appended to the original name. For example, you see QuerySubject and QuerySubject1 in your project.

7. After they are imported, query items cannot be deleted without deleting the entire query subject.
8. If you want to import a characteristic structure and a key figure structure, select the "***SAP BW Dual Structures Support***" check box.
9. The content of the **Select Objects** page is updated to reflect the dual structures in your data source.

   If you are re-importing the same SAP Business Warehouse query (for example, because the underlying data source has changed), you must use the same setting for this check box. IBM Cognos 8 Framework Manager does not allow you to select a different setting for the same query because different objects are then generated in the model and this leads to errors.
10. Select the languages you want to import.

    These languages must exist in the SAP Business Warehouse data source.

    You can add languages to your project later, but you cannot go back and import the language-specific metadata from the data source. After the import is complete, the language-specific metadata must be added manually.
11. Indicate whether you want IBM Cognos 8 Framework Manager to show the short name, long name or the technical name for the dimensions.
12. If you select short name and the field is empty, the long name is shown. If you select the long name and the field is empty, the short name is shown. If you select either short name or long name and both fields are empty, the technical name is shown.
13. To have objects in the model organized the same way that they are organized in Business Explorer Query Designer, select the "***Enhance model for SAP BW organization of objects***" option. You will then have a folder for each characteristic.
14. Click **Next**.

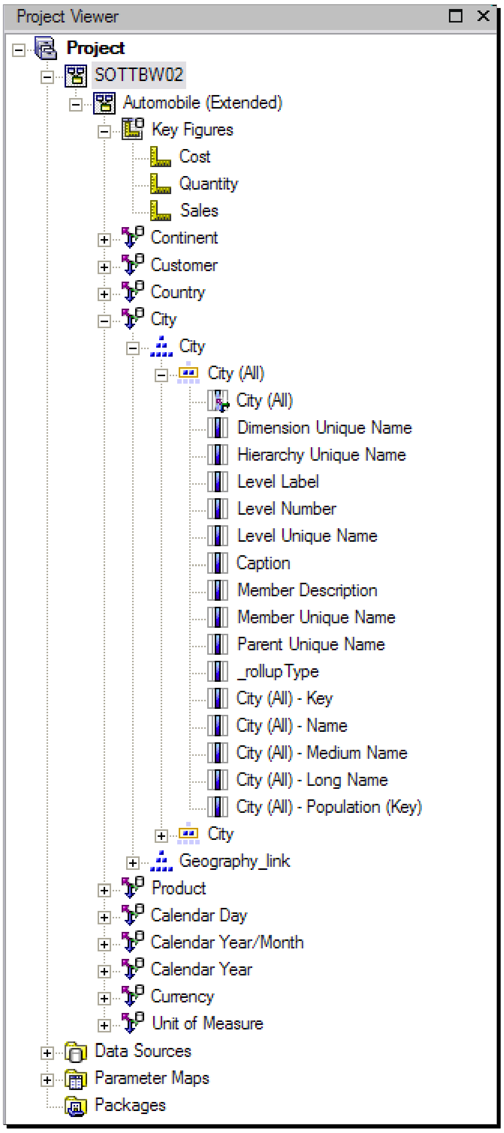
    A list of objects that could not be imported appears along with counts of objects that were imported.
15. Click **Finish**.

3.9.3 Examining the SAP Business Warehouse Metadata after an Import

To explore the SAP Business Warehouse metadata within IBM Cognos 8 Framework Manager:

In the Project Viewer pane, expand the SAP Business Warehouse namespace. The namespace now contains a list of children namespaces, which represent each of the SAP Business Warehouse Queries that were imported from the SAP Business Warehouse data source.

The graphic below depicts the appropriate Basic Query, Key Figures, Levels and Attributes of the Automobile Query example. These contain the appropriate query subjects that were generated, during the import, representing key figures and dimensions.



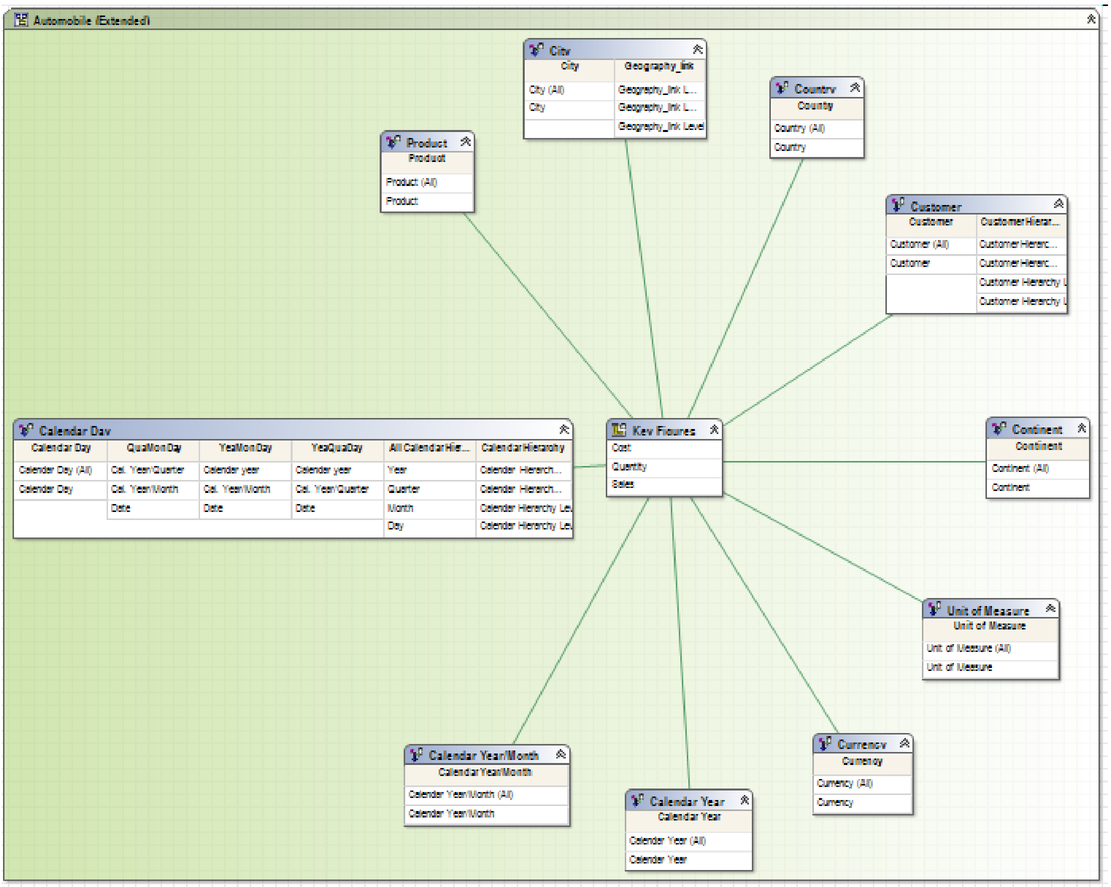
Project Viewer

- Project
  - SOTTBW02
    - Automobile (Extended)
      - Key Figures
        - Cost
        - Quantity
        - Sales
      - Continent
      - Customer
      - Country
      - City
        - City
          - City (All)
            - City (All)
            - Dimension Unique Name
            - Hierarchy Unique Name
            - Level Label
            - Level Number
            - Level Unique Name
            - Caption
            - Member Description
            - Member Unique Name
            - Parent Unique Name
            - _rollupType
            - City (All) - Key
            - City (All) - Name
            - City (All) - Medium Name
            - City (All) - Long Name
            - City (All) - Population (Key)
          - City
        - Geography_link
      - Product
      - Calendar Day
      - Calendar Year/Month
      - Calendar Year
      - Currency
      - Unit of Measure
  - Data Sources
  - Parameter Maps
  - Packages

3.9.4 Configure and Explore the Object Diagram

1. In the Project Info View, click the Diagram link.
2. The various namespaces that were created during the import appear.
3. Expand the Query namespace.
4. Notice the layout of the objects. We would like to represent a Star Schema where the fact table (key figures) is at the center of all its related dimensions.
5. On the toolbar, click the Auto Layout button.
6. In the Layout Style drop down box, select Star.
7. Slide the Sibling Distance slider all the way to the left to make the diagram more manageable to read, click Apply, and then and then click Close.
8. Go to Diagram – Zoom and select Fit All to automatically resize the view

9. The layout has changed to a Star layout and appears as shown below.



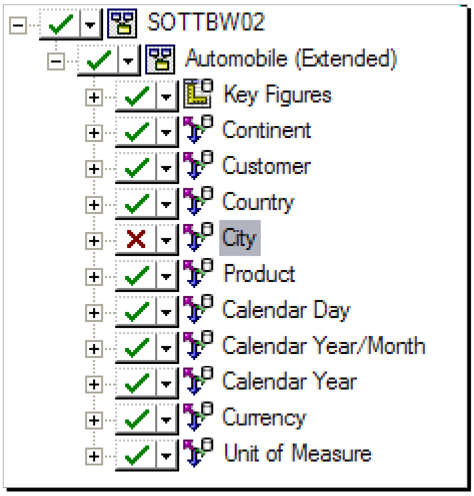
3.9.5 Create and Publish a Package for Power Users.

1. In the Project Viewer pane, right-click Packages, point to Create, and then click Package.

2. The Create Package Wizard appears.

3. In the Name box, type (Your Package Name), and then click Next.

4. If required, change the default selections to exclude any of the items in your package (the image below depicts City as an example), and then click Finish.



- ✓ SOTTBW02
  - ✓ Automobile (Extended)
    - ✓ Key Figures
    - ✓ Continent
    - ✓ Customer
    - ✓ Country
    - ✗ City
    - ✓ Product
    - ✓ Calendar Day
    - ✓ Calendar Year/Month
    - ✓ Calendar Year
    - ✓ Currency
    - ✓ Unit of Measure

5. A dialog box appears indicating that the package was created successfully, and prompts to open the Publish Package wizard.

6. Click Yes.

7. If necessary, clear the Enable model versioning check box, then accept the remaining defaults, and then click Publish.

8. A message appears within the wizard indicating that the package was successfully published.

9. Click Finish to close the Publish Package wizard.

10. From the File menu, click Save to save the Framework Manager model.

3.10 Modelling SAP Business Warehouse metadata and Naming Conventions

When modelling a SAP Business Warehouse data source within IBM Cognos 8 Framework Manager it may be beneficial to consider the following approaches.

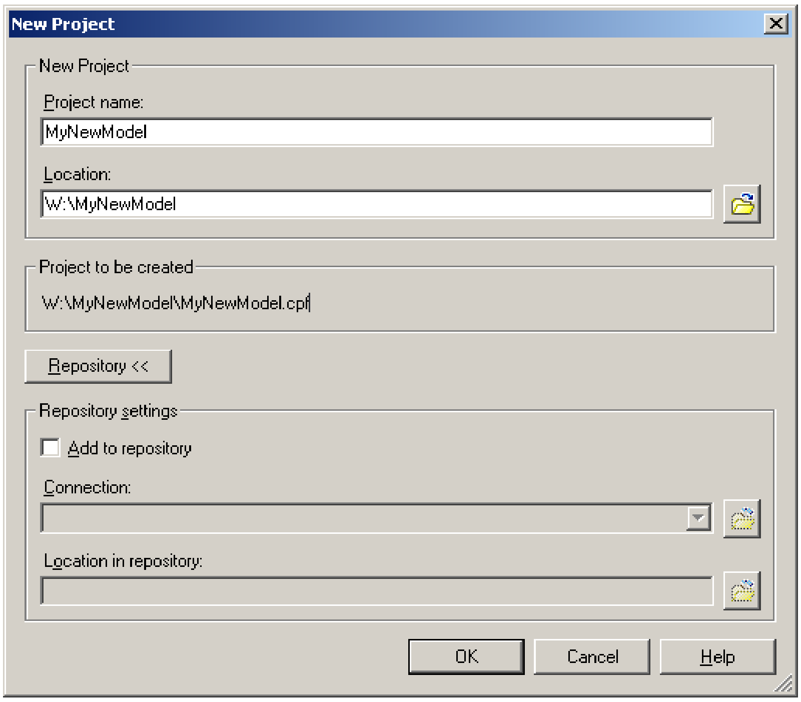
- **Framework Manager Directory Structure**
  To help you manage, share, and secure different versions of your metadata, you can configure IBM Cognos 8 Framework Manager to use a source control system. You must already have one of the following source control system clients set up on the same computer as IBM Cognos 8 Framework Manager:
  - o Component Software Concurrent Versions System
  - o Visual Source Safe

  For more information about installing and setting up source control systems, see Repository Control in the IBM Cognos 8 Framework Manager User Guide.

  If you do not have the source control system you may be able to take advantage of existing infrastructure backup procedures. Generally LAN locations are backed up at regular intervals. Therefore any projects saved there would be backed up as well. In order to implement this:

1. Create a folder on a LAN location available to the model developers.
2. Secure the folder using the NTLM folder security
3. On the Framework Manager system map a network drive to LAN share

4. When creating a new IBM Cognos 8 Framework Manager model specify the mapped drive location created in the previous step.



New Project

New Project

Project name: MyNewModel

Location: W:\MyNewModel

Project to be created: W:\MyNewModel\MyNewModel.cpf

Repository settings

□ Add to repository

Connection:

Location in repository:

In the example above the project name is MyNewModel saved out to the network drive W:\

- **Project Name**

  All Project names should be representative of the high-level subject area that's being modelled.

- **Data Source Name**

  The system administrator will have created the SAP Business Warehouse data source for the developers prior to the start of the project. The naming convention for SAP Business Warehouse data will be kept generic in order to lower the risk of confusion when migrating from Test to Production.

  For example, if a SAP Business Warehouse data source was named after the source system SAPBWDEV, this name would be associated to all the IBM Cognos 8 objects right through the application stack. Once the development reports are moved to production, it would appear as if the production server is still running against the development data source. By keeping a generic name of "***SAPBW***", the confusion is kept to a minimum.

- **Parent Namespace**
  The Parent Namespace should follow a convention that is representative of the entire subject matter. For example, if the data being modeled is Automobile data the Parent Namespace should be named Automobile or something easily identifiable as related to the subject matter of the Meta-data model.

4 IBM Cognos 8 Report Studio

The following section outlines how to leverage the SAP Business Warehouse objects for reporting in IBM Cognos 8 Report Studio, and illustrates proven methods within IBM Cognos 8 Report Studio that will help to ensure a successful implementation of IBM Cognos 8 with the SAP Business Warehouse.

The following sections will provide general rules to remember when building all IBM Cognos 8 with the SAP Business Warehouse reports, and then move into details for functionality and performance for prompt and report building.

4.1 General Overview

Before moving into specific details for building IBM Cognos 8 Report Studio Reports, the following are rules of thumb to apply to all IBM Cognos 8 for use with the SAP Business Warehouse reports.

1. SAP Business Warehouse is an OLAP source which is fundamentally different than a relational source and even other OLAP data sources, ie, IBM Cognos 8 PowerPlay and MSSQL Analysis Services. This means OLAP data sources have multiple axis (ie many dimensions) whereas relational data sources are two dimensional. SQL language is used to select data from a relational source whereas MDX is used to read data from an OLAP source. Hence when building crosstab reports on an OLAP source, MDX language is passed to the source which delivers all possible combinations of data intersections.

2. Leverage the BEx Queries. In the relational world, these equate to a database view where additional business value is built in, such as calculations, filtering, etc.

3. Push filtering to the SAP Business Warehouse server as much as possible to deliver a focused result back to the IBM Cognos 8 server. This includes using BEx variables and filtering in the BEx Query.

4. It is recommended to use the level object to populate the field in a report, except in those cases where the key is a better match for formatting (no leading zeroes, slash characters, etc).

5. For comparison purposes, it is important to note that SAP BEx tools behave differently than IBM Cognos 8. SAP uses RFC calls (which are more SQL like) not MDX calls and therefore are not subject necessarily to the same OLAP constraints that IBM Cognos 8 faces using MDX.

4.2 Overview of Configuration settings that apply to this studio.

Improving performance for reporting from SAP Business Warehouse can be achieved by modifying configuration settings as outlined in other sections of this document. There are several configuration changes which can positively impact the performance of IBM Cognos 8 data reports, however the parameters noted below are key ones specific to SAP Business Warehouse and Report Studio.

Configuration settings for overall report performance tuning:

- UseMDXToRetrieveMembersLimit
- UseMDXToRetrieveMembersFor

Configuration settings for prompt performance tuning:

- UseFastGetMembers
- UseFastGetMembersFor
- UseSAPOrdinalsForMembers

The parameters noted in this section are defined more explicitly in the configuration settings that were outlined in the previous section.

4.3 Prompt Overview

4.3.1 General Guidelines

When developing reports against SAP Business Warehouse, there are some general guidelines that you should remember when considering the use of prompts in your IBM Cognos 8 reports.

- BEx variables are well supported and allow the SAP Business Warehouse data source to pre-filter the result set. Use of these variables is recommended as they can generate prompts which will either require optional or mandatory input from the IBM Cognos 8 user.
- IBM Cognos 8 will automatically attempt to generate an appropriate prompt based on the metadata captured from SAP Business Warehouse. However if a different prompt style is required, then it is possible to customize the prompt page.
- For positional and layout changes, you can use the "generated prompt" prompt type, or you may use a different prompt type and have that populate the BEx variable. In this way you can utilize BEx variables without sacrificing the flexibility of a custom IBM Cognos 8 prompt page.

- Report authors can build and use a IBM Cognos 8 prompt to get answers in the event the BEx variable does not exist, instead of waiting until a new variable is available in SAP Business Warehouse/BEx.
- IBM Cognos 8 prompts can also be used for hierarchy selection and to emulate BEx selection option variables.

The sections below will illustrate prompt techniques for the following scenarios which are pertinent to SAP Business Warehouse:

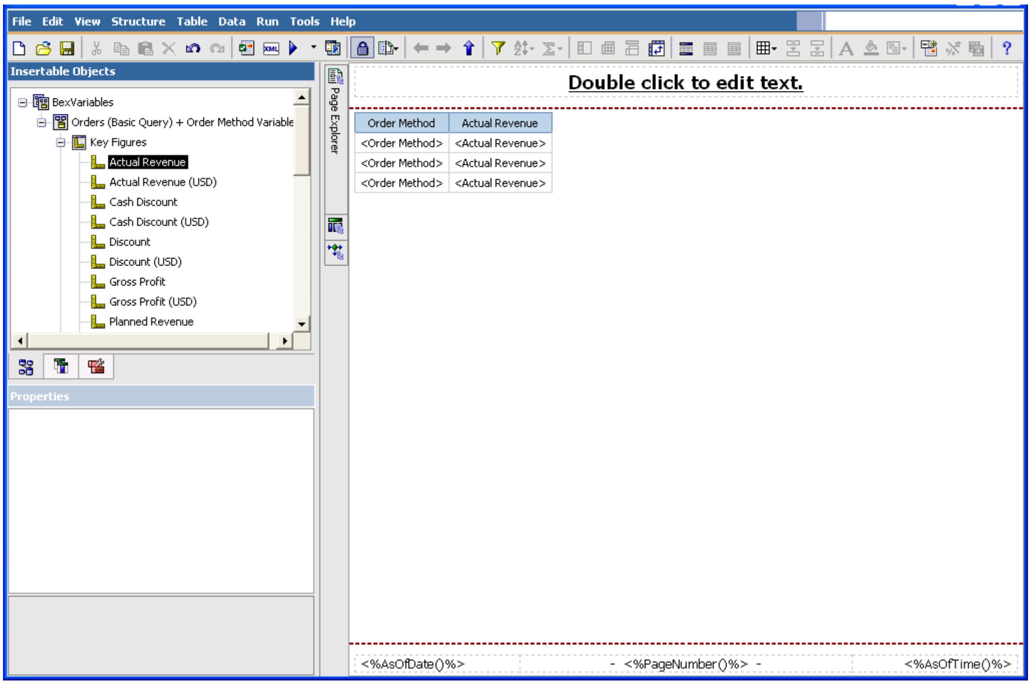
- Using BEx Variables without associating them with a Cognos 8 prompt
- Creating a Cognos 8 prompt page to format BEx variables
- IBM Cognos 8 search and select prompt
- Preselected values to Populate a Prompt Query
- Detect BEx variable Uniqueness

4.3.2 Using BEx Variables without associating them with an IBM Cognos 8 Prompt

When a mandatory BEx variable is defined in the SAP Business Warehouse query, you do not always have to include the variable within an IBM Cognos 8 prompt. This is because IBM Cognos 8 will automatically attempt to generate an appropriate prompt based on the metadata captured from SAP Business Warehouse.

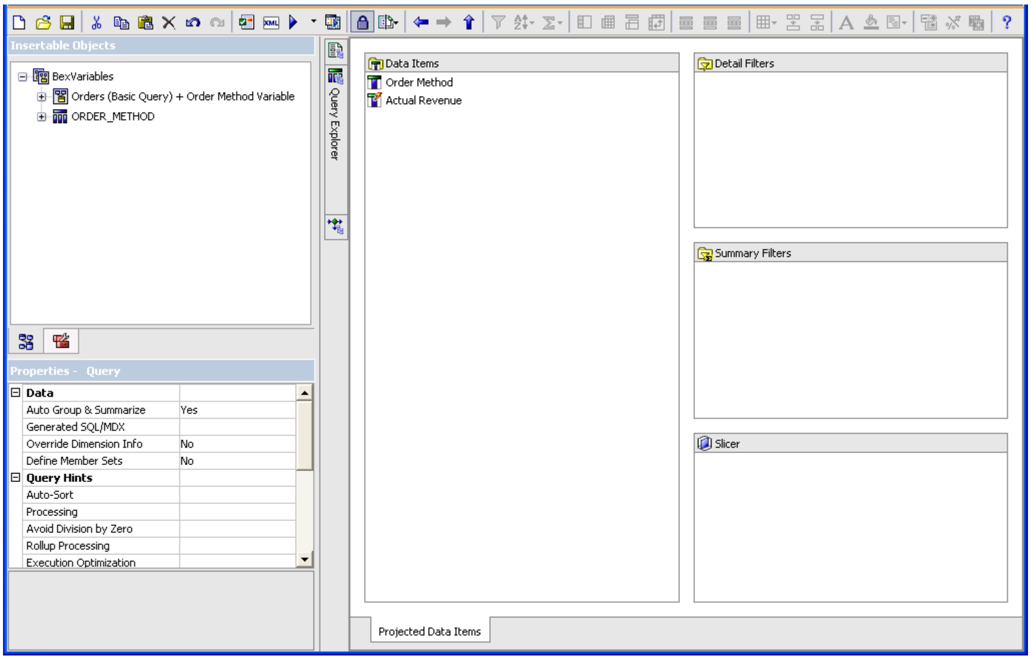
The steps below illustrate how an IBM Cognos 8 report will behave if the BEx mandatory variable is not explicitly defined within an IBM Cognos 8 prompt.

1. Create the report normally as you would any Report Studio report.



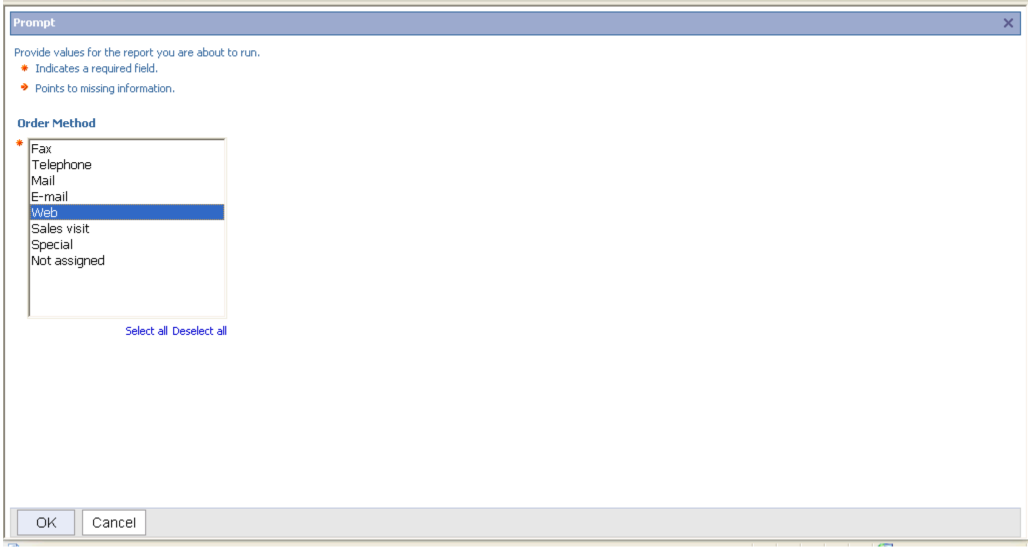
Order Method	Actual Revenue
<Order Method>	<Actual Revenue>
<Order Method>	<Actual Revenue>
<Order Method>	<Actual Revenue>

2. Notice that the report query does not contain a filter. However, when running the query, the user is still prompted for a value and the report result contains only the information which satisfies the BEx variable condition. This is because the SAP Business Warehouse mandatory variable was included within the SAP Business Warehouse query, and in turn is defined in the IBM Cognos 8 metadata model that was used in this IBM Cognos 8 package.

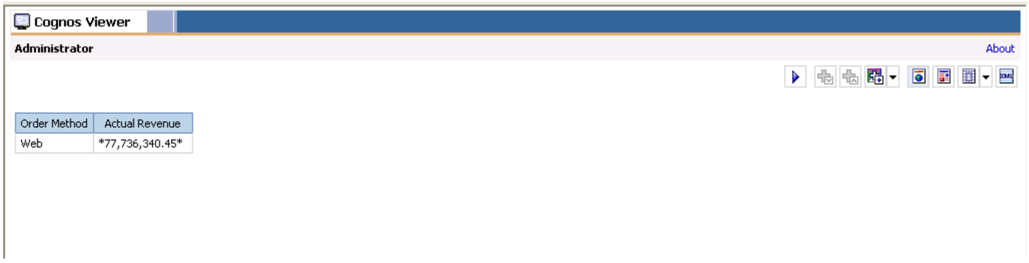


Properties - Query

Data	
Auto Group & Summarize	Yes
Generated SQL/MDX	
Override Dimension Info	No
Define Member Sets	No
Query Hints	
Auto-Sort	
Processing	
Avoid Division by Zero	
Rollup Processing	
Execution Optimization	



Provide values for the report you are about to run.
* Indicates a required field.
Points to missing information.



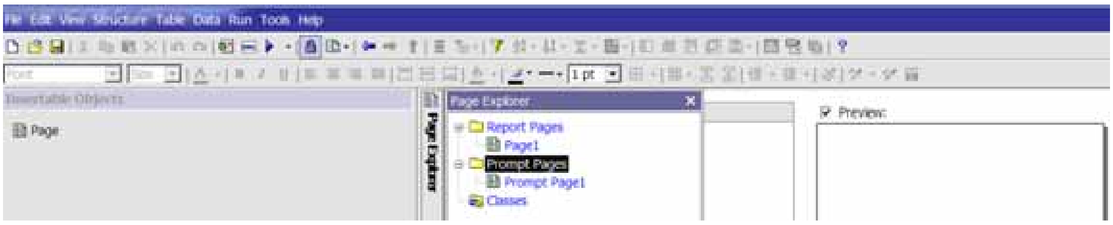
Order Method	Actual Revenue
Web	*77,736,340.45*

4.3.3 Creating the Prompt Page to format BEx variables

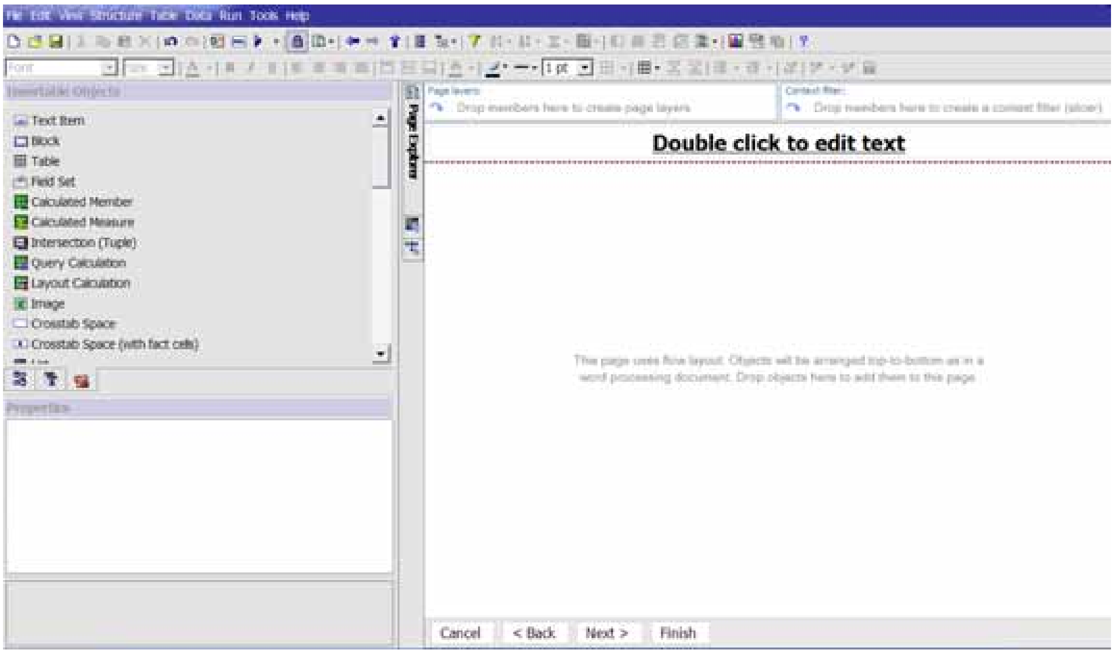
Although IBM Cognos 8 has the ability to automatically generate appropriate prompts based on the SAP BEx variable, a report builder does have the ability to customize the prompt for additional formatting.

The following steps enable additional formatting of BEx variables in IBM Cognos 8 reports.

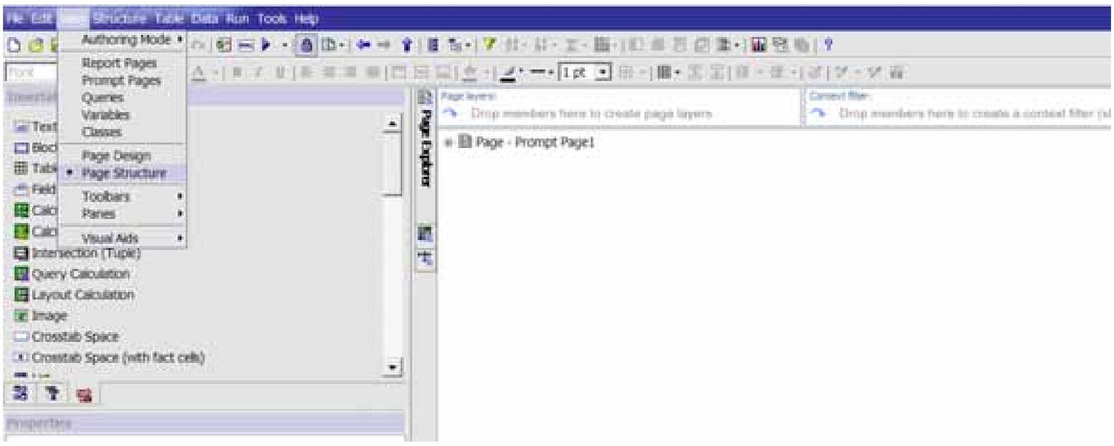
1. Using the Page explorer, create a new prompt page.



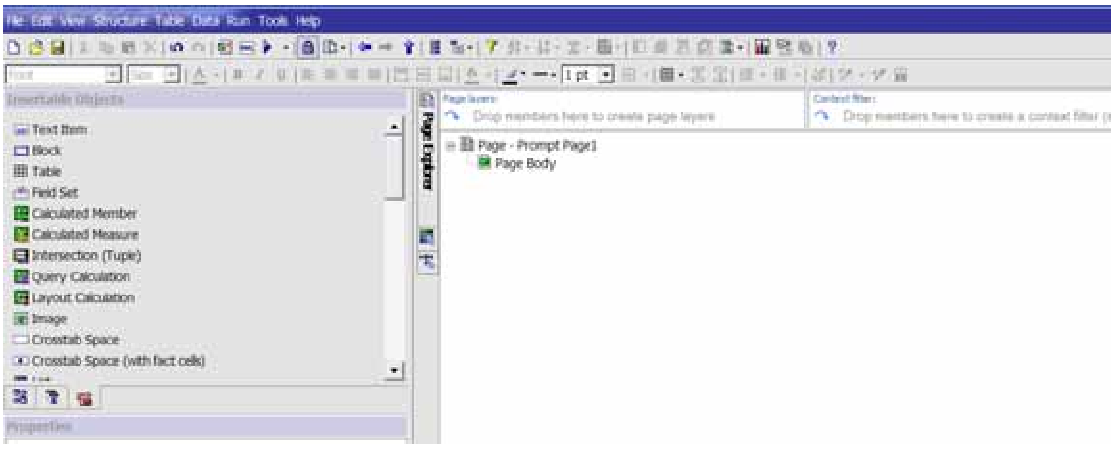
2. Within the Page Explorer, double click on Prompt Page1. The following view should be displayed.


File Edit View Structure Table Data Run Tools Help
Insertable Objects
Text Item
Block
Table
Field Set
Calculated Member
Calculated Measure
Intersection (Tuple)
Query Calculation
Layout Calculation
Image
Crosstab Space
Crosstab Space (with fact cells)
Properties
Page Explorer
Page layers:
Drop members here to create page layers
Context filter:
Drop members here to create a context filter (slicer)
Double click to edit text
The page uses flow layout. Objects will be arranged top-to-bottom as in a word processing document. Drop objects here to add them to this page.
Cancel
< Back
Next >
Finish

3. Using the View menu command, switch from Page Design to Page Structure.



4. Delete all the objects, except for the Page Body. Once the objects have been removed, the screen should display as follows.

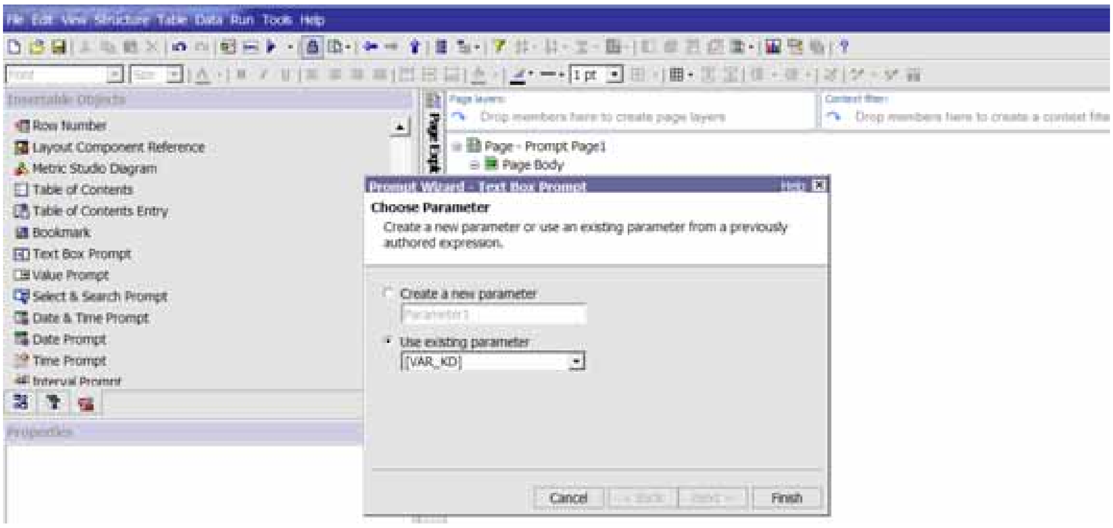


5. From the Insertable Objects pane, locate the Block object. Then drag this object under the Page Body.

6. From the Insertable Objects pane, locate the Text Box Prompt and drag it on top of the Block object.

7. When presented with the Prompt Wizard dialog box. Select the "Use existing parameter" radio button.

8. From the list of available parameters, select the SAP BEx variable desired. For this example, the parameter will be [VAR_KD].



Prompt Wizard - Text Box Prompt

Choose Parameter

Create a new parameter or use an existing parameter from a previously authored expression.

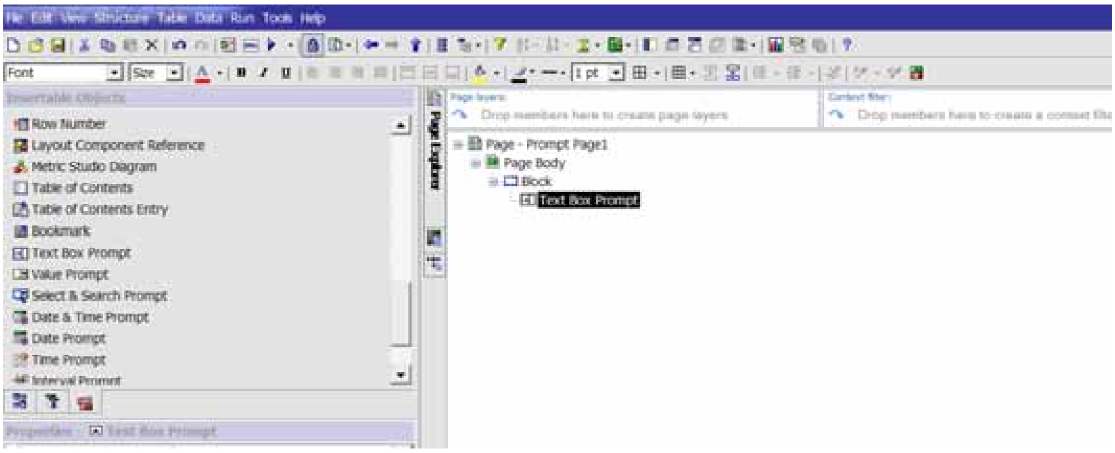
○ Create a new parameter

Parameter1

● Use existing parameter

[VAR_KD]

9. Once completed, select Finish. The Page Structure view should look like the following.



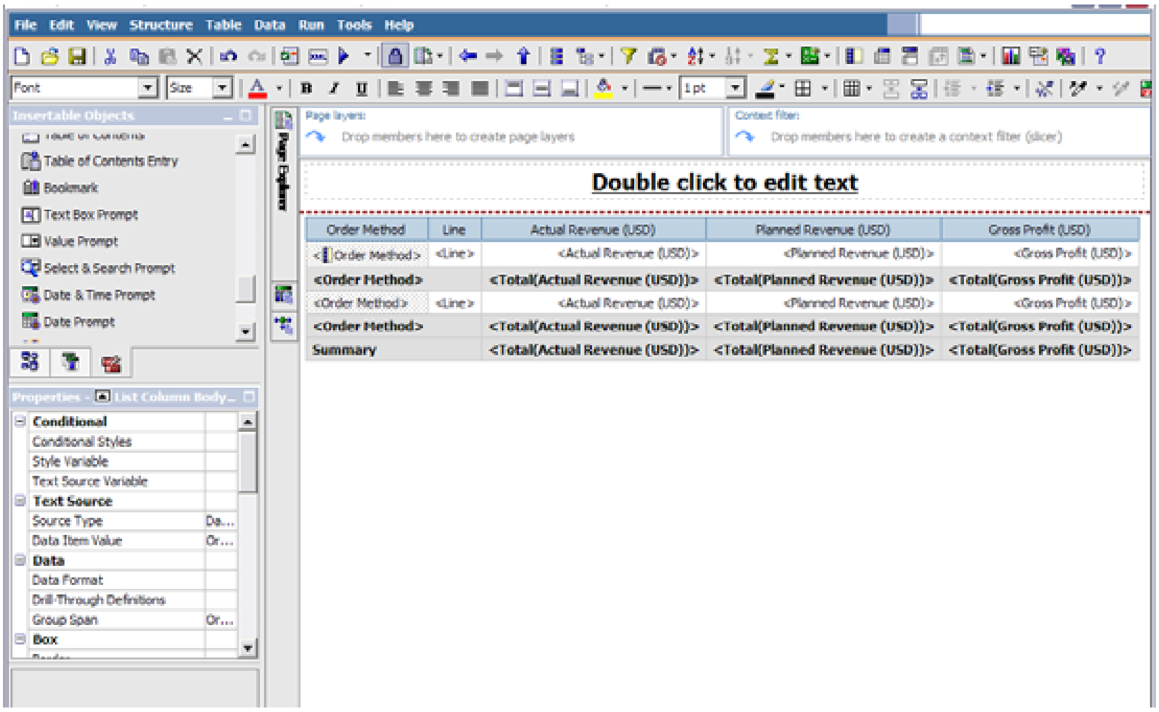
10. If the report is executed now, the prompt page will appear with the appropriate SAP Business Warehouse variable prompt in the prompt page. Text boxes, etc, may be added to the prompt page for more formatting.

4.3.4 IBM Cognos 8 Search and Select Prompt

The Selection Option BEx variable allows interval, single select, multi select, and search/select prompting scenarios. The Selection Option BEx variable is imported as a type of complex in IBM Cognos 8 Framework Manager. When displayed in an IBM Cognos 8 report, it appears as a single select picklist.

To emulate the Selection Option BEx variable in IBM Cognos 8, multiple optional BEx variables can be used for the characteristic for the required prompt types, ie, interval, single select, and multi select options on an IBM Cognos 8 prompt page.

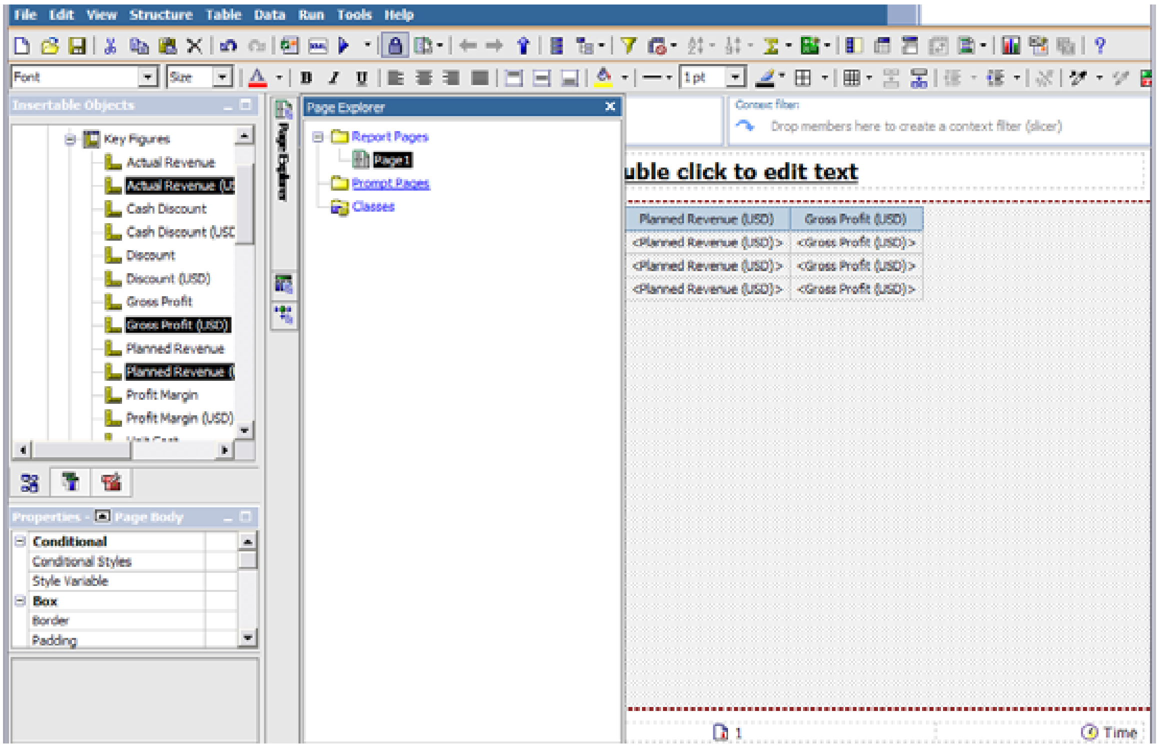
The following example of a Search & Select Prompt leverages a multi-value BEx variable. The report has been built, and now we need to build the prompt for the report below.



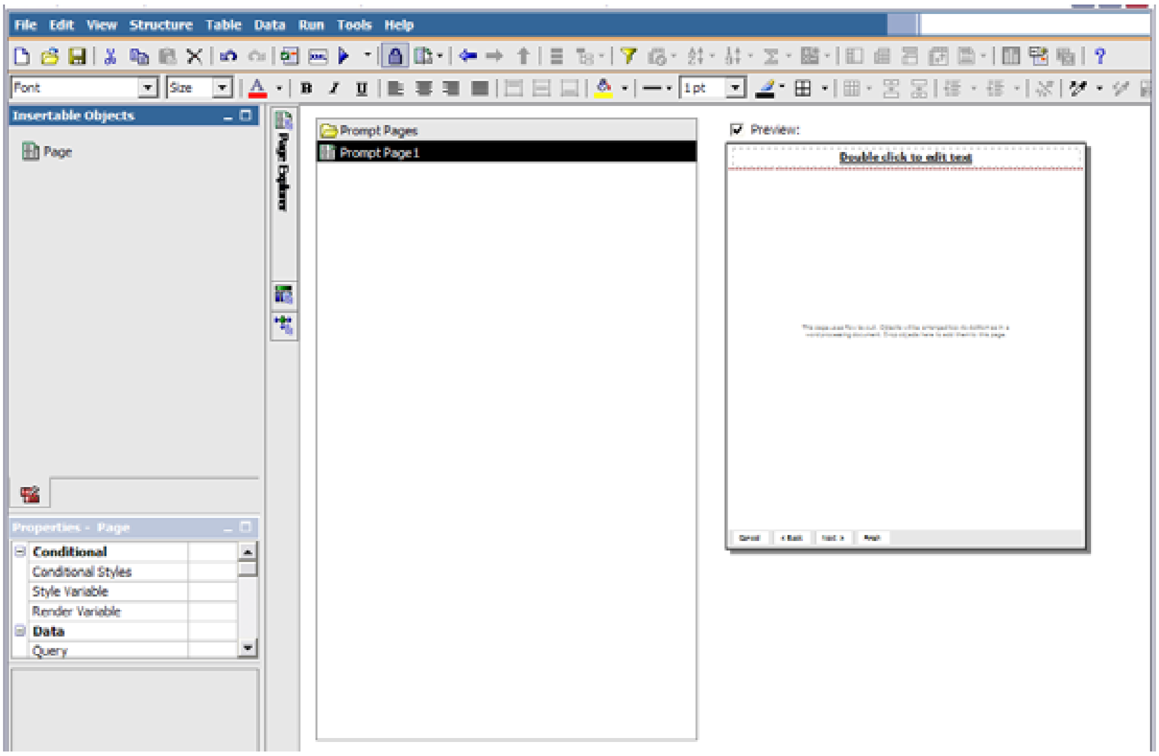
Conditional	
Conditional Styles	
Style Variable	
Text Source Variable	
Text Source	
Source Type	Da...
Data Item Value	Or...
Data	
Data Format	
Drill-Through Definitions	
Group Span	Or...
Box	

Order Method	Line	Actual Revenue (USD)	Planned Revenue (USD)	Gross Profit (USD)
<Order Method>	<Line>	<Actual Revenue (USD)>	<Planned Revenue (USD)>	<Gross Profit (USD)>
<Order Method>		**<Total(Actual Revenue (USD))>**	**<Total(Planned Revenue (USD))>**	**<Total(Gross Profit (USD))>**
<Order Method>	<Line>	<Actual Revenue (USD)>	<Planned Revenue (USD)>	<Gross Profit (USD)>
<Order Method>		**<Total(Actual Revenue (USD))>**	**<Total(Planned Revenue (USD))>**	**<Total(Gross Profit (USD))>**
Summary		**<Total(Actual Revenue (USD))>**	**<Total(Planned Revenue (USD))>**	**<Total(Gross Profit (USD))>**

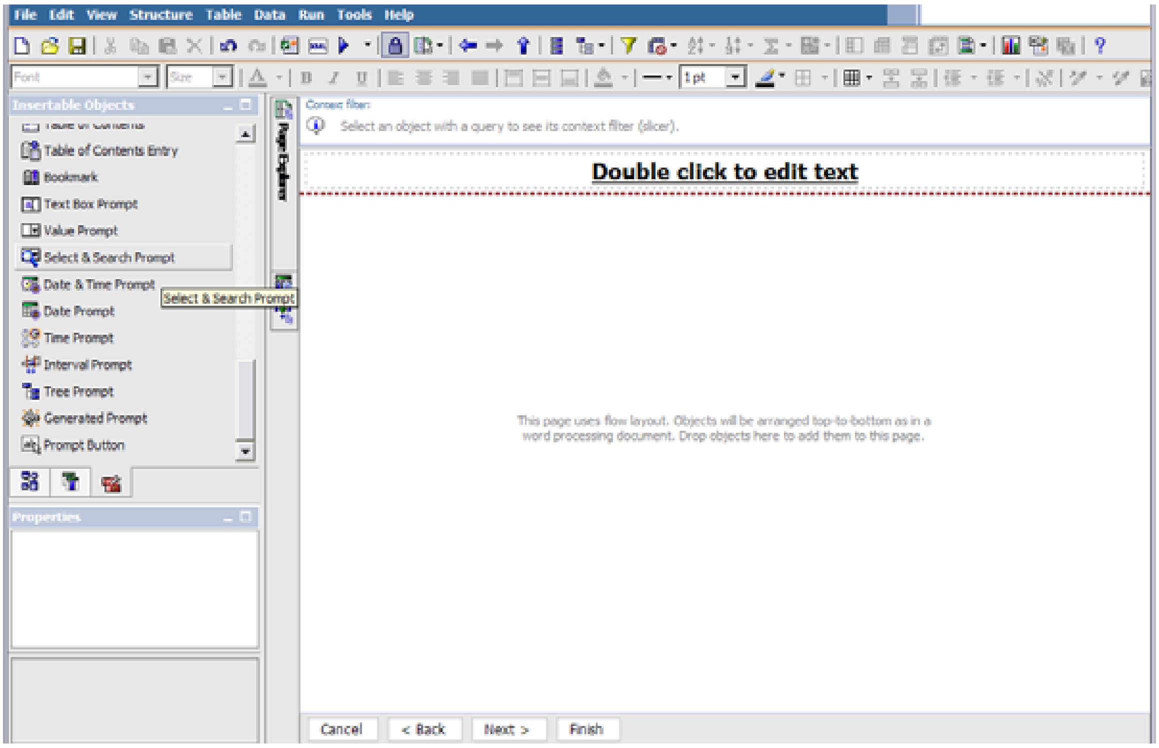
1. From Page Explorer select Prompt Pages.



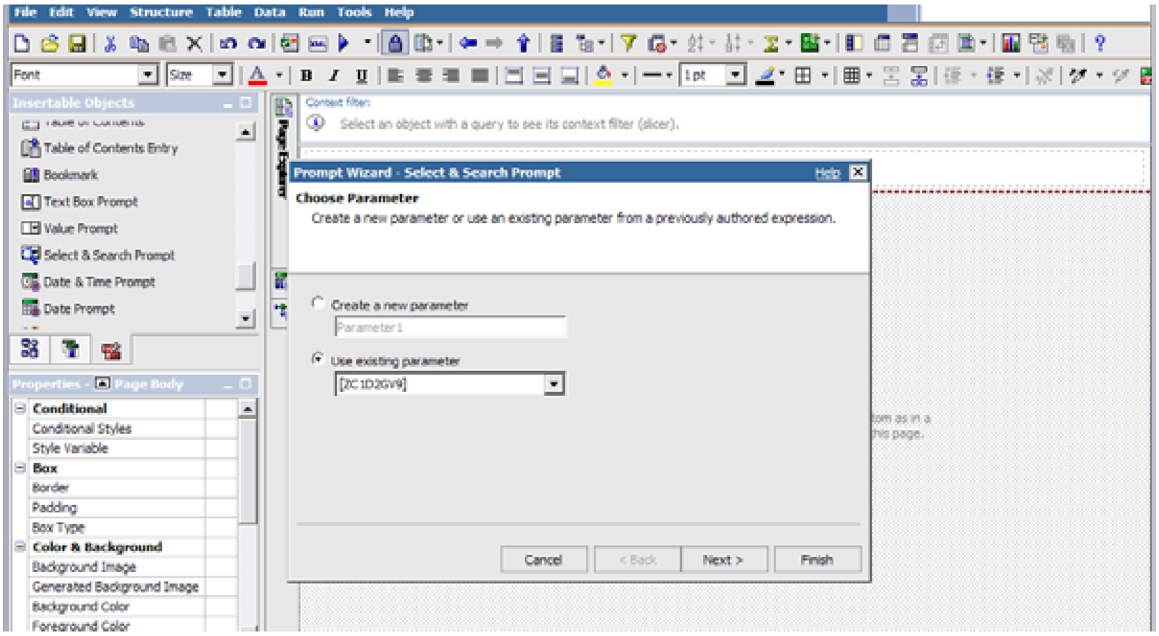
2. Drag a Page to the Prompt Pages window. Double click to open the prompt page.



3. From the Tool Box find and drag a Search & Select Prompt onto the Prompt Page.


File Edit View Structure Table Data Run Tools Help
Insertable Objects
Table of Contents Entry
Bookmark
Text Box Prompt
Value Prompt
Select & Search Prompt
Date & Time Prompt
Date Prompt
Time Prompt
Interval Prompt
Tree Prompt
Generated Prompt
Prompt Button
Select & Search Prompt
Properties
Page Explorer
Context filter:
Select an object with a query to see its context filter (slicer).
Double click to edit text
This page uses flow layout. Objects will be arranged top-to-bottom as in a word processing document. Drop objects here to add them to this page.
Cancel
< Back
Next >
Finish

4. The Prompt Wizard first asks for the parameter/BEx variable to use. Click Next.


Prompt Wizard - Select & Search Prompt

Choose Parameter

Create a new parameter or use an existing parameter from a previously authored expression.

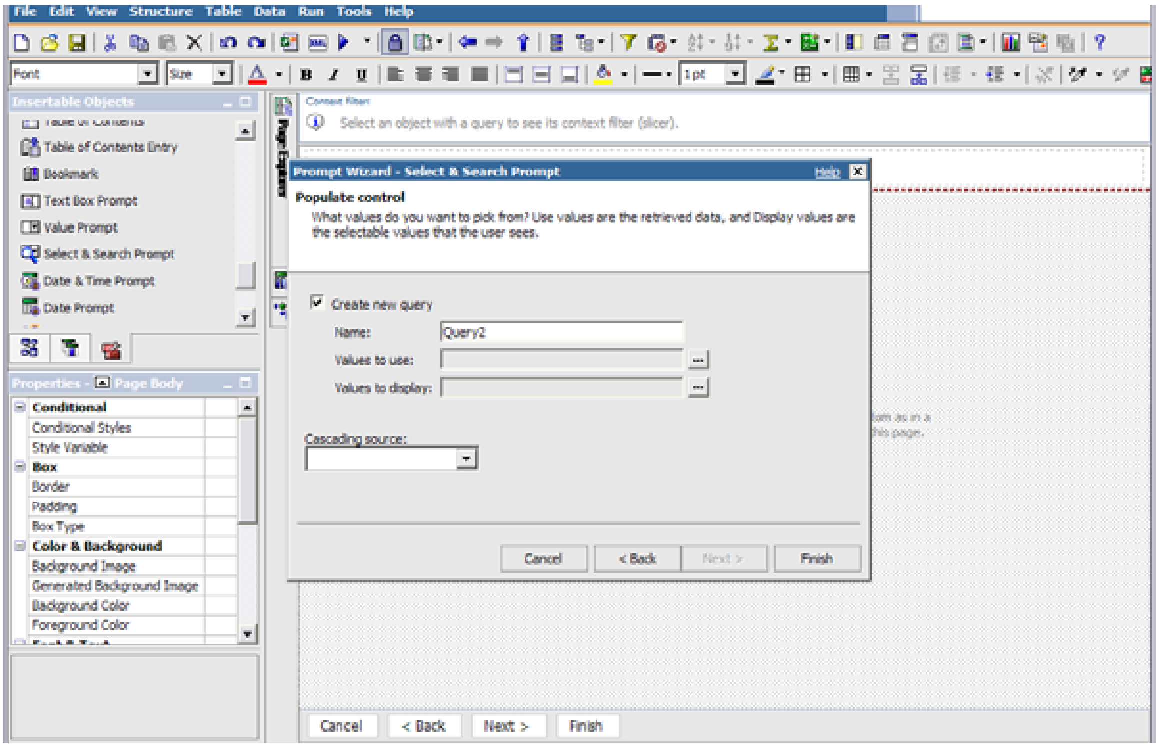
○ Create a new parameter

Parameter1

● Use existing parameter

[ZC1D2GV9]

5. Next, create a query to populate values to search against then to be passed to the multi-value BEx variable.



Prompt Wizard - Select & Search Prompt

Populate control

What values do you want to pick from? Use values are the retrieved data, and Display values are the selectable values that the user sees.

☑ Create new query

Name: Query2

Values to use:

Values to display:

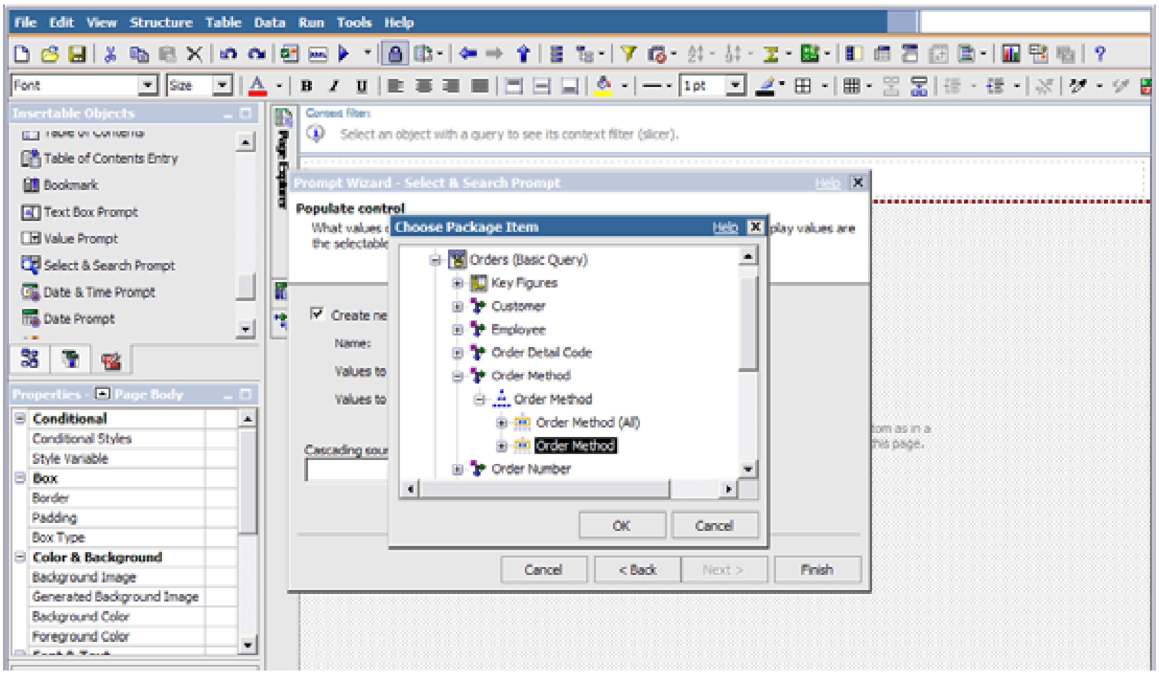
Cascading source:

6. Use a BEx query that contains no BEx variables for the new query. In this case, the BEx variable is for Order Method.

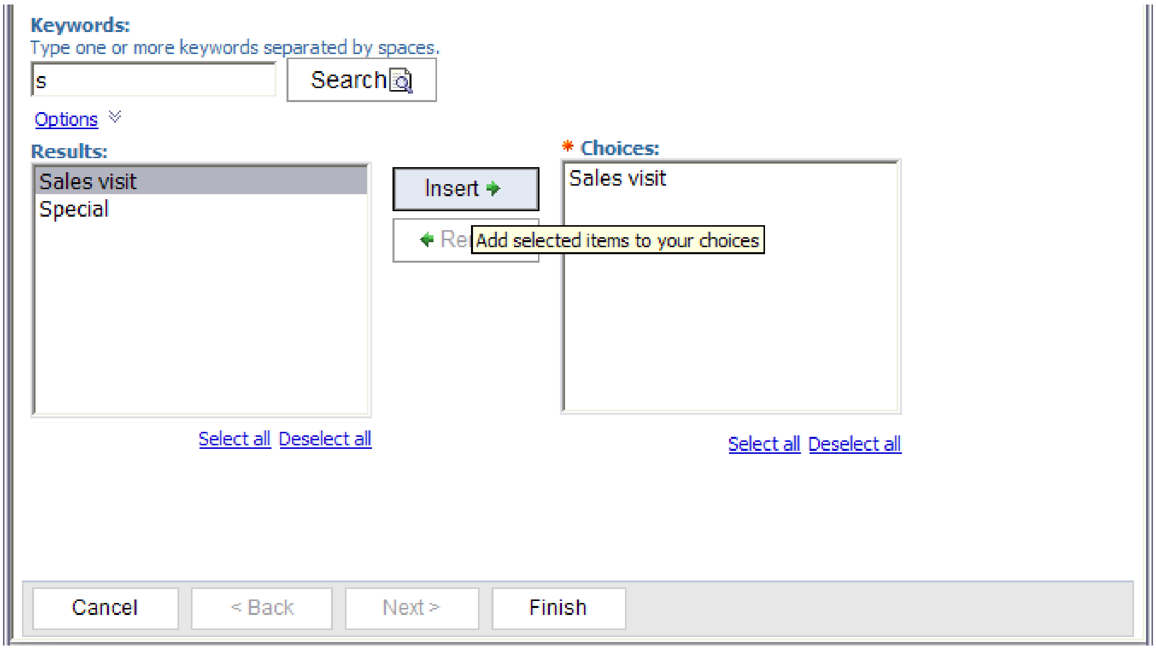
7. For the Value to Use property, expand the BEx query with no BEx variables and select Order Method. Then click OK.

8. For Values to Display property, click Order Method again. Note: Depending on how the BEx variable was created, if it only uses key values and you need to search on descriptions then the use value would be the business key and the description would be the level.

9. Click OK, then click Finish.



10. Execute report and the prompt page displays with the Search & Select Prompt using a multi-value BEx variable. First, searching on strings containing “s”, Sales visit is selected and inserted into the Choices box.


Keywords:
Type one or more keywords separated by spaces.

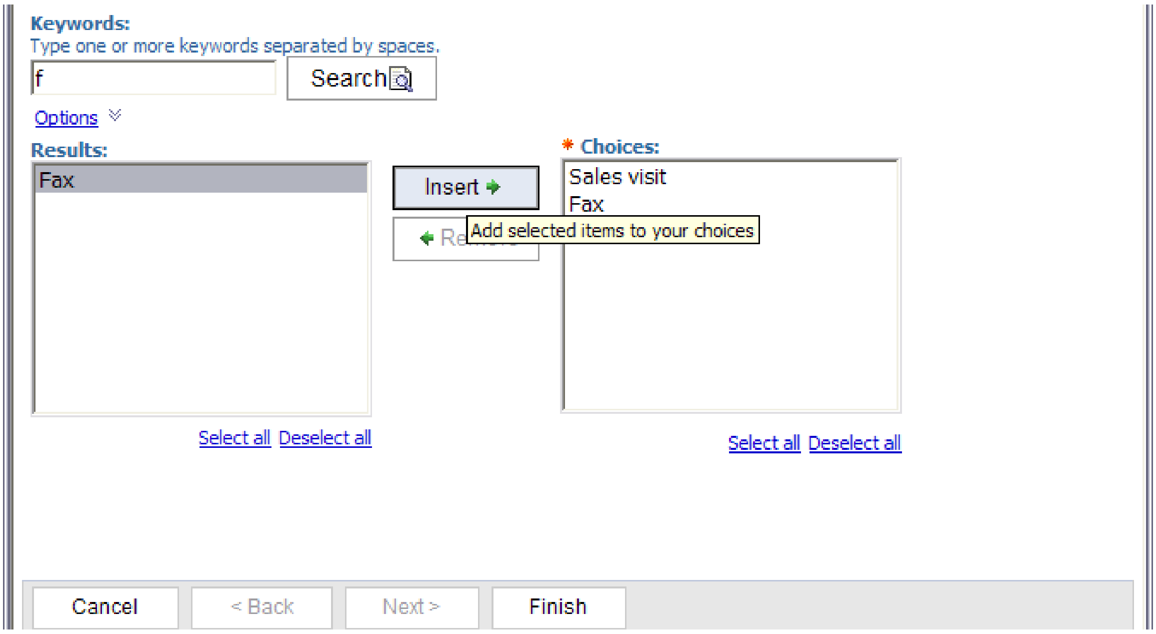
s

Results:
Sales visit
Special

Choices:
Sales visit

Add selected items to your choices

11. Next searching on “f”, Fax is returned, selected and added to Choice box.



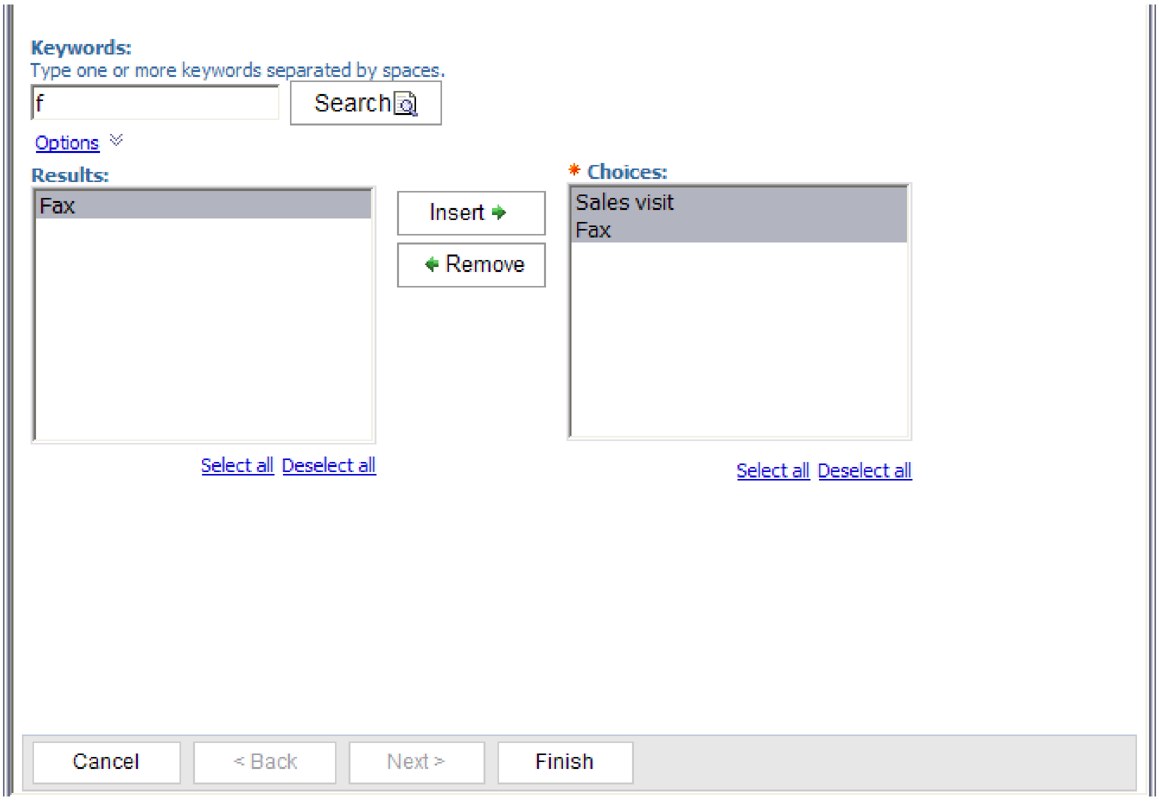
Keywords:
Type one or more keywords separated by spaces.
f

Results:
Fax

Choices:
Sales visit
Fax

Add selected items to your choices

12. Select all items in the Choices box to filter on then click on Finish.

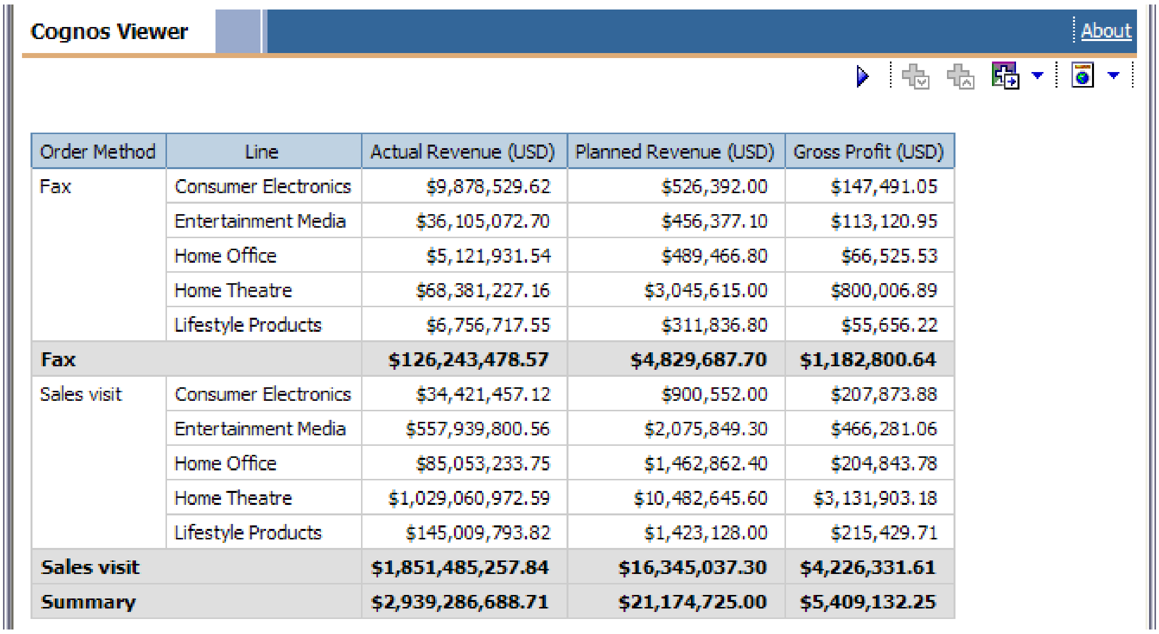


Keywords:
Type one or more keywords separated by spaces.
f

Results:
Fax

Choices:
Sales visit
Fax

13. The following is report filtered on the selected values:



Order Method	Line	Actual Revenue (USD)	Planned Revenue (USD)	Gross Profit (USD)
Fax	Consumer Electronics	$9,878,529.62	$526,392.00	$147,491.05
	Entertainment Media	$36,105,072.70	$456,377.10	$113,120.95
	Home Office	$5,121,931.54	$489,466.80	$66,525.53
	Home Theatre	$68,381,227.16	$3,045,615.00	$800,006.89
	Lifestyle Products	$6,756,717.55	$311,836.80	$55,656.22
Fax		**$126,243,478.57**	**$4,829,687.70**	**$1,182,800.64**
Sales visit	Consumer Electronics	$34,421,457.12	$900,552.00	$207,873.88
	Entertainment Media	$557,939,800.56	$2,075,849.30	$466,281.06
	Home Office	$85,053,233.75	$1,462,862.40	$204,843.78
	Home Theatre	$1,029,060,972.59	$10,482,645.60	$3,131,903.18
	Lifestyle Products	$145,009,793.82	$1,423,128.00	$215,429.71
Sales visit		**$1,851,485,257.84**	**$16,345,037.30**	**$4,226,331.61**
Summary		**$2,939,286,688.71**	**$21,174,725.00**	**$5,409,132.25**

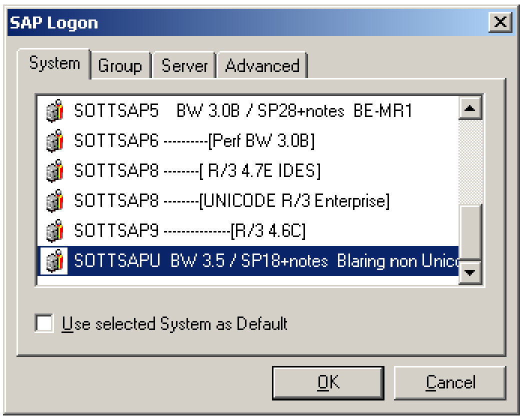
4.3.5 Pre-selected values to Populate a Prompt Query

An InfoQuery can be created and used to populate a prompt with restricted pre-selected values. These selected values will not need to be suppressed. This method could be extended to use in a cascading prompt scenario.

This restructure method can be applied to most IBM Cognos 8 Report Studio problematic prompting scenarios.

Steps to implement:

1. In IBM Cognos 8 Report Studio, identify the queries that are used to populate prompts.
2. For each of these queries, make a list of the query item being used.
3. Each of the items from the list above will be replaced by a new query item which is based on an InfoQuery whose values have been restricted.
4. To begin creating an InfoQuery with restricted values, launch the Query Designer from a computer that has BEx installed.
5. When the "*SAP Logon*" dialog box appears, select the appropriate "*SAP server*" and click the OK.

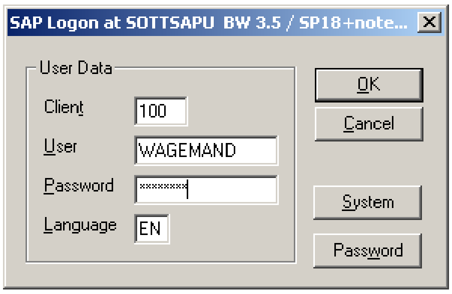


SAP Logon

SOTTSAP5 BW 3.0B / SP28+notes BE-MR1
SOTTSAP6 ---------[Perf BW 3.0B]
SOTTSAP8 -------[R/3 4.7E IDES]
SOTTSAP8 -------[UNICODE R/3 Enterprise]
SOTTSAP9 -------------[R/3 4.6C]
SOTTSAPU BW 3.5 / SP18+notes Blaring non Unic

☐ Use selected System as Default

6. Enter the correct Client, User, Password and Language setting. Once completed select OK.

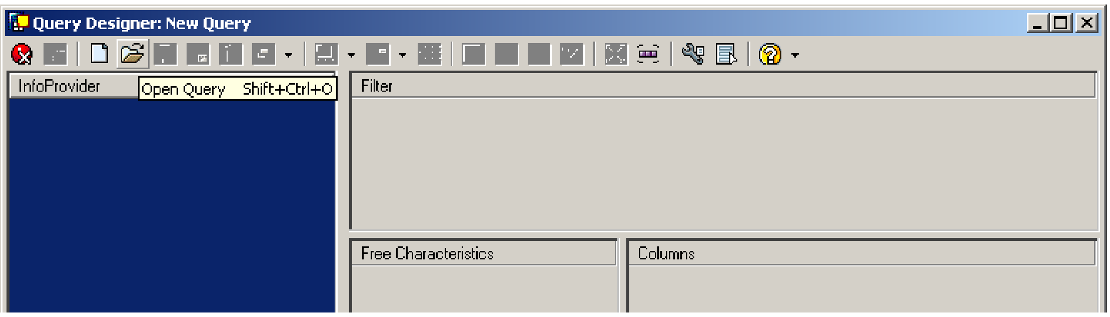


SAP Logon at SOTTSAPU BW 3.5 / SP18+note...

User Data

Client 100
User WAGEMAND
Password ********
Language EN

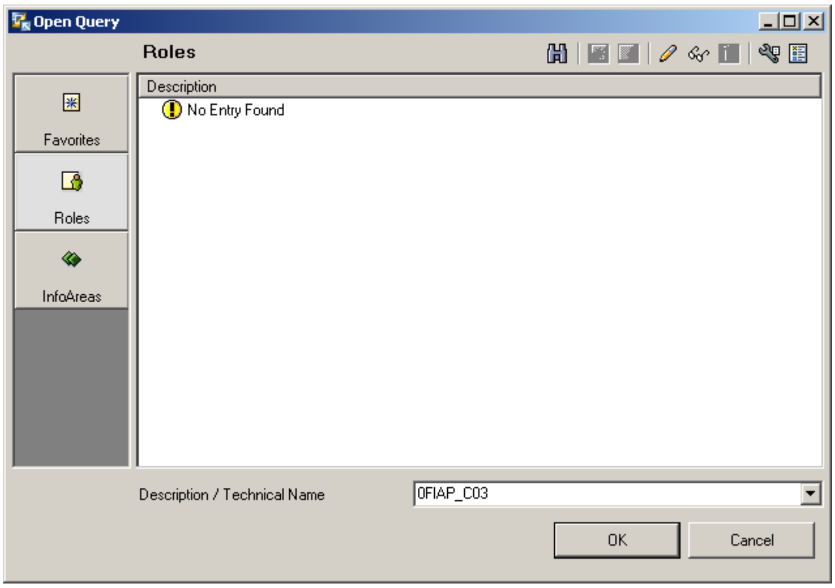
7. When the Query Designer application loads, click the folder icon located on the top toolbar.



Query Designer: New Query

Open Query Shift+Ctrl+O

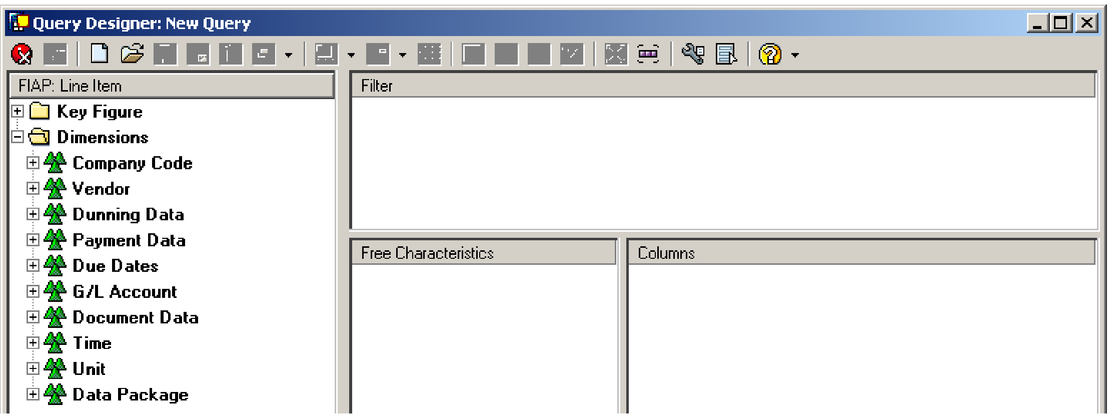
8. Enter the Description/ Technical Name of the SAP Business Warehouse object that the original query items were based on. This can be obtained from the Namespace screen tip of the original model namespace within IBM Cognos 8 Framework Manager.



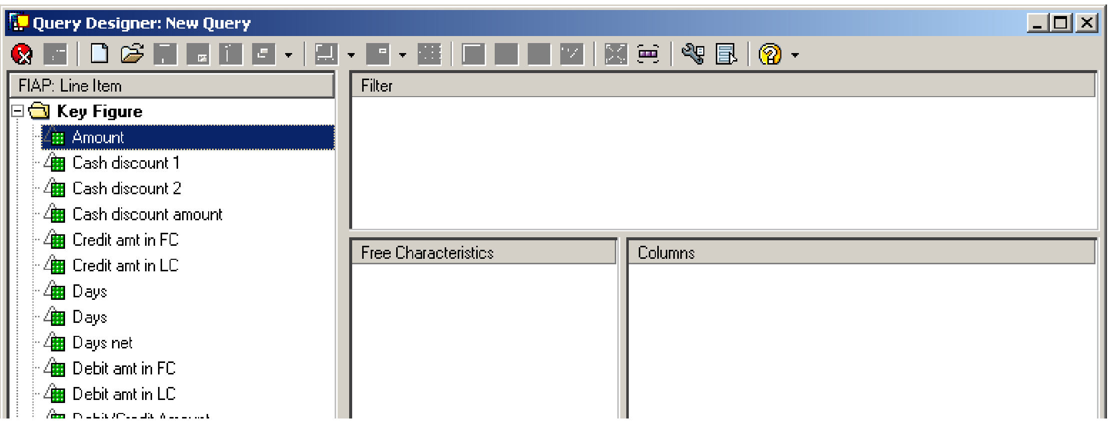
No Entry Found

Description / Technical Name: 0FIAP_C03

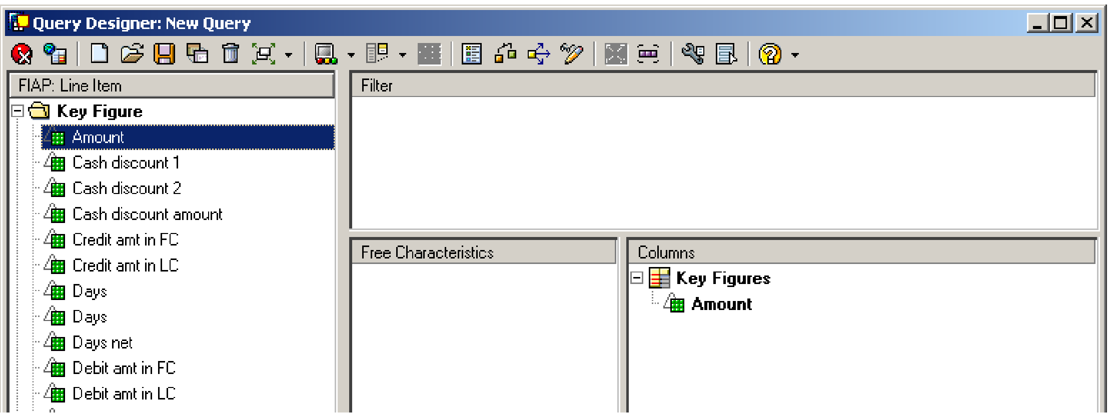
9. Press OK to display the following screen.



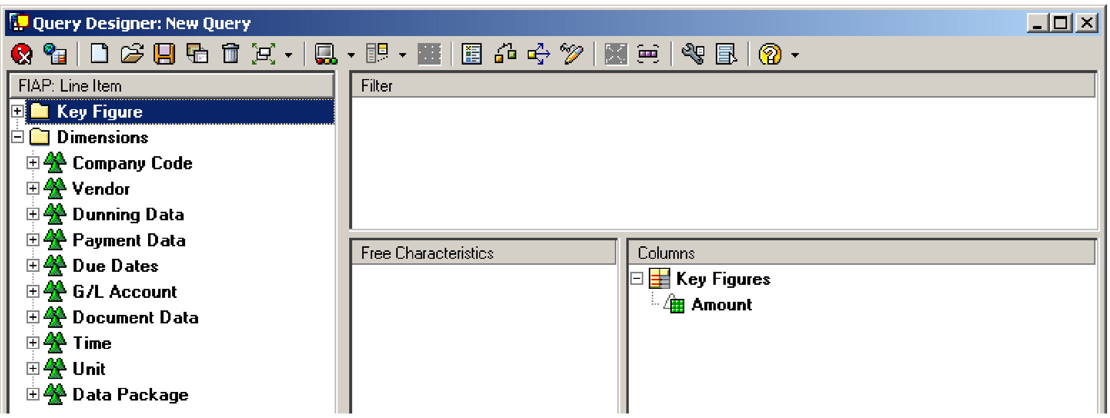
10. Expand the Key Figure folder, and select a Key Figure by clicking on it.



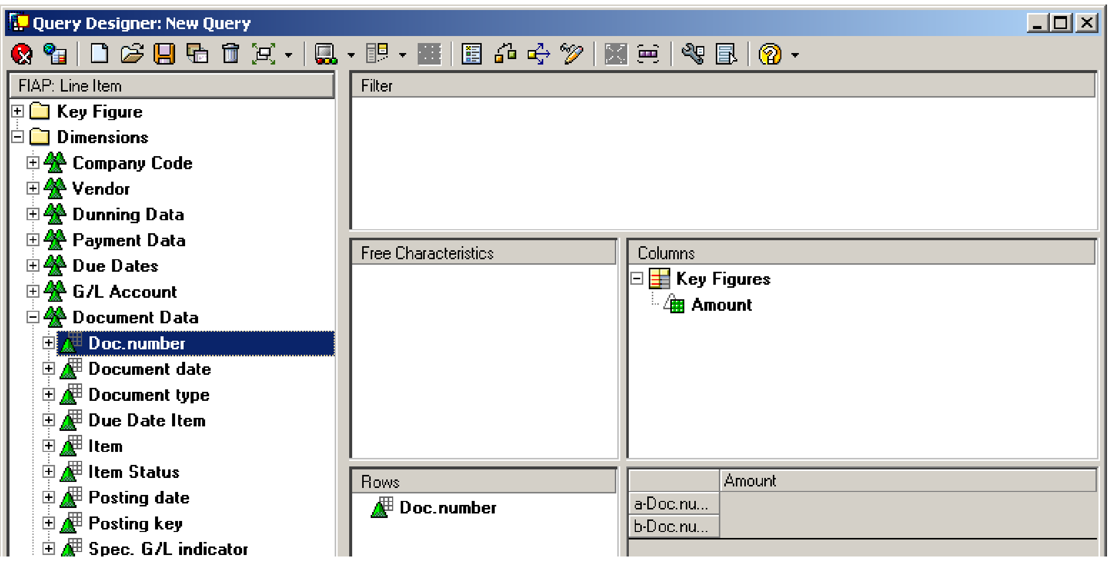
11. Drag the selected Key Figure into the Columns window.



12. Close the Key Figure folder and expand the Dimension folder.

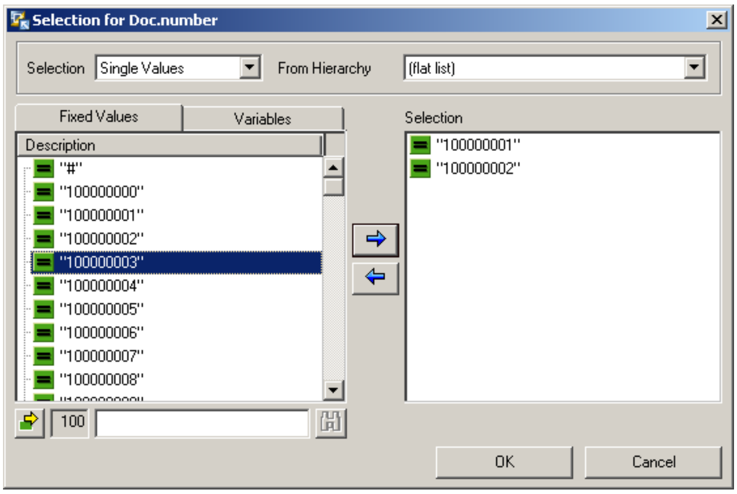


13. Locate the characteristics objects identified by the list of query items in step 2 and drag these items into the Rows window.



14. Right click the first item in the Rows window and select Restrict.

15. Select the values required to satisfy the IBM Cognos 8 Report Studio query prompt. The selected values can be moved between the left and right window by pressing the arrow buttons.

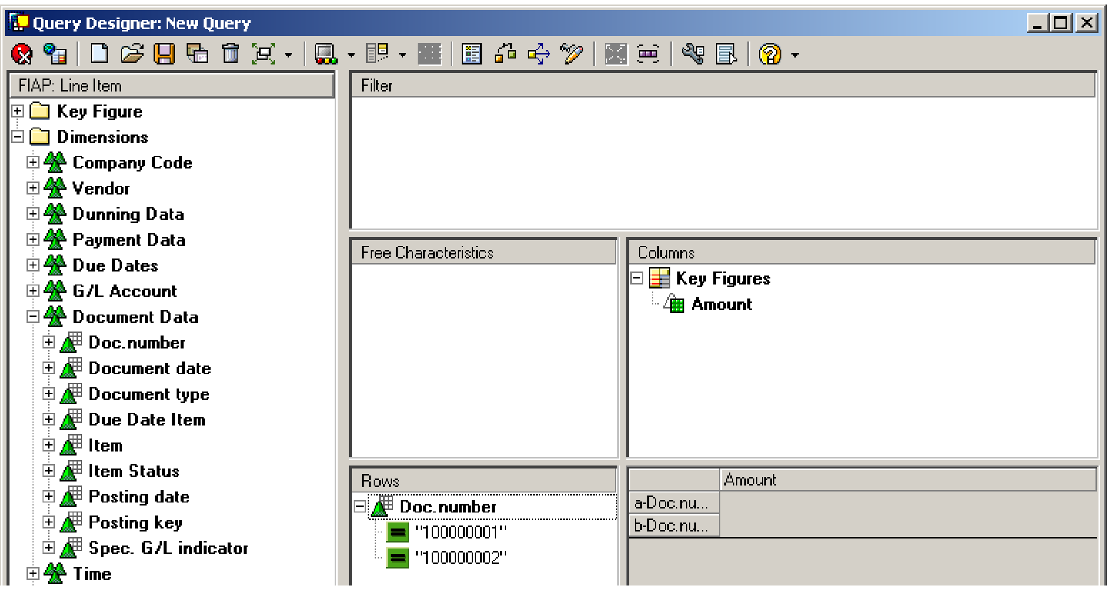


Selection for Doc.number

Selection: Single Values — From Hierarchy: (flat list)

Selection: "100000001", "100000002"

16. Once the desired selections have been made, click the OK button.



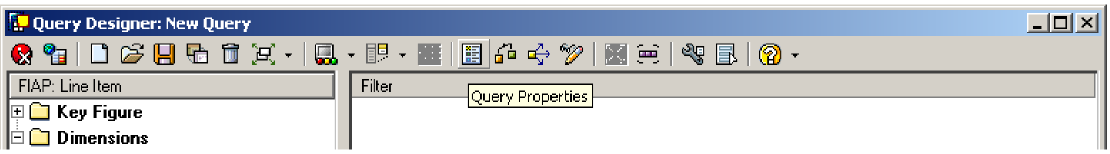
Query Designer: New Query

Rows: Doc.number — "100000001", "100000002"

Columns: Key Figures — Amount

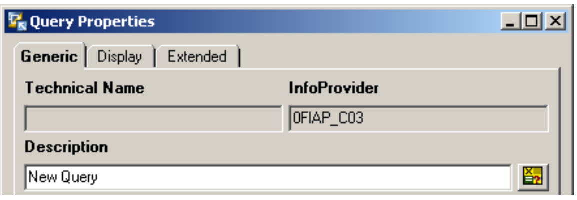
	Amount
a-Doc.nu...	
b-Doc.nu...	

17. Select the Query Properties on the top toolbar.



Query Designer: New Query

18. Click on the Extended tab

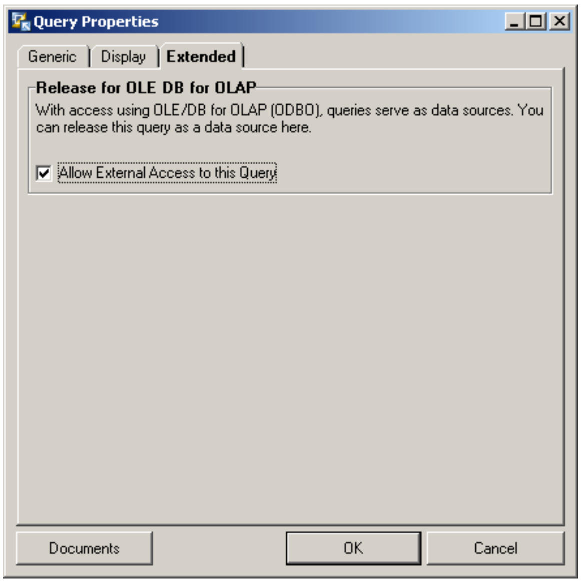


Technical Name

InfoProvider: 0FIAP_C03

Description: New Query

19. Click the select box to “Allow External Access to this Query” and click the OK button.

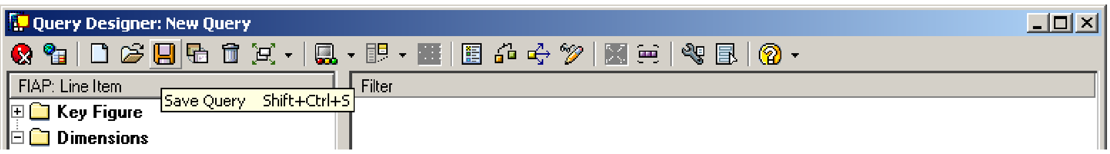


Release for OLE DB for OLAP

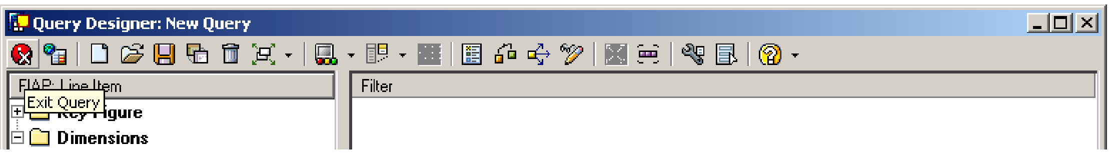
With access using OLE/DB for OLAP (ODBO), queries serve as data sources. You can release this query as a data source here.

☑ Allow External Access to this Query

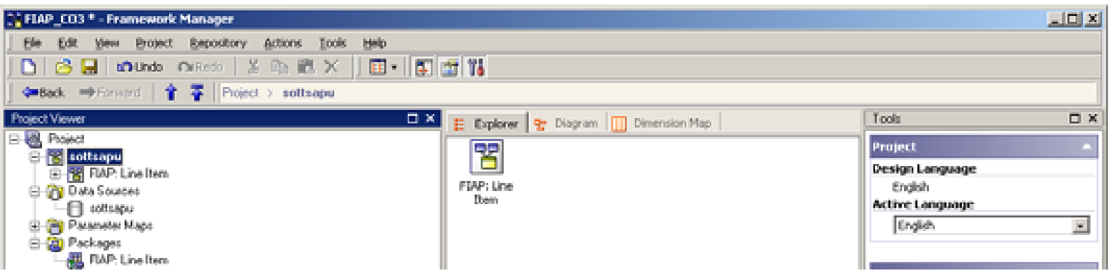
20. Select the save icon on the top toolbar to save this query. When prompted give the query an appropriate technical name and description.



21. Exit the query from the top toolbar.

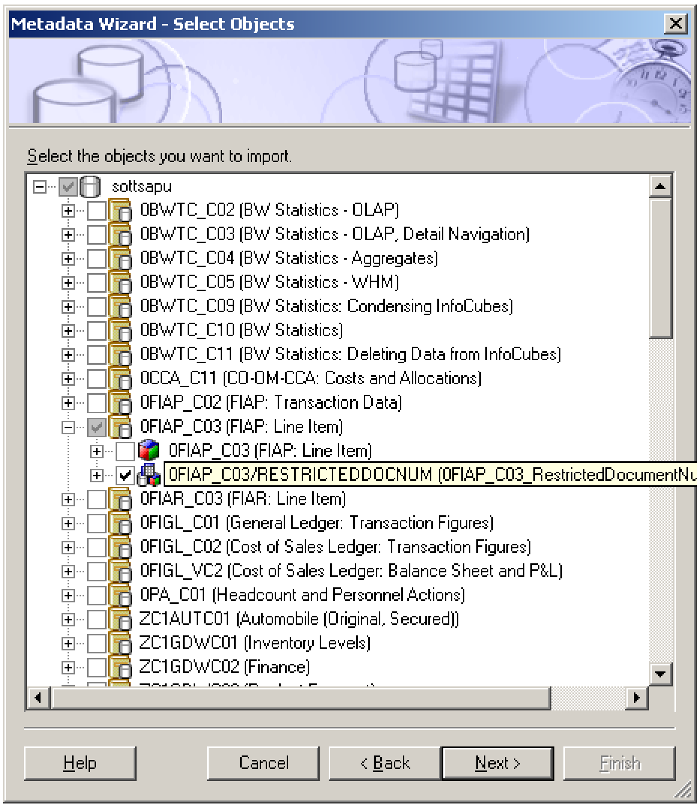


22. Launch IBM Cognos 8 Framework Manager and open the original model used for the existing reports.



23. Right click on the top level namespace and select ‘Run Metadata Wizard”. Traverse the screens, selecting the correct data source.

24. The previously created InfoQuery should be visible in the Select Objects Import window.

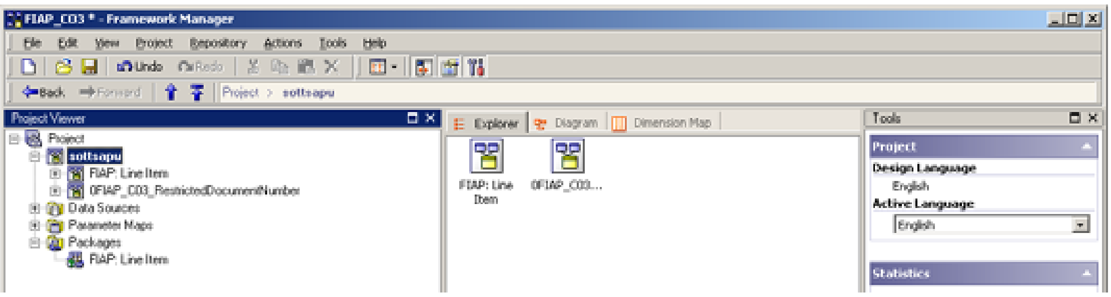


Metadata Wizard - Select Objects

Select the objects you want to import.

- ☑ soltsapu
  - □ 0BWTC_C02 (BW Statistics - OLAP)
  - □ 0BWTC_C03 (BW Statistics - OLAP, Detail Navigation)
  - □ 0BWTC_C04 (BW Statistics - Aggregates)
  - □ 0BWTC_C05 (BW Statistics - WHM)
  - □ 0BWTC_C09 (BW Statistics: Condensing InfoCubes)
  - □ 0BWTC_C10 (BW Statistics)
  - □ 0BWTC_C11 (BW Statistics: Deleting Data from InfoCubes)
  - □ 0CCA_C11 (CO-OM-CCA: Costs and Allocations)
  - □ 0FIAP_C02 (FIAP: Transaction Data)
  - ☑ 0FIAP_C03 (FIAP: Line Item)
    - □ 0FIAP_C03 (FIAP: Line Item)
    - ☑ 0FIAP_C03/RESTRICTEDDOCNUM (0FIAP_C03_RestrictedDocumentNu
  - □ 0FIAR_C03 (FIAR: Line Item)
  - □ 0FIGL_C01 (General Ledger: Transaction Figures)
  - □ 0FIGL_C02 (Cost of Sales Ledger: Transaction Figures)
  - □ 0FIGL_VC2 (Cost of Sales Ledger: Balance Sheet and P&L)
  - □ 0PA_C01 (Headcount and Personnel Actions)
  - □ ZC1AUTC01 (Automobile (Original, Secured))
  - □ ZC1GDWC01 (Inventory Levels)
  - □ ZC1GDWC02 (Finance)

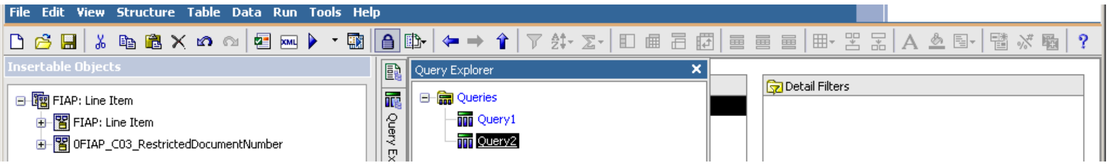
25. Ensure that the checkbox for the newly created InfoQuery has been selected. Traverse the remainder of the screens to complete the import. When completed the object should now appear under a new namespace in the IBM Cognos 8 Framework Manager model.



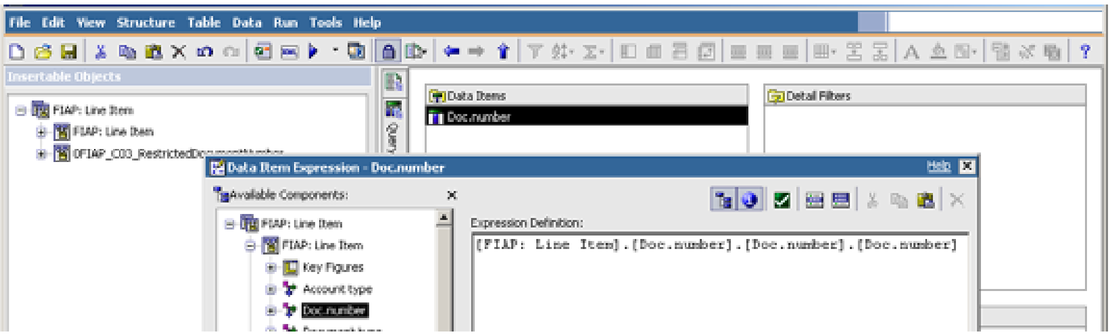
26. Save the changes and publish the package.

27. Once published, open the existing report in IBM Cognos 8 Report Studio.

28. Via the Query Explorer, select one of the prompt queries.

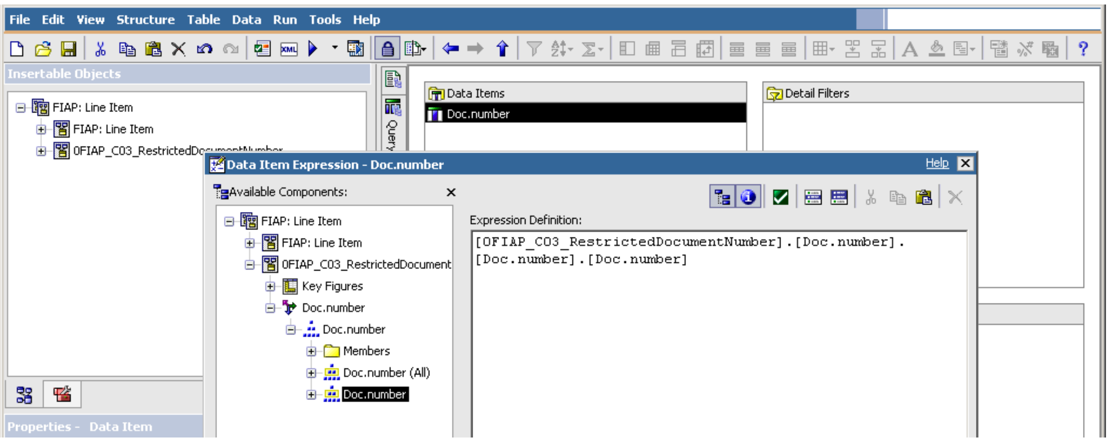


29. Double click on the QueryItem in the prompt query.



[FIAP: Line Item].[Doc.number].[Doc.number].[Doc.number]

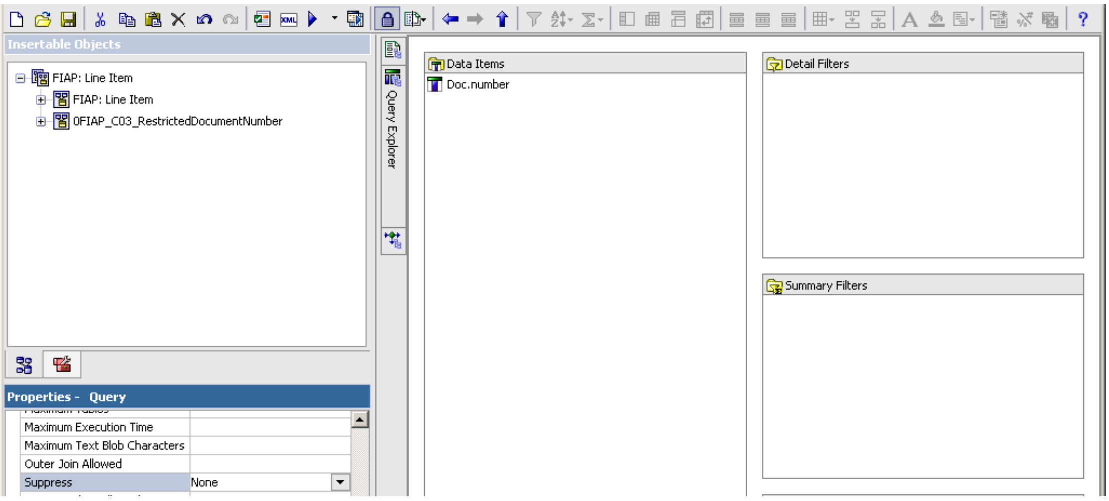
30. Delete the expression, and select the same element from within the newly imported InfoQuery.



Expression Definition:

```
[OFIAP_C03_RestrictedDocumentNumber].[Doc.number].
[Doc.number].[Doc.number]
```

31. Click OK. Then ensure that the Suppress has been set to “None” on the properties of the prompt query.



Maximum Execution Time
Maximum Text Blob Characters
Outer Join Allowed
Suppress None

32. Run the query, and you will notice the only prompt values displayed will be the selected values in the InfoQuery restriction.

4.3.6 Detect BEx Variable Uniqueness

In the event a BEx variable with the same technical name is used in two different BEx queries and is used in the same Report Studio report, the BEx variable will be displayed once and the value(s) entered will filter both BEx queries. To force the BEx variable to be displayed once for each BEx query and to allow different values to be selected for filtering, the DetectSAPVariableUniqueness parameter in qfs_config.xml can be set to true.

This is a global setting and will impact ALL Report Studio reports. More information on parameter settings are provided in Section 3 - IBM Cognos 8 SAP Provider. Below you will see which file & setting to modify to invoke the SAP variable uniqueness.

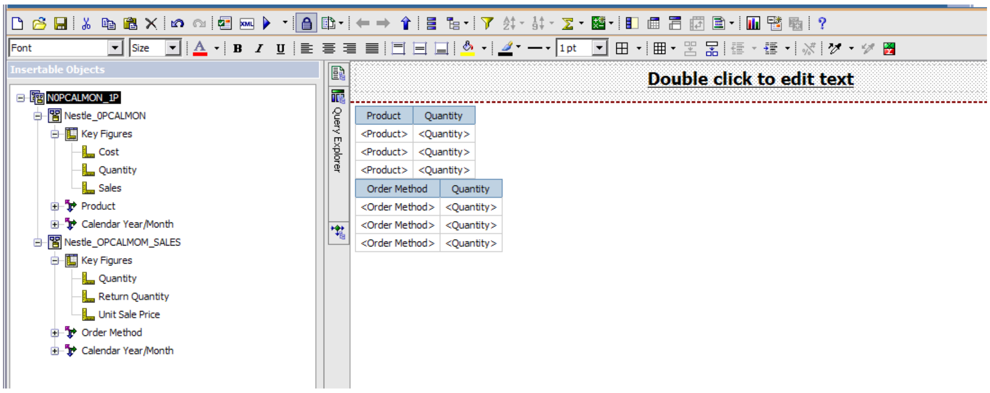
1. To turn on the setting, open <c8_install>/configuration/qfs_config.xml and browse to the following location.

   <provider name="OlapQueryProvider" libraryName="oqp">

   And ensure it has a child element which exists and is set to true.

   <parameter name="DetectSAPVariableUniqueness" value="true"/>

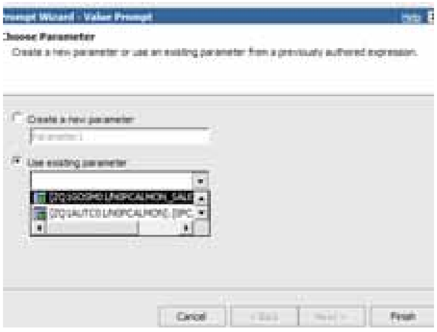
2. Open the IBM Cognos 8 package with duplicate variables and create a report with elements from each.



Product	Quantity
<Product>	<Quantity>
<Product>	<Quantity>
<Product>	<Quantity>

Order Method	Quantity
<Order Method>	<Quantity>
<Order Method>	<Quantity>
<Order Method>	<Quantity>

3. Create a prompt page and add prompts for each duplicate variable. This step is required for the functionality to work.



Choose Parameter

Create a new parameter or use an existing parameter from a previously authored expression.

○ Create a new parameter

Parameter1

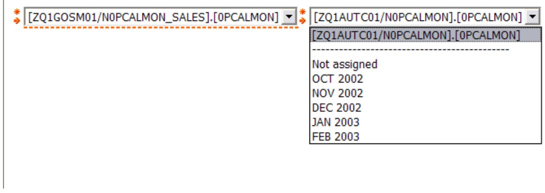
● Use existing parameter

[ZQ1GOSM01/N0PCALMON_SALE

[ZQ1AUTC01/N0PCALMON].[0PC

4. Note each duplicate variable has been prefixed by its cube name while non-duplicate variables remain non-prefixed.

5. Run the Report and fill in prompt values. Then click Finish.



[ZQ1GOSM01/N0PCALMON_SALES].[0PCALMON]

[ZQ1AUTC01/N0PCALMON].[0PCALMON]

[ZQ1AUTC01/N0PCALMON].[0PCALMON]

Not assigned
OCT 2002
NOV 2002
DEC 2002
JAN 2003
FEB 2003

4.4 Prompt Performance Tuning

4.4.1 General Guidelines

There are multiple approaches which can positively impact the performance of prompts used in IBM Cognos 8 Report Studio. These approaches include:

- Using the configuration settings covered in the configuration section of this document:
  - UserFastGetMembers
  - UserFastGetMembersFor
  - UseSAPOrdinalsForMembers
- Using Suppression to manipulate the product behaviour discussed below
- 'Use for parameter info' query setting discussed below
- Creating an InfoQuery with pre selected values to populate the prompt query.

4.4.2 Using Suppression

This is a technique used to improve the efficiency of prompt queries by reducing the number of members returned instead of returning all dimension members.

The suppression property on a query forces a non empty clause to be added after the select component of an MDX statement. This not only suppresses the resulting query data set but it also forces IBM Cognos 8 to fully deconstruct the prompt query as it would a data query.

Since prompt queries are usually simple and consist of several dimension columns, treating them as a full data query may result in some additional overhead which impacts performance.

By turning the suppression property off on the prompt query, the IBM Cognos 8 query planner is no longer forced to generate an MDX statement. The data can now be retrieved using the BAPI_MDPROVIDER_GET_MEMBERS metadata call saving the query deconstruction overhead. It is important to note that the BAPI_MDPROVIDER_GET_MEMBERS call will return all members regardless of whether or not they have populated key figures in the InfoCube.

4.4.3 'Use for parameter info' query setting

When the 'Use for parameter info' setting is used, queries created in Report Studio to reconcile the prompts are executed before the queries to populate the report.

In SAP Business Warehouse environments where there are non-hierarchical data source (BEx) variables, the number of BEx variables along with the number of possible values for these BEx variables could have a significant negative impact. The use of the 'Use for parameter info' query hint could offer performance benefits.

4.5 Reporting Overview

4.5.1 General Guidelines

When creating list reports and bringing in multiple dimensions from SAP Business Warehouse, you should remember this is an OLAP source. You are returning individual cell values. This is referred to as a cross join. As an example, if 1 million materials and 1 million customers are requested, the result set contains 1 trillion cell intersections.

The following are additional guidelines for improved functionality and performance.

For performance:

- Use of Display and Navigational Attributes

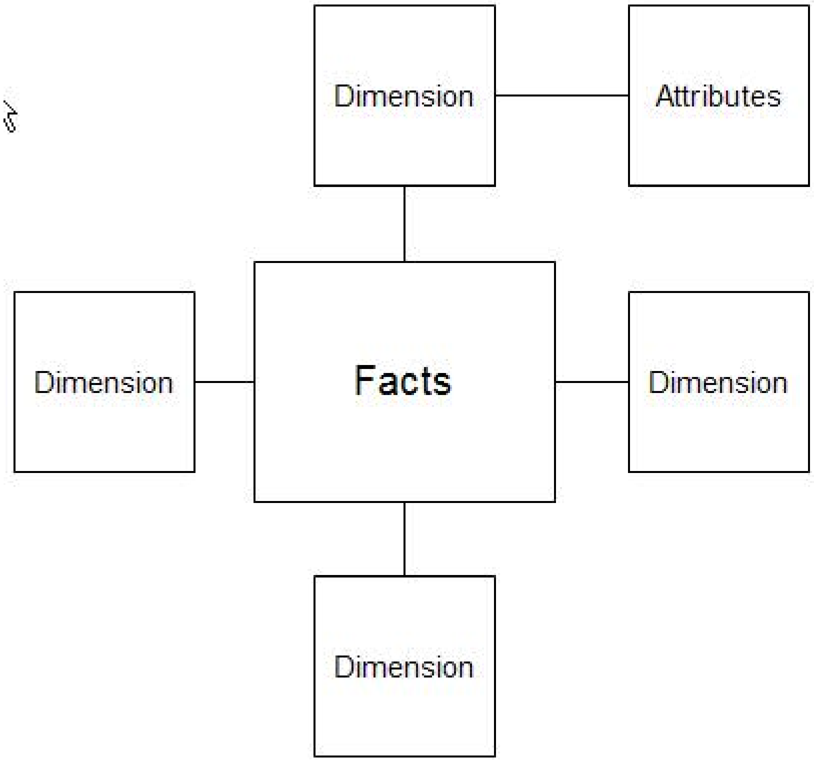
To report from SAP Business Warehouse and other heterogeneous data in one report:

- Union Between SAP Business Warehouse and Relational Data
- Master Detail Queries

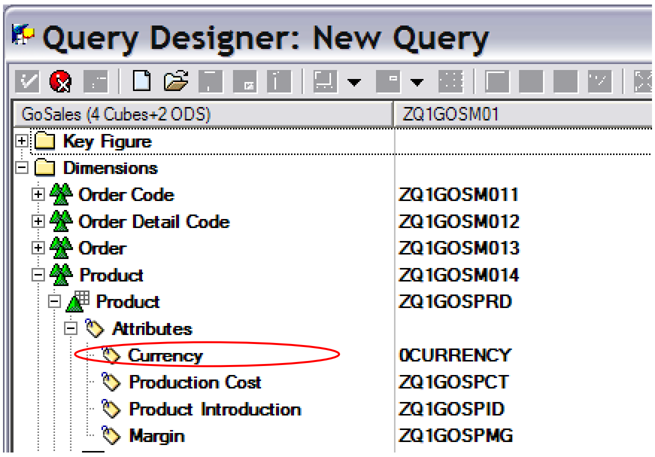
4.5.2 Use of Display and Navigational Attributes

Using a display attribute instead of an available navigational attribute will impact report performance.

Display attributes: Use the dimension levels and avoid use of display attributes for reporting AND filtering reports because they are not part of the dimension table.

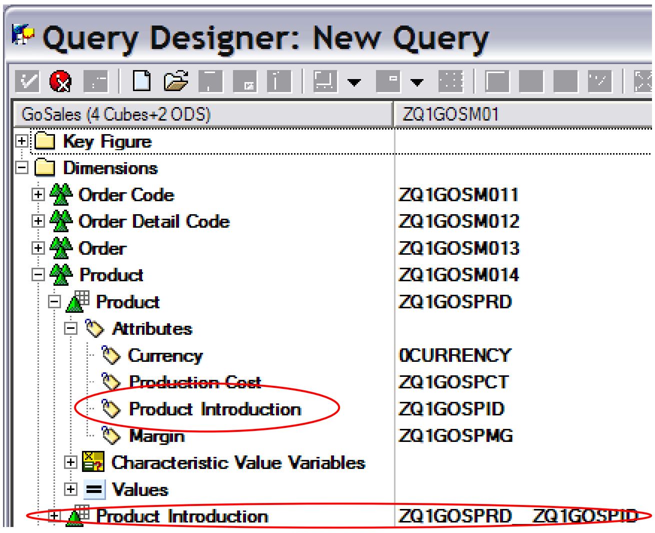


Display attributes can be defined as characteristic details which are intended for display only. These attributes cannot be grouped or sorted within Business Explorer (BEx). While they can be grouped and sorted in IBM Cognos 8, the processing takes place on the IBM Cognos 8 server instead of the SAP Business Warehouse. A display attribute is identified within BEx Query Designer by the yellow tag icon, as illustrated below.


Query Designer: New Query

GoSales (4 Cubes+2 ODS)	ZQ1GOSM01
Key Figure	
Dimensions	
Order Code	ZQ1GOSM011
Order Detail Code	ZQ1GOSM012
Order	ZQ1GOSM013
Product	ZQ1GOSM014
Product	ZQ1GOSPRD
Attributes	
Currency	0CURRENCY
Production Cost	ZQ1GOSPCT
Product Introduction	ZQ1GOSPID
Margin	ZQ1GOSPMG

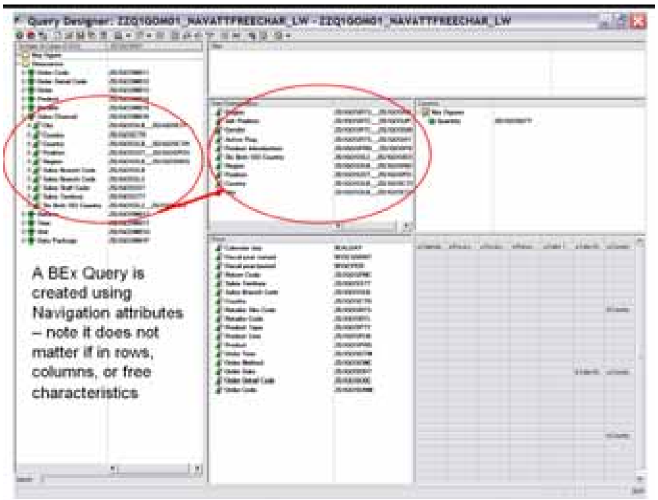
If a navigational attribute has been created, it will appear as a characteristic of the dimension in the BEx query and as a dimension in IBM Cognos 8. Navigational attributes can be grouped and sorted within BEx and IBM Cognos 8. An example of a navigational attribute is illustrated below. Please note that the Product Introduction shows up as a display attribute and a characteristic of Product. Navigational attributes will adhere to the two part name joined by a double underscore (CharacteristicTech-Name__AttributeTechName).



Query Designer: New Query

GoSales (4 Cubes+2 ODS)	ZQ1GOSM01
Key Figure	
Dimensions	
Order Code	ZQ1GOSM011
Order Detail Code	ZQ1GOSM012
Order	ZQ1GOSM013
Product	ZQ1GOSM014
Product	ZQ1GOSPRD
Attributes	
Currency	0CURRENCY
Production Cost	ZQ1GOSPCT
Product Introduction	ZQ1GOSPID
Margin	ZQ1GOSPMG
Characteristic Value Variables	
Values	
Product Introduction	ZQ1GOSPRD__ZQ1GOSPID

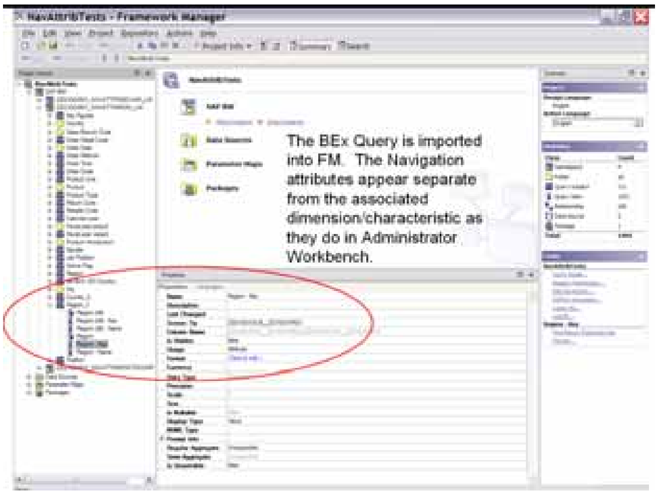
Navigation Attributes must be part of a SAP Business Warehouse Query, and cannot be accessed directly from the InfoCube/ODS structure via IBM Cognos 8.



Query Designer: ZZQ1GOSM01_NAVATTFREECHAR_LW - ZZQ1GOSM01_NAVATTFREECHAR_LW

A BEx Query is created using Navigation attributes – note it does not matter if in rows, columns, or free characteristics

After import of BEx Query into IBM Cognos 8 Framework Manager, navigation attributes will appear as dimensions as Region_1 does below. Note its technical name in the properties pain which identifies it as navigational attribute. It will appear as a display attribute as well.



The BEx Query is imported into FM. The Navigation attributes appear separate from the associated dimension/characteristic as they do in Administrator Workbench.

Once a package is created and published, select the navigation attribute as opposed to the display attribute when building reports in Report Studio. IBM Cognos 8 creates MDX that is translated to SQL to pass to the underlying database. If the display attribute is used, the generated SQL will read the fact table sequentially to pull the data referenced by the display attribute. The navigation attribute generates SQL that leverages the join defined in the navigational attribute definition.

4.6 Union between SAP NetWeaver Business Warehouse and Relational data

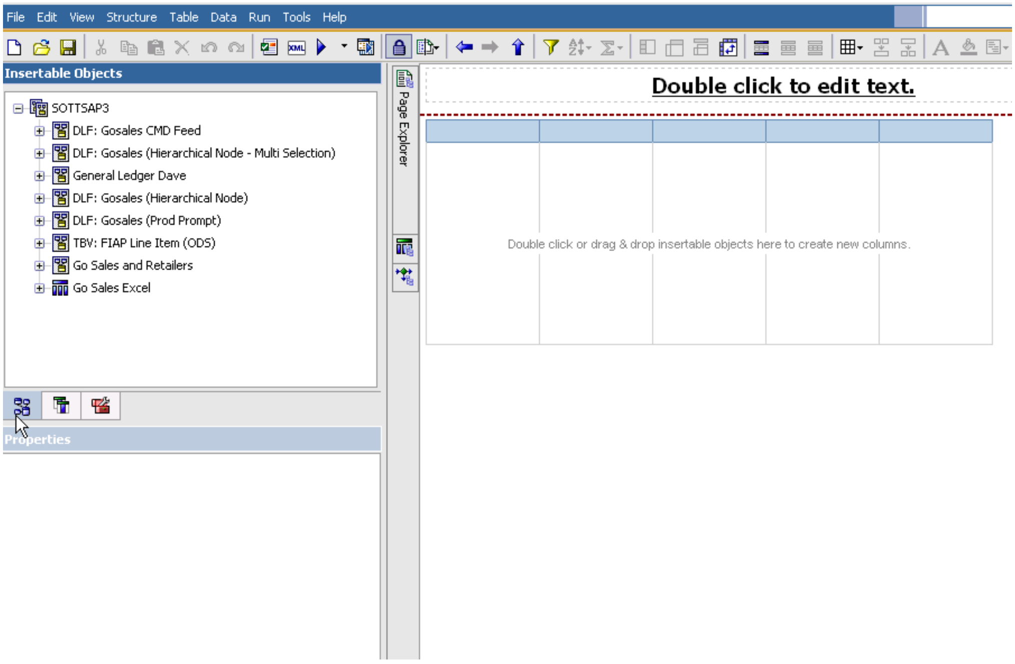
To bring in data from SAP Business Warehouse and a relational source, or two different SAP Business Warehouse data sources, and have the ability to create calculations across the two data sources, a union report can be created in report studio.

This requires a key or text field to match between the two data sources. In the Report Studio Query Explorer, two queries should be built, one against each data source, with the matching columns that are defined appropriately for data types. Then a union query is created to union the data from each data source to build the report.

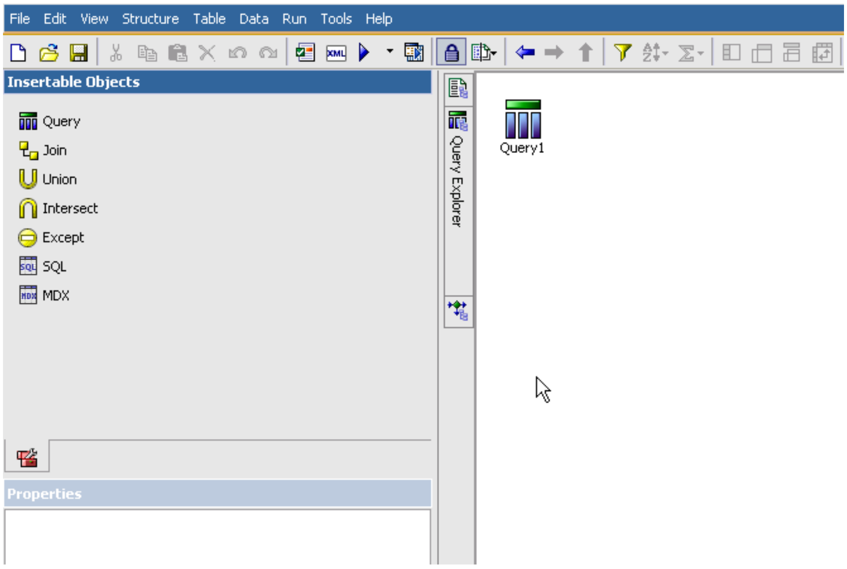
This method brings the result set back from each data source and 'unions' the data on the IBM Cognos 8 server.

4.6.1 Creating the Report and Adding a Union

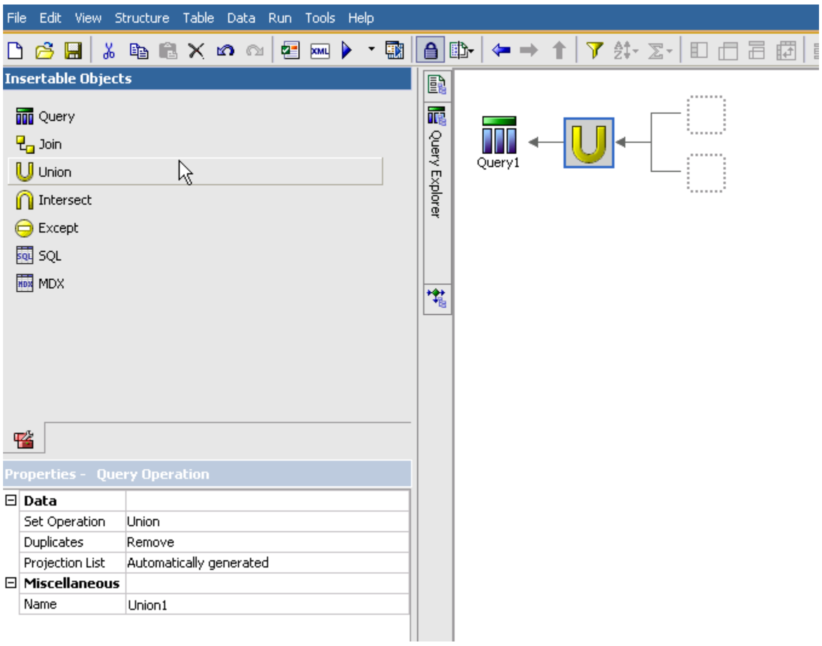
1. Start in Report Studio with a package that has both the SAP Business Warehouse and Relational data source, choose a list frame.


File Edit View Structure Table Data Run Tools Help
Insertable Objects
SOTTSAP3
DLF: Gosales CMD Feed
DLF: Gosales (Hierarchical Node - Multi Selection)
General Ledger Dave
DLF: Gosales (Hierarchical Node)
DLF: Gosales (Prod Prompt)
TBV: FIAP Line Item (ODS)
Go Sales and Retailers
Go Sales Excel
Properties
Page Explorer
Double click to edit text.
Double click or drag & drop insertable objects here to create new columns.

2. Go to Query Explorer.



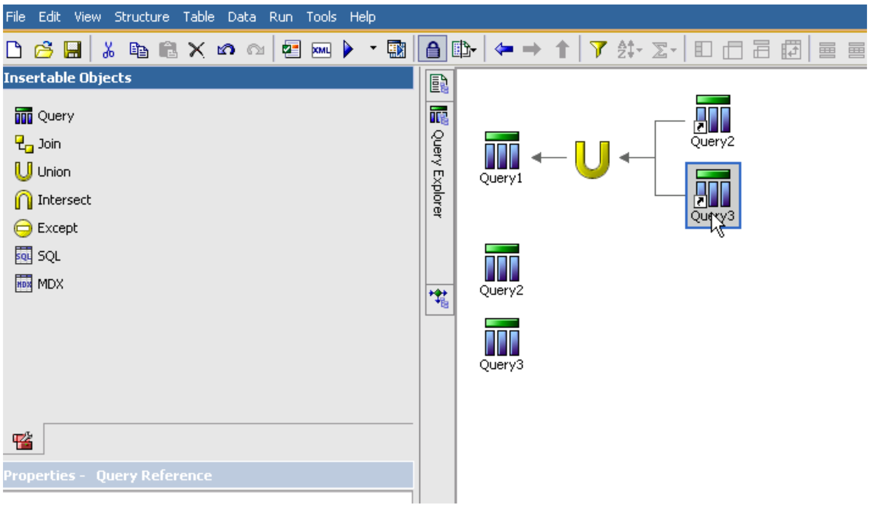
3. Add a Union from the Insertable Objects Window.


File Edit View Structure Table Data Run Tools Help
Insertable Objects
Query
Join
Union
Intersect
Except
SQL
MDX
Query Explorer
Query1
Properties - Query Operation

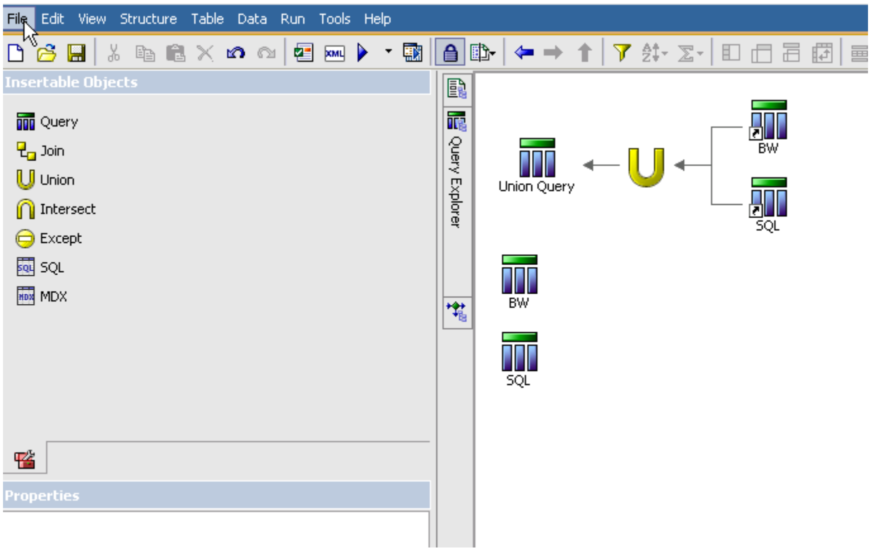
Data	
Set Operation	Union
Duplicates	Remove
Projection List	Automatically generated
Miscellaneous	
Name	Union1

4.6.2 Creating the Queries

1. Add two new queries.

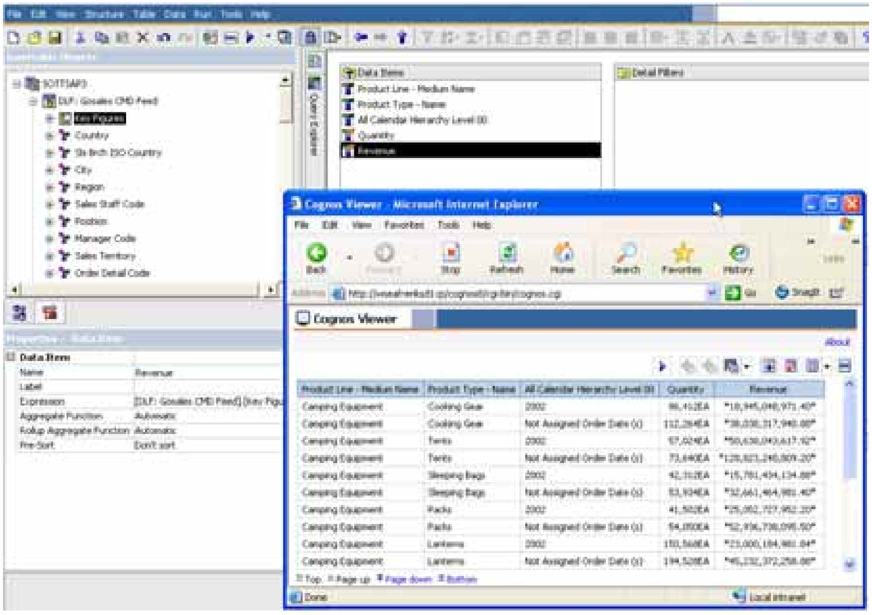


2. Change Names to Query Origin.



4.6.3 Adding Data Items to the SAP Business Warehouse Query

1. Add SAP Business Warehouse Data Items to query.

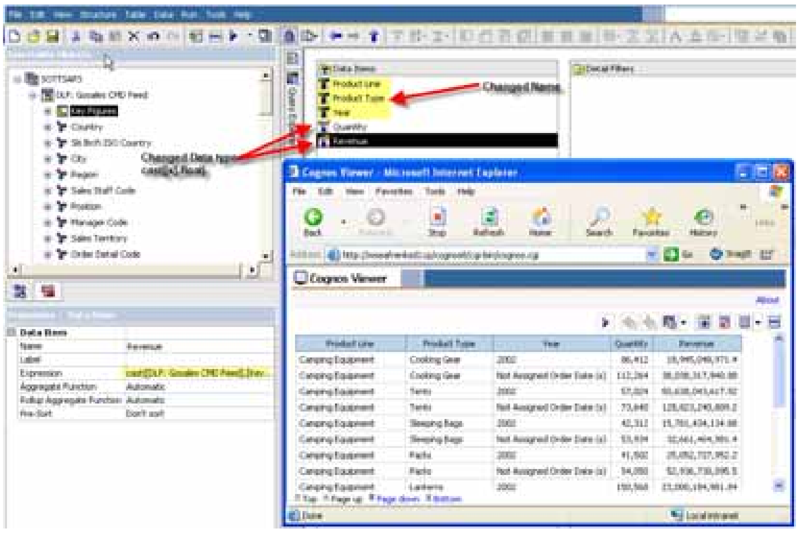

File Edit View Structure Table Data Run Tools Help
Insertable Objects
SOTTSAP3
DLF: Gosales CMD Feed
Key Figures
Country
Sls Brch ISO Country
City
Region
Sales Staff Code
Position
Manager Code
Sales Territory
Order Detail Code
Query Explorer
Data Items
Product Line - Medium Name
Product Type - Name
All Calendar Hierarchy Level 00
Quantity
Revenue
Detail Filters
Cognos Viewer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address http://vseahenks01.cp/cognos8/cgi-bin/cognos.cgi
Go
Cognos Viewer
About
Top Page up Page down Bottom
Done
Local intranet

Properties - Data Item

Data Item	
Name	Revenue
Label	
Expression	[DLF: Gosales CMD Feed].[Key Figu
Aggregate Function	Automatic
Rollup Aggregate Function	Automatic
Pre-Sort	Don't sort

Product Line - Medium Name	Product Type - Name	All Calendar Hierarchy Level 00	Quantity	Revenue
Camping Equipment	Cooking Gear	2002	96,412EA	*18,945,040,971.40*
Camping Equipment	Cooking Gear	Not Assigned Order Date (c)	112,284EA	*38,038,317,940.00*
Camping Equipment	Tents	2002	57,024EA	*50,638,043,617.92*
Camping Equipment	Tents	Not Assigned Order Date (c)	73,640EA	*128,823,240,809.20*
Camping Equipment	Sleeping Bags	2002	42,312EA	*15,781,434,134.88*
Camping Equipment	Sleeping Bags	Not Assigned Order Date (c)	53,934EA	*32,661,464,981.40*
Camping Equipment	Packs	2002	41,502EA	*25,052,727,952.20*
Camping Equipment	Packs	Not Assigned Order Date (c)	54,050EA	*52,936,730,095.50*
Camping Equipment	Lanterns	2002	153,568EA	*23,000,184,981.04*
Camping Equipment	Lanterns	Not Assigned Order Date (c)	194,528EA	*45,232,372,258.08*

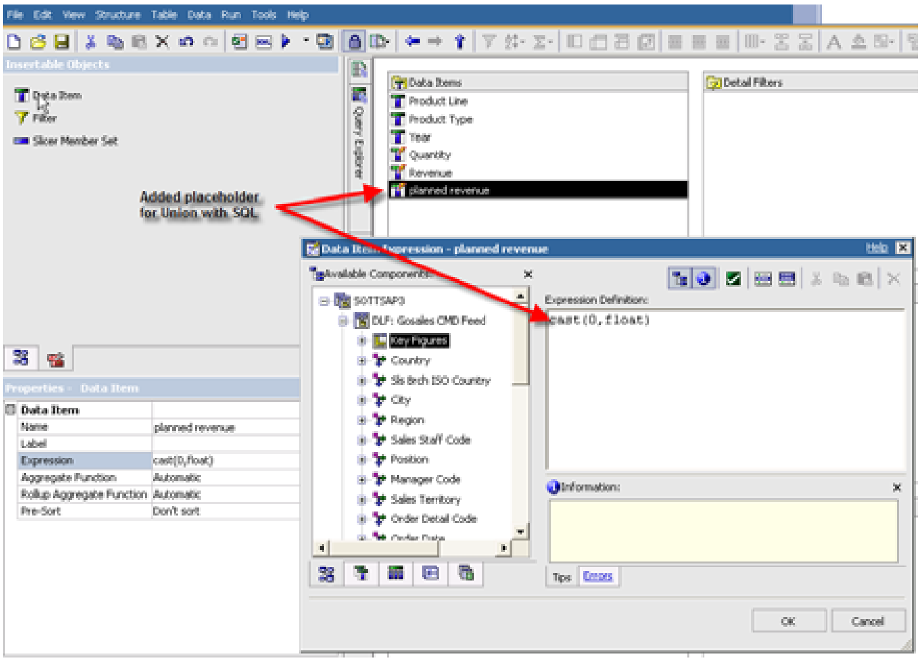
2. Change names to simplify, and *cast* key figures as FLOAT data types (this is not necessarily required with IBM Cognos 8.4, but can be used if data type incompatibilities are encountered).



Data Item	
Name	Revenue
Label	
Expression	cast([DLF: Gosales CMD Feed].[Key...
Aggregate Function	Automatic
Rollup Aggregate Function	Automatic
Pre-Sort	Don't sort

Product Line	Product Type	Year	Quantity	Revenue
Camping Equipment	Cooking Gear	2002	86,412	18,995,098,971.4
Camping Equipment	Cooking Gear	Not Assigned Order Date (s)	112,264	38,038,317,940.88
Camping Equipment	Tents	2002	57,029	50,638,043,617.92
Camping Equipment	Tents	Not Assigned Order Date (s)	73,640	128,023,240,809.2
Camping Equipment	Sleeping Bags	2002	42,312	15,781,434,134.88
Camping Equipment	Sleeping Bags	Not Assigned Order Date (s)	53,934	32,461,464,391.4
Camping Equipment	Packs	2002	41,502	25,052,727,952.2
Camping Equipment	Packs	Not Assigned Order Date (s)	54,050	52,936,730,095.5
Camping Equipment	Lanterns	2002	150,560	23,200,184,981.84

3. Add a placeholder data item for Union.

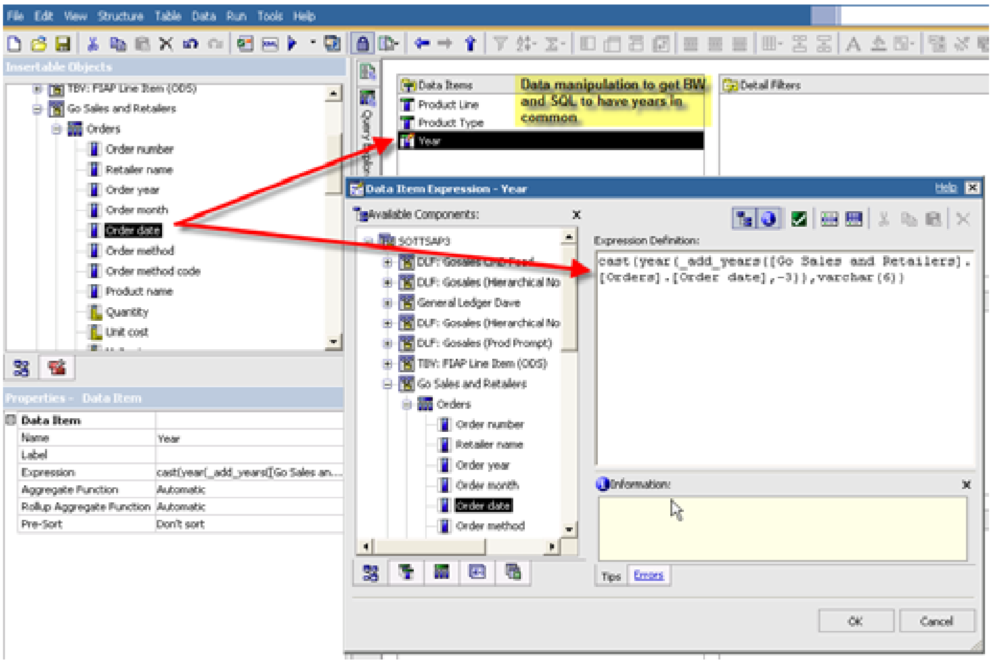


```
cast(0,float)
```

Data Item	
Name	planned revenue
Label	
Expression	cast(0,float)
Aggregate Function	Automatic
Rollup Aggregate Function	Automatic
Pre-Sort	Don't sort

4.6.4 Adding Data Items to the SQL Query

1. Switch to the SQL query for adding data items. With the SAP Business Warehouse dates in mind for the case below, Order Date may be cast as varchar.



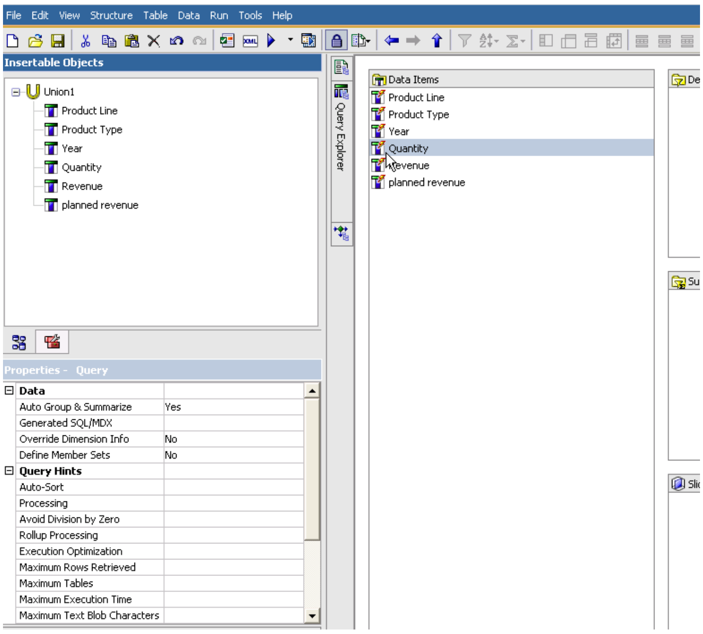
Data manipulation to get BW and SQL to have years in common.

```
cast(year(_add_years([Go Sales and Retailers].
[Orders].[Order date],-3)),varchar(6))
```

Properties - Data Item

Data Item	
Name	Year
Label	
Expression	cast(year(_add_years([Go Sales an...
Aggregate Function	Automatic
Rollup Aggregate Function	Automatic
Pre-Sort	Don't sort

2. Add rest of Measures, two place holders for Quantity and Revenue [cast(0,float)] along with the real measure - Planned Revenue cast as a float. (if needed)

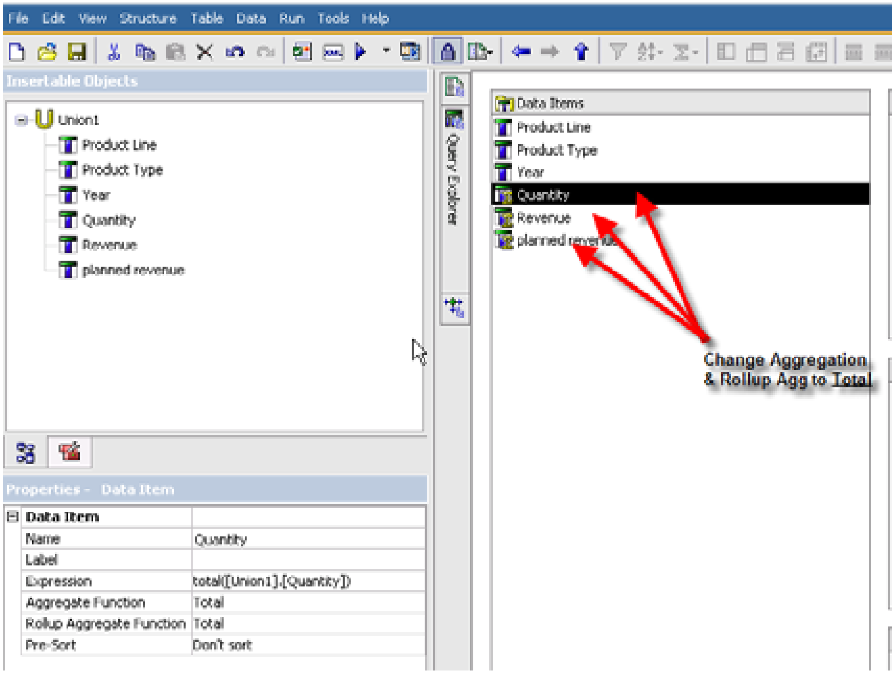


Properties - Query

Data	
Auto Group & Summarize	Yes
Generated SQL/MDX	
Override Dimension Info	No
Define Member Sets	No
Query Hints	
Auto-Sort	
Processing	
Avoid Division by Zero	
Rollup Processing	
Execution Optimization	
Maximum Rows Retrieved	
Maximum Tables	
Maximum Execution Time	
Maximum Text Blob Characters	

4.6.5 Changing the Aggregation and Rollup for the Union

1. Go to the Union Query in Query Explorer, and change aggregation & rollup to total. It is also necessary to surround each query item definition with the total(...) function to ensure correct aggregation.

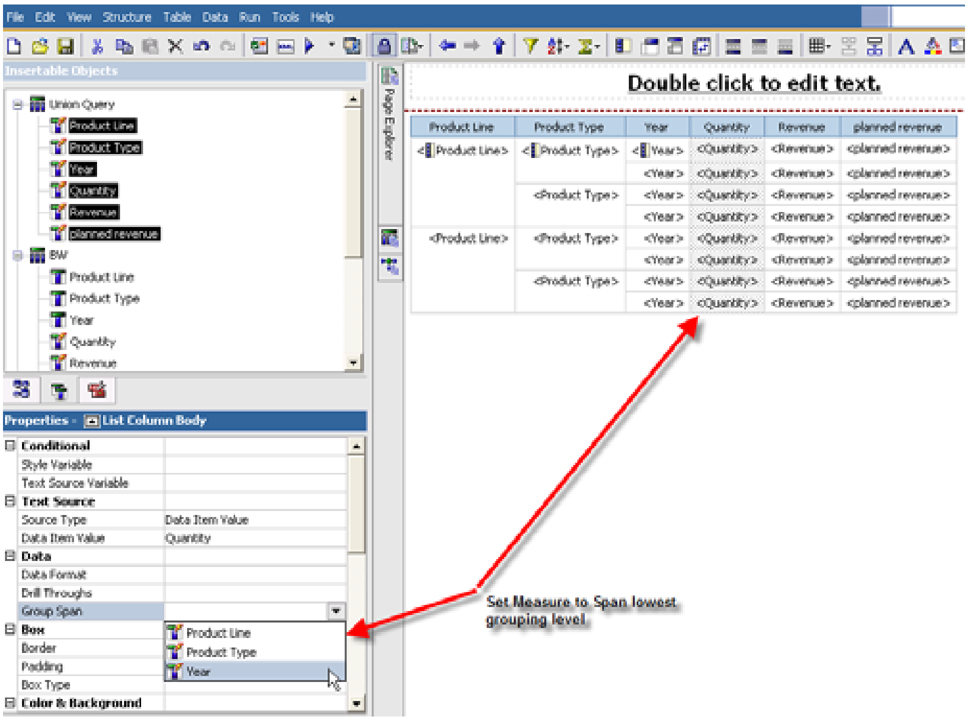


Data Item	
Name	Quantity
Label	
Expression	total([Union1].[Quantity])
Aggregate Function	Total
Rollup Aggregate Function	Total
Pre-Sort	Don't sort

Change Aggregation & Rollup Agg to Total

4.6.6 Adding Data to the List Frame

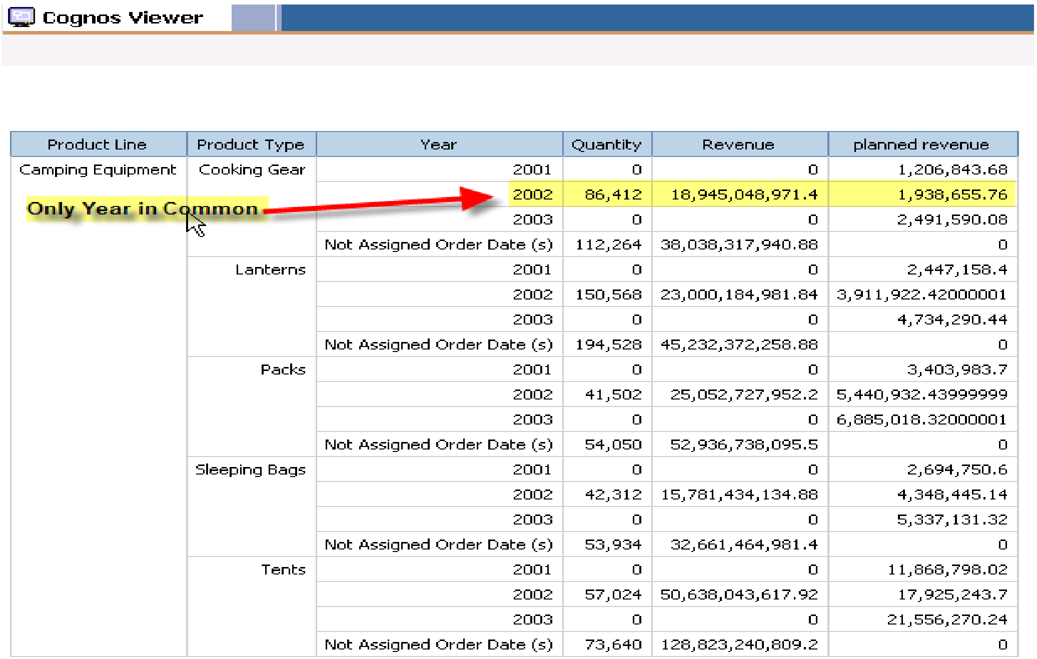
1. From the page explorer, add the column from the Union Query to the list, and group on Product Line, Product Type, and Year. For each measure, make sure the Group Span is set to the lowest level of the Grouping (Year in our case).



Source Type	Data Item Value
Data Item Value	Quantity
Group Span	Year

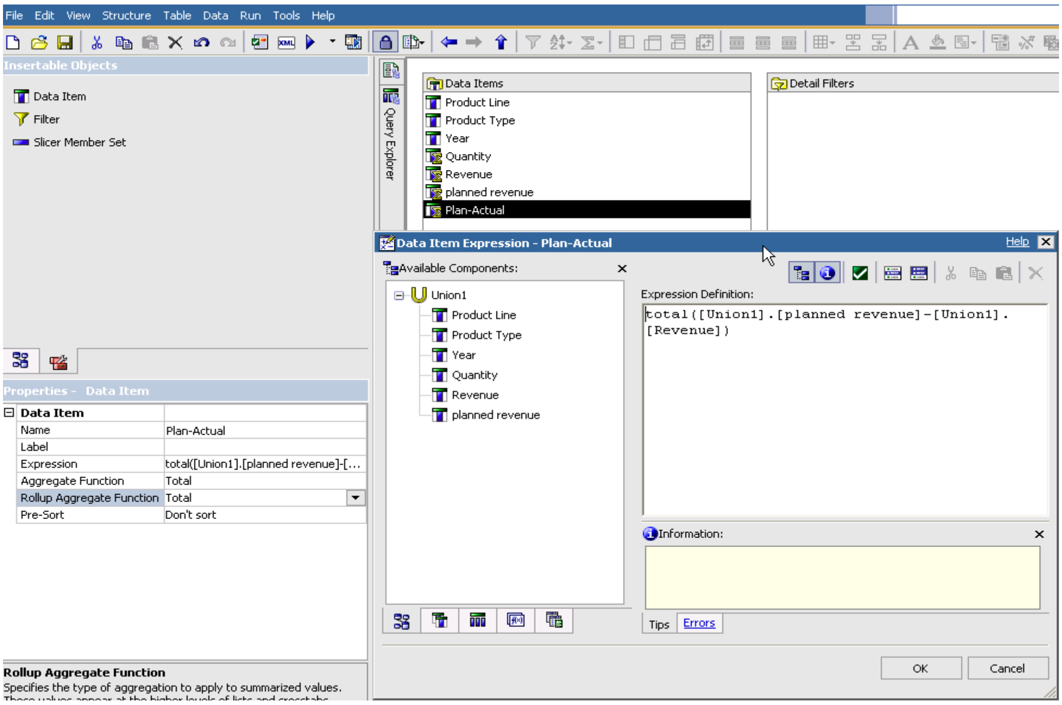
Set Measure to Span lowest grouping level

2. The report can then be run and the unioned data is displayed.



Product Line	Product Type	Year	Quantity	Revenue	planned revenue
Camping Equipment	Cooking Gear	2001	0	0	1,206,843.68
		2002	86,412	18,945,048,971.4	1,938,655.76
		2003	0	0	2,491,590.08
		Not Assigned Order Date (s)	112,264	38,038,317,940.88	0
	Lanterns	2001	0	0	2,447,158.4
		2002	150,568	23,000,184,981.84	3,911,922.42000001
		2003	0	0	4,734,290.44
		Not Assigned Order Date (s)	194,528	45,232,372,258.88	0
	Packs	2001	0	0	3,403,983.7
		2002	41,502	25,052,727,952.2	5,440,932.43999999
		2003	0	0	6,885,018.32000001
		Not Assigned Order Date (s)	54,050	52,936,738,095.5	0
	Sleeping Bags	2001	0	0	2,694,750.6
		2002	42,312	15,781,434,134.88	4,348,445.14
		2003	0	0	5,337,131.32
		Not Assigned Order Date (s)	53,934	32,661,464,981.4	0
	Tents	2001	0	0	11,868,798.02
		2002	57,024	50,638,043,617.92	17,925,243.7
		2003	0	0	21,556,270.24
		Not Assigned Order Date (s)	73,640	128,823,240,809.2	0

3. To add a calculation - Go back to Query Explorer, add a new data element (Planned Revenue vs. Revenue), and set the rollup and aggregation types and add total(xxxx) to the new calculation. Add to the list frame in Page Explorer and run.

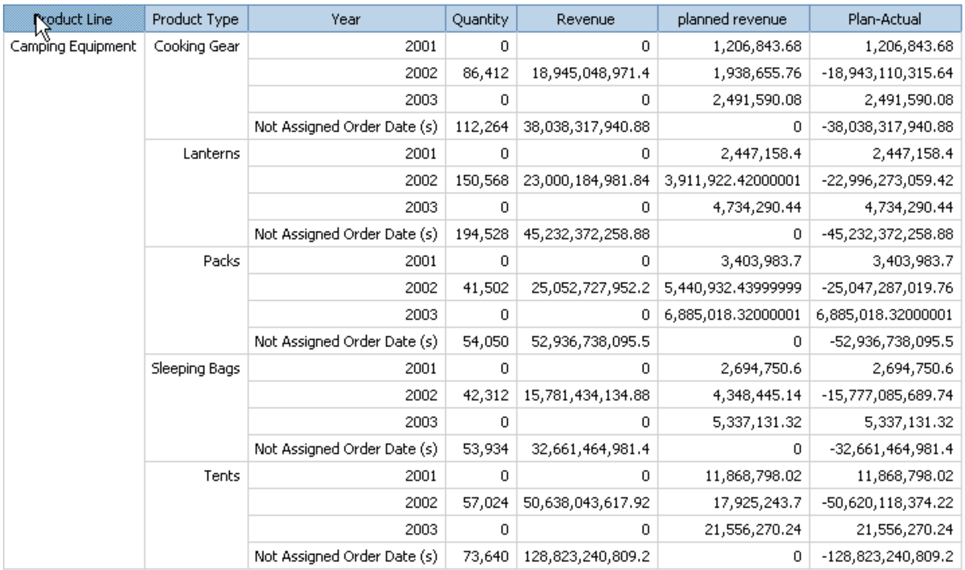


```
total([Union1].[planned revenue]-[Union1].
[Revenue])
```

Data Item	
Name	Plan-Actual
Label	
Expression	total([Union1].[planned revenue]-[...
Aggregate Function	Total
Rollup Aggregate Function	Total
Pre-Sort	Don't sort

Rollup Aggregate Function
Specifies the type of aggregation to apply to summarized values.

4. The end result appears as follows.



Product Line	Product Type	Year	Quantity	Revenue	planned revenue	Plan-Actual
Camping Equipment	Cooking Gear	2001	0	0	1,206,843.68	1,206,843.68
		2002	86,412	18,945,048,971.4	1,938,655.76	-18,943,110,315.64
		2003	0	0	2,491,590.08	2,491,590.08
		Not Assigned Order Date (s)	112,264	38,038,317,940.88	0	-38,038,317,940.88
	Lanterns	2001	0	0	2,447,158.4	2,447,158.4
		2002	150,568	23,000,184,981.84	3,911,922.42000001	-22,996,273,059.42
		2003	0	0	4,734,290.44	4,734,290.44
		Not Assigned Order Date (s)	194,528	45,232,372,258.88	0	-45,232,372,258.88
	Packs	2001	0	0	3,403,983.7	3,403,983.7
		2002	41,502	25,052,727,952.2	5,440,932.43999999	-25,047,287,019.76
		2003	0	0	6,885,018.32000001	6,885,018.32000001
		Not Assigned Order Date (s)	54,050	52,936,738,095.5	0	-52,936,738,095.5
	Sleeping Bags	2001	0	0	2,694,750.6	2,694,750.6
		2002	42,312	15,781,434,134.88	4,348,445.14	-15,777,085,689.74
		2003	0	0	5,337,131.32	5,337,131.32
		Not Assigned Order Date (s)	53,934	32,661,464,981.4	0	-32,661,464,981.4
	Tents	2001	0	0	11,868,798.02	11,868,798.02
		2002	57,024	50,638,043,617.92	17,925,243.7	-50,620,118,374.22
		2003	0	0	21,556,270.24	21,556,270.24
		Not Assigned Order Date (s)	73,640	128,823,240,809.2	0	-128,823,240,809.2

4.6.7 Master Detail Queries

Data from different data sources can also be brought into a report leveraging the Master Detail relationship in Report Studio. In particular, it is useful for two data sources with a one-to-many relationship. This could be two different Infocubes, one at a summary level and the second at a detail level.

If the SAP Business Warehouse is one data source and the second is a relational data source, use the SAP Business Warehouse data source as the master query in the relationship. Note the master detail relationship will not allow calculations across the data sources.

Note that steps to build a master detail relationship are provided within the Report Studio user guide.

4.7 Report Performance Tuning

4.7.1 General Performance Tuning Guidelines

The following outline some general rules of thumb to apply when building a Report Studio report using the SAP Business Warehouse:

1. Leverage filtering on the SAP Business Warehouse server to limit result sets returned to IBM Cognos 8.

2. Do not filter on display attributes, but preferably on the level object which will pass a mdx filter to SAP Business Warehouse.

3. When possible in Report Studio, suppress nulls triggered by a filter statement or in query properties within Query Explorer.

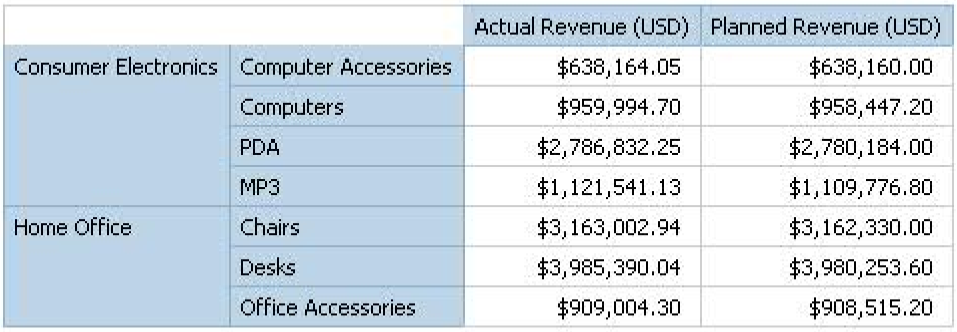
4. Remember the cross joins! As stated in the Report Studio overview section, for each characteristic brought in, the result set delivered from the SAP Business Warehouse will contain the multiple of each characteristic's members. 1000 Products * 1000 Customers returns 1,000,000 rows in the result set.

5. If practices outlined in this document are applied and better performance is desired, log a call with support and/or review configuration settings for UseMDXToRetrieveMembersFor and UseMDXToRetrieveMembersLimit which are outlined in configuration section of this document.

6. Remember IBM Cognos 8 works with the Business Warehouse Accelerator (BWA) out of the box with no additional configuration. Although BWA improves the SAP Business Warehouse query performance, continue to use practices as outlined within the SAP Business Warehouse performance tuning guide.

4.7.2 Summary Filters, Detail Filters and Slicers

Filtering in a crosstab: In a crosstab report, filtering columns on the outermost edge will have better performance than filtering columns which are nested. Below, the report is filtered on Product Line, ie, Consumer Electronics and Home Office and not Product Type, ie Computer Accessories, etc.

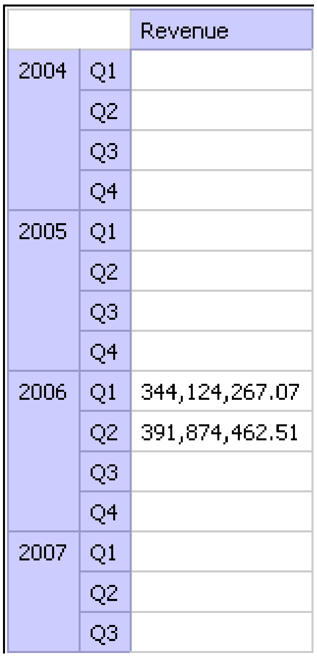


		Actual Revenue (USD)	Planned Revenue (USD)
Consumer Electronics	Computer Accessories	$638,164.05	$638,160.00
	Computers	$959,994.70	$958,447.20
	PDA	$2,786,832.25	$2,780,184.00
	MP3	$1,121,541.13	$1,109,776.80
Home Office	Chairs	$3,163,002.94	$3,162,330.00
	Desks	$3,985,390.04	$3,980,253.60
	Office Accessories	$909,004.30	$908,515.20

Summary Filter: Data is filtered after it is received from SAP Business Warehouse, then processed and aggregated on the IBM Cognos 8 server.

Detail Filter: This is a filter which is pushed down as part of the query (filtering before aggregation). IBM Cognos 8 filters (on members) are usually sent down, assuming no calculation is performed on the filter operation, unless it is an attribute filter. It should be noted that the IBM Cognos 8 filters get sent down as an extra edge on an axis and not in a WHERE clause. You MAY see a filter sent to the SAP Business Warehouse in a WHERE clause of a data fetch but that is rare. It should also be noted that NO IBM COGNOS 8 FILTERS will be sent down on an axis on MDX metadata fetches - only on the fact data fetches.

Slicer Filter: When you filter data, members that do not meet the filter criteria are removed from the report. Slicer filters are different from other filters. A slicer does not remove members from a report. Instead, their values are removed, and you see blank cells. For example, if you have a crosstab with Years and Quarters as rows, and Revenue as columns and you define a slicer member set that consists of the first two quarters in 2006, you get the following result:



		Revenue
2004	Q1	
	Q2	
	Q3	
	Q4	
2005	Q1	
	Q2	
	Q3	
	Q4	
2006	Q1	344,124,267.07
	Q2	391,874,462.51
	Q3	
	Q4	
2007	Q1	
	Q2	
	Q3	

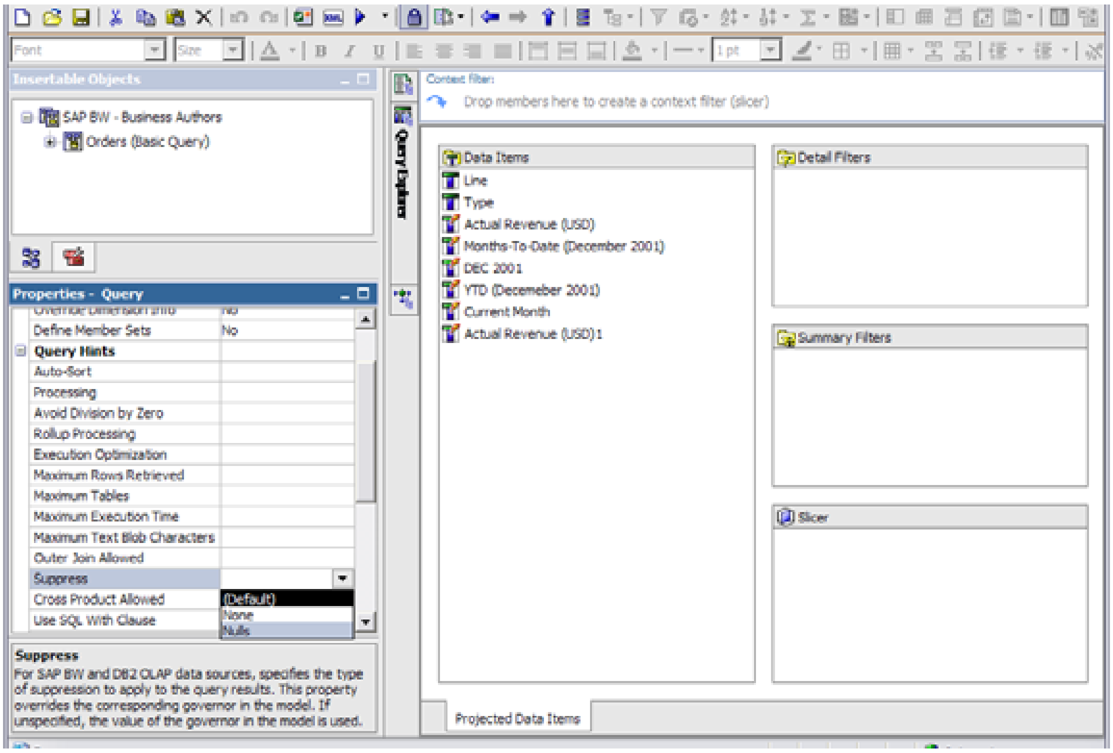
In addition, any summary values in the report are recomputed to reflect the results returned by the slicer. Slicers offer good performance, but they are not the same as filters.

4.7.3 Null suppression

The following two options are provided to you as potential methods to suppress null values from within Report Studio, which may filter nulls from the data set passed from the SAP Business Warehouse.

Option 1 - Use suppress nulls within the Query Explorer of Report Studio rather than building a filter since it is only sent down on an axis where suppress nulls inserts a “select non-empty” clause.

To set up, select the “suppress nulls” option in the query explorer as seen below:



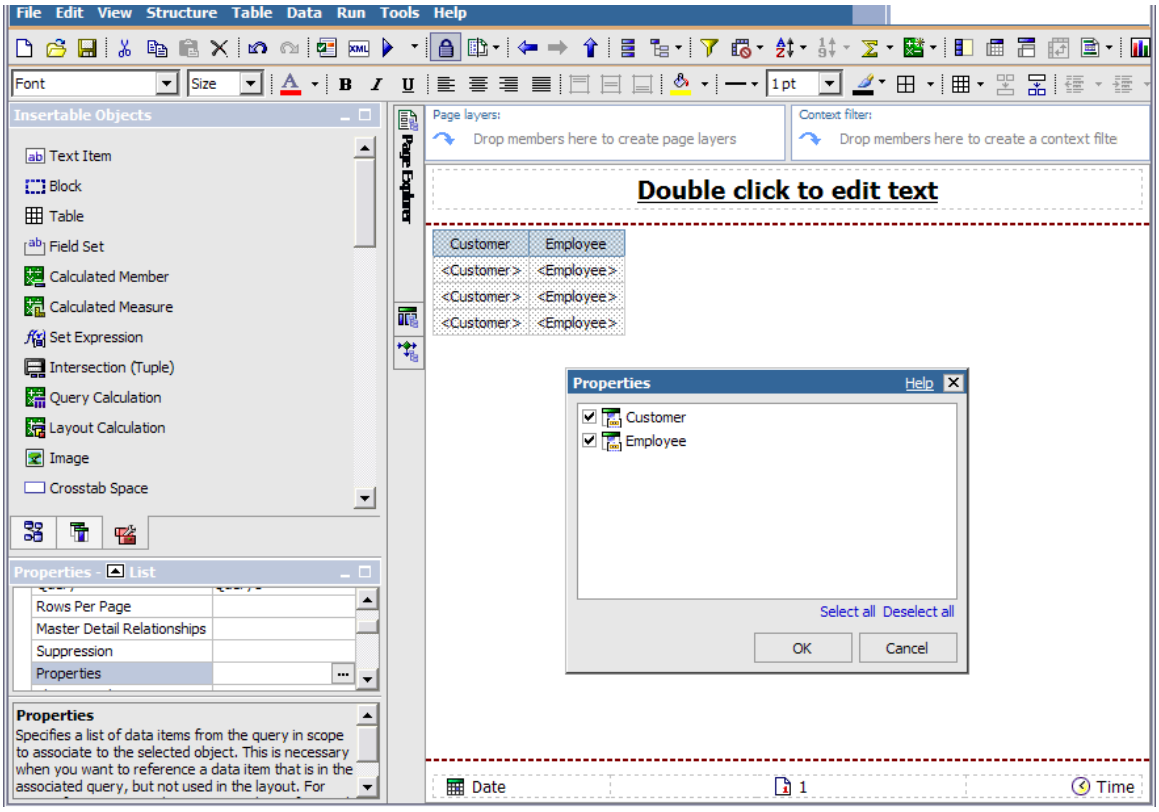
Override Dimension Info	No
Define Member Sets	No
Query Hints	
Auto-Sort	
Processing	
Avoid Division by Zero	
Rollup Processing	
Execution Optimization	
Maximum Rows Retrieved	
Maximum Tables	
Maximum Execution Time	
Maximum Text Blob Characters	
Outer Join Allowed	
Suppress	
Cross Product Allowed	(Default) None Nulls
Use SQL With Clause	

Suppress
For SAP BW and DB2 OLAP data sources, specifies the type of suppression to apply to the query results. This property overrides the corresponding governor in the model. If unspecified, the value of the governor in the model is used.

Option 2 - Null suppression is leveraged in Report Studio. Setting query suppression pushes suppression back to the SAP Business Warehouse server for processing.

To set within the Report Studio Report View, select the list or crosstab. From the properties pane, find the *properties* property and click the ellipses to display a window with characteristics of the report as displayed in screenshot below. Select one or all to invoke this suppression.

This allows dimensional data suppression based on the default key figure (first key figure listed in the IBM Cognos 8 Framework Manager model after import) containing non-zero values for all selected characteristics.



Properties

Specifies a list of data items from the query in scope to associate to the selected object. This is necessary when you want to reference a data item that is in the associated query, but not used in the layout. For

Other options to suppress nulls are:

- Filtering the dimensions directly, ie, typing in the filter expression editor ‘filter ([0customer], [revenue] is not NULL)’.
- There is also a “suppress nulls” setting in the query explorer properties section.

5 IBM Cognos 8 Query Studio

5.1 Report Do’s and Don’ts

The following statements provide general guidance which can be applied within the Query Studio environment when working with SAP Business Warehouse.

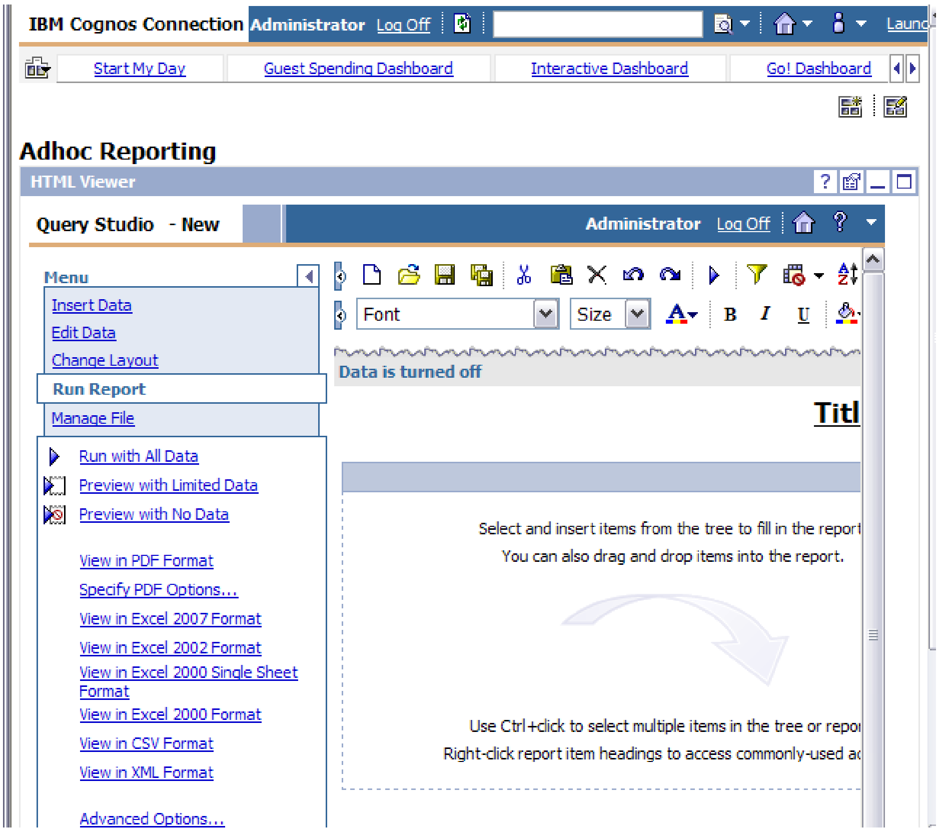
1. Use BEx queries with variables and filters to restrict the data set from BW.

2. To simplify reporting for end users, build a FM package ‘fit for purpose’ with only one data source. For example, deliver a package that delivers only what the users need from a data source, rather than allowing them to see all available content. And consider hiding or omitting elements, such as several alternate hierarchies, which they do not need. This approach will help to alleviate user-confusion and mitigate the probability that users will attempt to view high volumes of data which may provide poor performance. Details and suggestions are reviewed in FM section for designing FM package.

3. All grouping and filtering performed within Query Studio will invoke processing on the IBM Cognos 8 server after the result set has been delivered from the SAP Business Warehouse server, which could impact report performance.

4. As discussed in the Report Studio section, bring dimension levels into the report as opposed to attributes.

5. If users work from interactive mode, keep queries simple. If building a report and you do not want data to refresh as each column is brought in, create the reports in *'Preview with No Data'* mode found in the Run Report menu.



5.2 Overview of configuration settings which apply to Query Studio

Note that some settings recommended for Report Studio will also help Query Studio report performance. Refer to the configuration section in this document for more details.

However the parameter that is of most use with Query Studio is UseMDXtoretrievemembers. In general, setting this parameter to 500 will help, however you may need to adjust this setting to a value that best suits your situation.

In the event you have applied all the practices specified in this document, you should log a call with IBM Cognos support to help assess the situation.

5.3 Prompt Overview

The following are guidelines and statements regarding SAP BEx variables which can generate prompts within Query Studio.

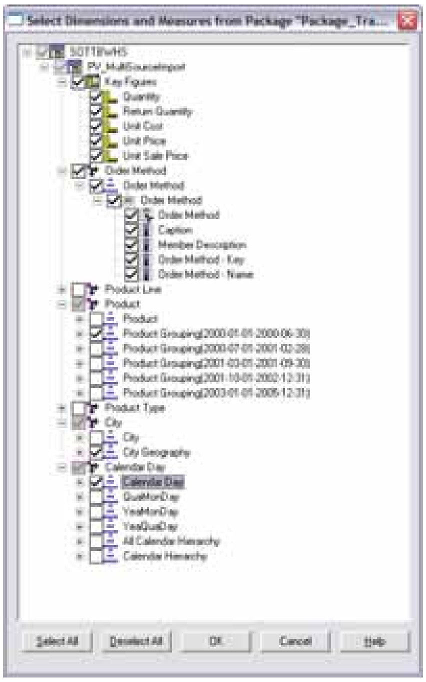
- Leverage BEx query optional variables as opposed to filtering within Query Studio. This approach will leverage the processing power of the SAP Business Warehouse server, and push filtering of the data to the SAP Business Warehouse data source.
- The format of the BEx variables will be determined by the definition of the variable within the BEx query, and changes made to the variable definition in Framework Manager.
- No formatting of the variable display format is done within Query Studio.

6 IBM Cognos 8 Transformer

6.1 Building a IBM Cognos 8 Transformer Model using a IBM Cognos 8 Package

When using a package as a data source, we will have transformer build all the dimensions automatically. While this approach is simple, performance against large cubes may suffer - especially against large hierarchies. To create the model:

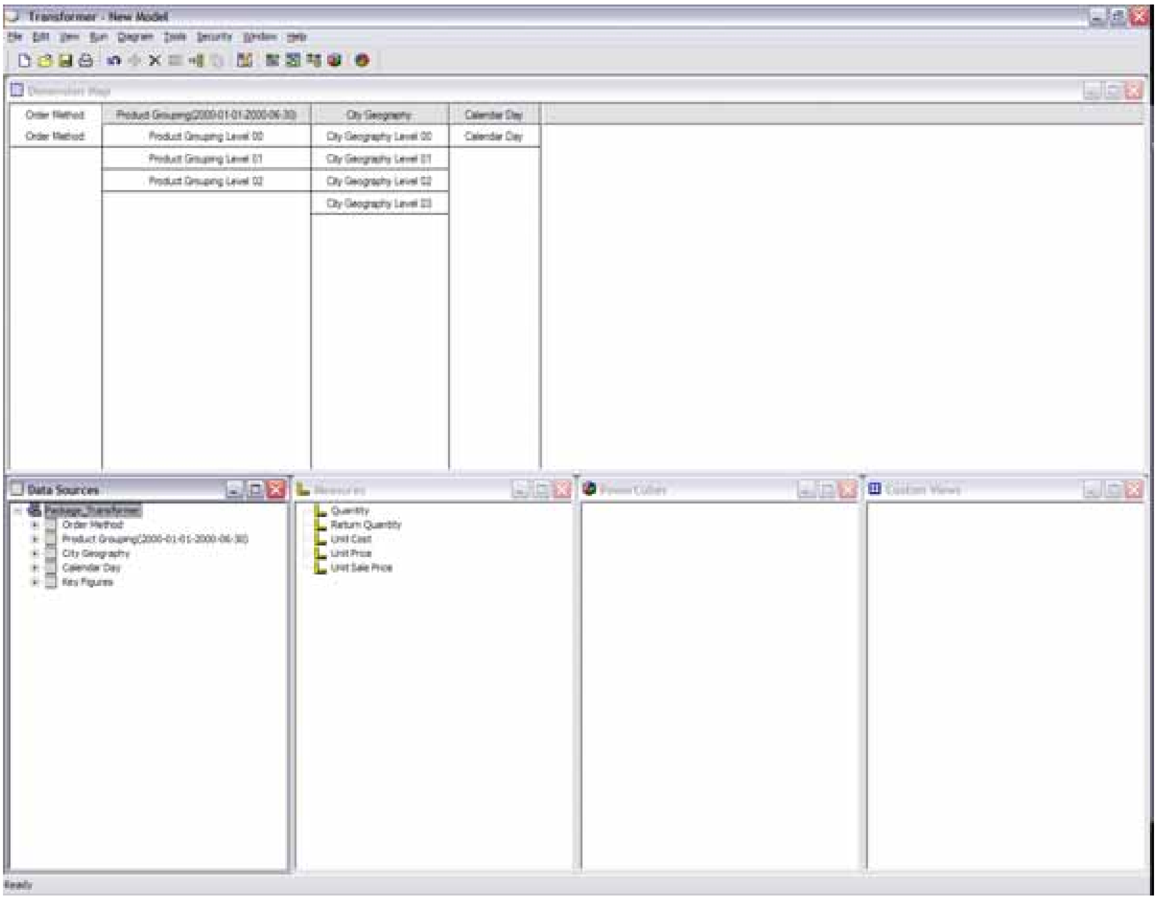
1. Open transformer and create a new model.
2. Cancel the “New Model” dialog.
3. Right-click on “Dimension Map” window and select “Insert Dimension from Package”.
4. Select the recently published package.
5. Select the dimensions required (only one hierarchy per dimension for the purpose of this exercise).
6. Select the key figures required.
7. Select the objects as per following diagram and select OK.



Select Dimensions and Measures from Package "Package_Tra...

- ☑ SOTTBWHS
  - ☑ PV_MultiSourceImport
    - ☑ Key Figures
      - ☑ Quantity
      - ☑ Return Quantity
      - ☑ Unit Cost
      - ☑ Unit Price
      - ☑ Unit Sale Price
    - ☑ Order Method
      - ☑ Order Method
        - ☑ Order Method
          - ☑ Order Method
          - ☑ Caption
          - ☑ Member Description
          - ☑ Order Method - Key
          - ☑ Order Method - Name
    - ☐ Product Line
    - ☑ Product
      - ☐ Product
      - ☑ Product Grouping(2000-01-01-2000-06-30)
      - ☐ Product Grouping(2000-07-01-2001-02-28)
      - ☐ Product Grouping(2001-03-01-2001-09-30)
      - ☐ Product Grouping(2001-10-01-2002-12-31)
      - ☐ Product Grouping(2003-01-01-2005-12-31)
    - ☐ Product Type
    - ☑ City
      - ☐ City
      - ☑ City Geography
    - ☑ Calendar Day
      - ☑ Calendar Day
      - ☐ QuaMonDay
      - ☐ YeaMonDay
      - ☐ YeaQuaDay
      - ☐ All Calendar Hierarchy
      - ☐ Calendar Hierarchy

This should create the following model.



Transformer - New Model

Order Method	Product Grouping(2000-01-01-2000-06-30)	City Geography	Calendar Day
Order Method	Product Grouping Level 00	City Geography Level 00	Calendar Day
	Product Grouping Level 01	City Geography Level 01	
	Product Grouping Level 02	City Geography Level 02	
		City Geography Level 03	

Data Sources

- Package_Transformer
  - Order Method
  - Product Grouping(2000-01-01-2000-06-30)
  - City Geography
  - Calendar Day
  - Key Figures

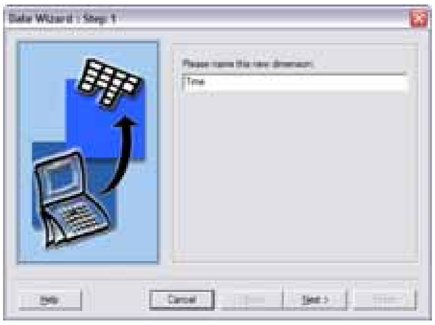
Measures

- Quantity
- Return Quantity
- Unit Cost
- Unit Price
- Unit Sale Price

There is some minor housekeeping to be performed at this point. “Calendar Day” was imported during the “Insert Dimension…” step. However, it would be best to populate the dimension using the “Date Wizard” utilizing fact data values instead to populate it given the date column is returned whole in the fact data results. We needed to include “Calendar Day” in the initial import in order to make sure it is included in the “Key Figures” query utilized to extract fact data.

To do this:

8. Delete the “Calendar Day” query from the “Data Source” window and invoke the “Date Wizard” by right-clicking the Dimension dialog box.

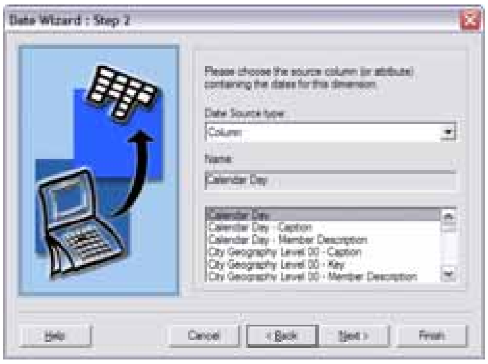


Date Wizard : Step 1

Please name this new dimension:

Time

9. Type a name and click next.



Date Wizard : Step 2

Please choose the source column (or attribute) containing the dates for this dimension.

Date Source type:

Column

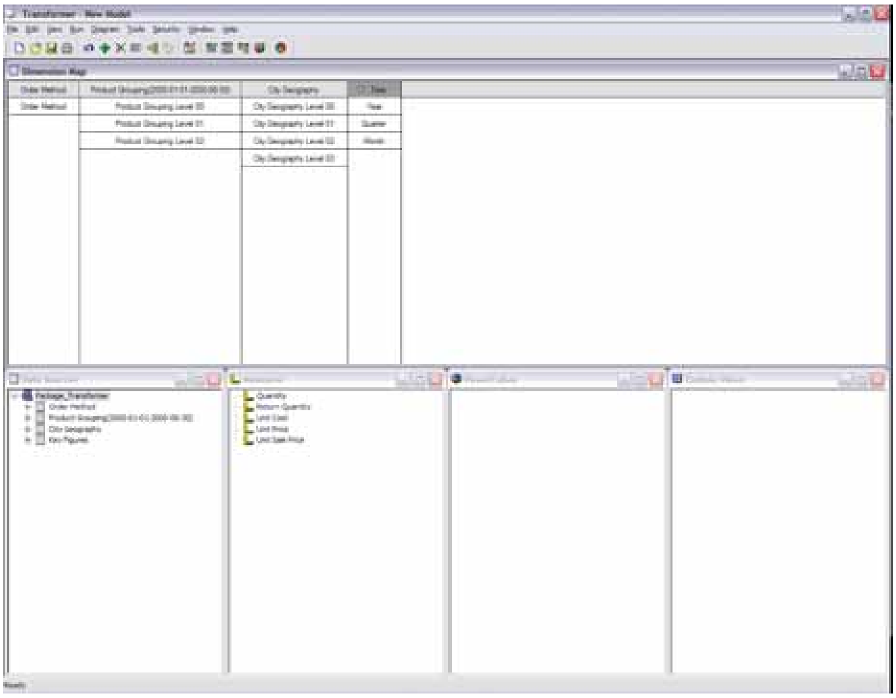
Name:

Calendar Day

Calendar Day
Calendar Day - Caption
Calendar Day - Member Description
City Geography Level 00 - Caption
City Geography Level 00 - Key
City Geography Level 00 - Member Description

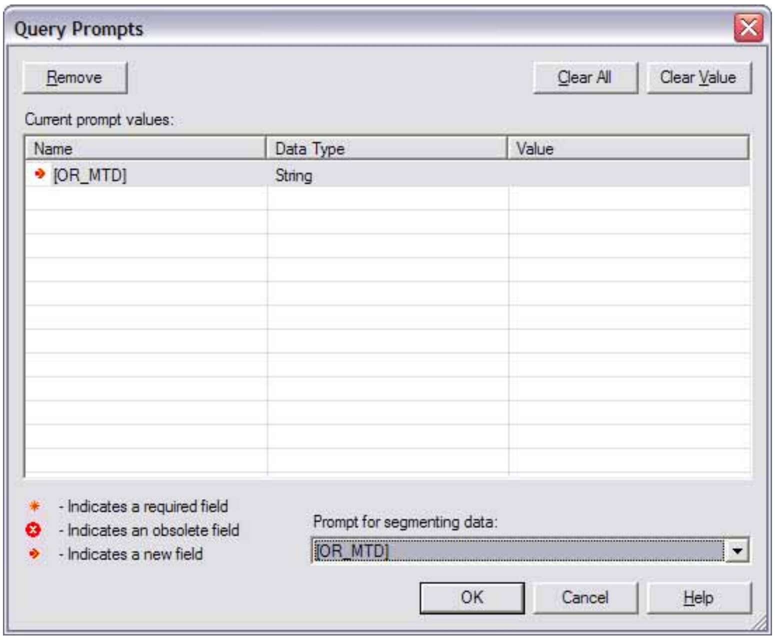
10. Select “Calendar Day” from the columns and click next.

11. Select what you like included in the date dimension. The final window should look like the following.



The final step required is to make sure the fact extract utilizes the defined optional variable. In order to accomplish this:

12. Right-click on “Key Figures” query within the “Data Source” window and select “Edit Prompts”.
13. Change “Prompt for Segmenting data” to the optional prompt used for slicing the data.



Current prompt values:

Name	Data Type	Value
[OR_MTD]	String	

- Indicates a required field
- Indicates an obsolete field
- Indicates a new field

Prompt for segmenting data: [OR_MTD]

14. At this point, click OK to generate the cube.

This approach to Transformer cube builds is simple and fairly quick to develop. There is a clear advantage of utilizing the optional variable approach (referred to as StreamBAPI) which is the single most important step to improve performance and exceed the 1 million cell SAP Business Warehouse limitation. However, there are other areas which can potentially be improved - most notably the extraction of hierarchies.

By default, given the approach taken above, we have left our engine perform all the work to build the hierarchies and dimensions. While this is simple and quick from the surface, this will require the engine to build the entire hierarchy tree utilizing one MDX statement. This one MDX statement will be constructed by fetching all the members of each level, finding all their children and perform this process for all levels in order to build the entire hierarchical tree. This step will also utilize fact tables in order to populate it.

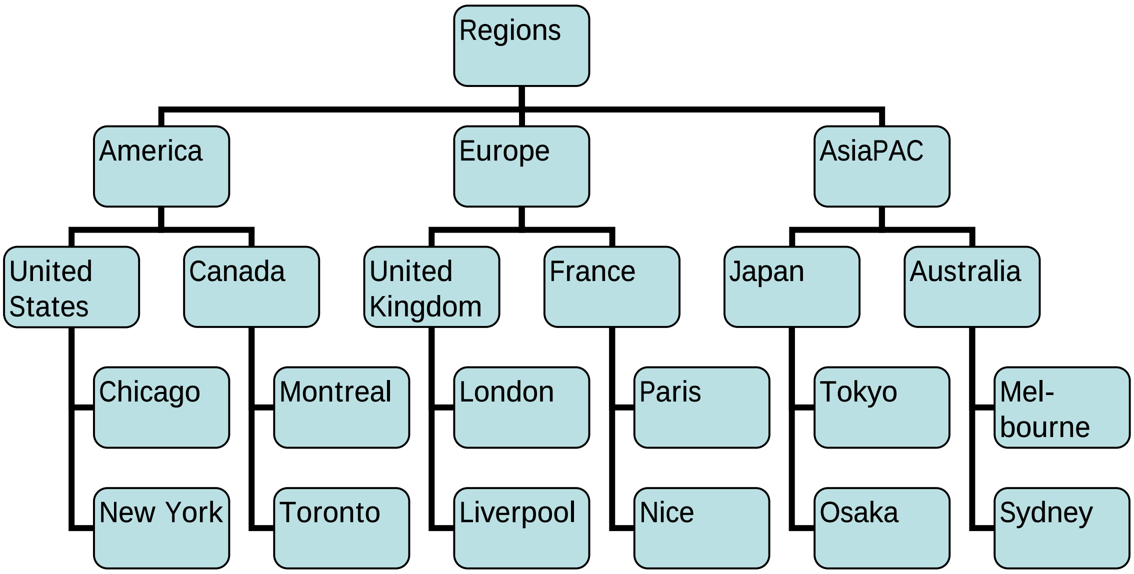
Consider a “Region” hierarchy containing three levels.

- Level_00 is Continent (America, Europe, AsiaPAC)
- Level_01 is Country (United States, Canada, UK, France, Australia, Japan)
- Level_02 is City (Chicago, New York, Montreal, Toronto, London, Liverpool, Paris, Nice, Melbourne, Sydney, Osaka, Tokyo)

When transformer is left to build the hierarchy, the general process will flow as such:

- Get members for Level_00 which returns “America, Europe, AsiaPAC”
- Get children (level_01) of America which returns “United States, Canada”
- Get children (level_02) of United States which returns “Chicago, New York”
- Get children (level_02) of Canada which returns “Montreal, Toronto”
- Get children (level_01) of Europe….

This process would be continued until the entire tree is built creating the following diagram:



Once created, we would build a final MDX statement to fetch all the members which have fact data. While easy and effective, this process may not perform adequately in some environment. What it gains in ease of creation may be lost in performance. While the bulk of the processing is usually consumed by the fact data extraction, it may be useful to gain some performance in other areas such as this one.

6.2 Using IBM Cognos 8 Framework Manager to Externalize Data for Consumption within IBM Cognos 8 Transformer

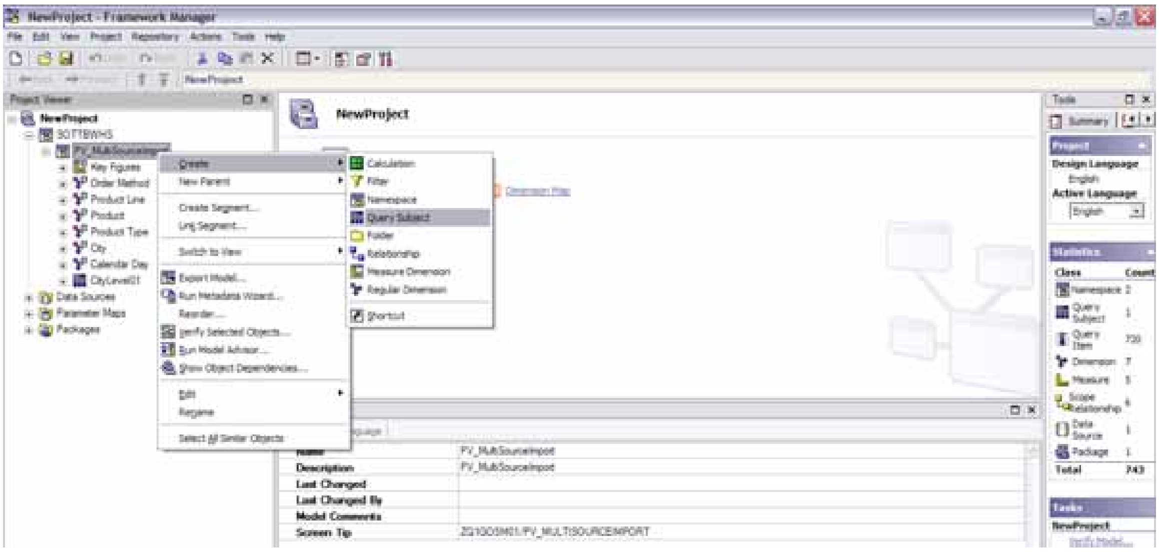
Another way to populate the hierarchy would be to export (CSV) the dimensions/hierarchies/levels from Framework Manager which would later be imported in Transformer. While extra steps are required, it's important to remember that you only need to create the model once. Careful planning only needs to be considered at the beginning.

The advantage of extracting hierarchies directly from FM allows the execution of MDX which avoids the inclusion of fact tables (it should be noted that various BW systems may cause different internal processing based on patch levels etc…). The MDX takes the following form:

```
WITH MEMBER [Measures].[COG_OQP_USR_COG_OQP_INT_m1noFact] AS '1' SE-
LECT [ZQ1GOSSLB__ZQ1GOSCIT          CITY_GEOGRAPHY].[LEVEL00].MEMBERS
DIMENSION PROPERTIES PARENT_LEVEL,
CHILDREN_CARDINALITY,
MEMBER_UNIQUE_NAME,
[ZQ1GOSSLB__ZQ1GOSCIT].[5ZQ1GOSSLB__ZQ1GOSCIT],
[ZQ1GOSSLB__ZQ1GOSCIT].[2ZQ1GOSSLB__ZQ1GOSCIT],
DESCRIPTION,
PARENT_UNIQUE_NAME
ON AXIS(0)
FROM [ZQ1GOSM01/PV_MULTISOURCEIMPORT] WHERE [Meas-
ures].[COG_OQP_USR_COG_OQP_INT_m1noFact] CELL PROPERTIES FORMAT_STRING
```

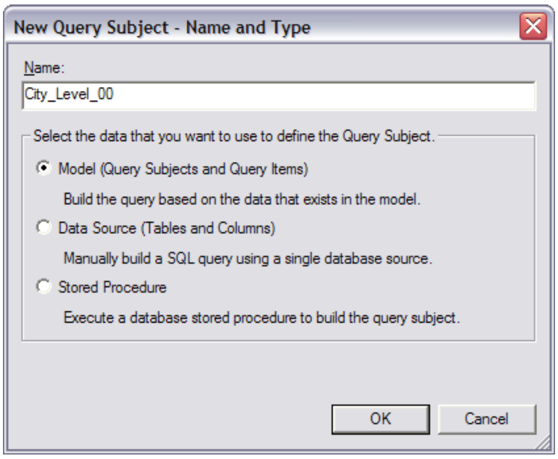
To extract a hierarchy manually within IBM Cognos 8 Framework Manager:

1. Create a new query subject



Name: PV_MultiSourceImport
Description: PV_MultiSourceImport
Last Changed
Last Changed By
Model Comments
Screen Tip: ZQ1GOSM01/PV_MULTISOURCEIMPORT

2. Name it accordingly then click OK:



New Query Subject - Name and Type

Name:
City_Level_00

Select the data that you want to use to define the Query Subject.

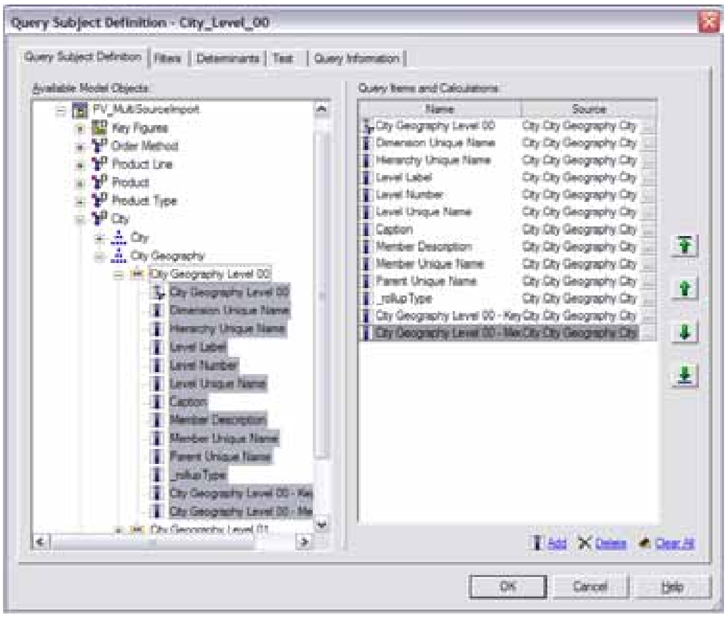
● Model (Query Subjects and Query Items)
Build the query based on the data that exists in the model.

○ Data Source (Tables and Columns)
Manually build a SQL query using a single database source.

○ Stored Procedure
Execute a database stored procedure to build the query subject.

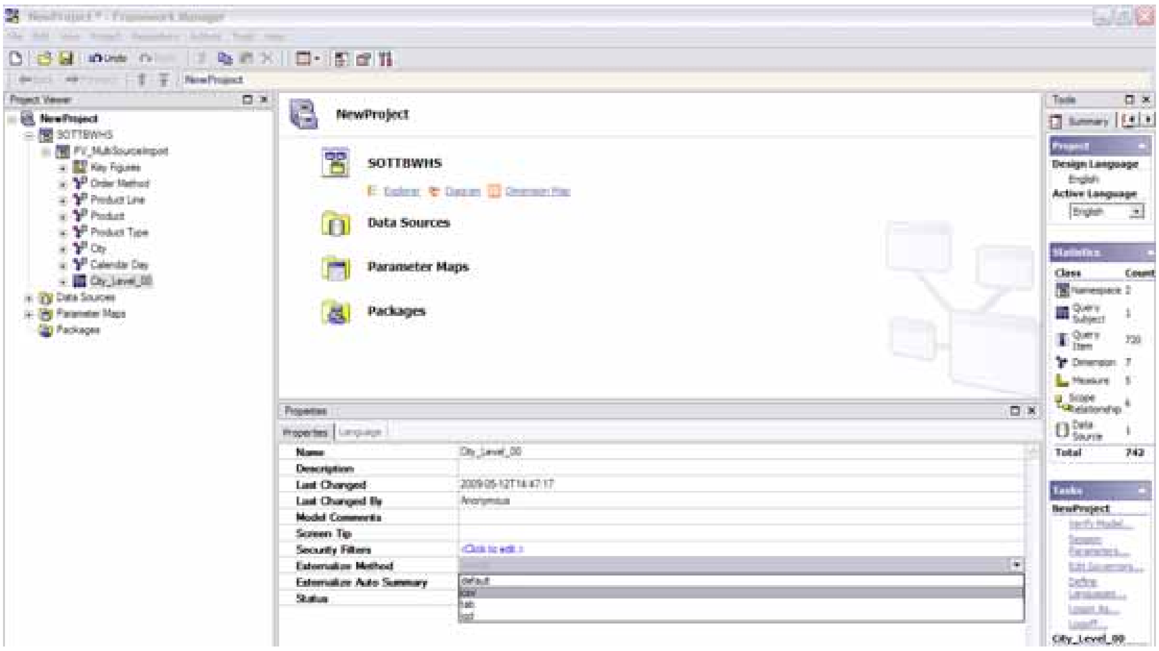
OK Cancel

3. Drag the concerned Level – in our case City Geography Level 00:



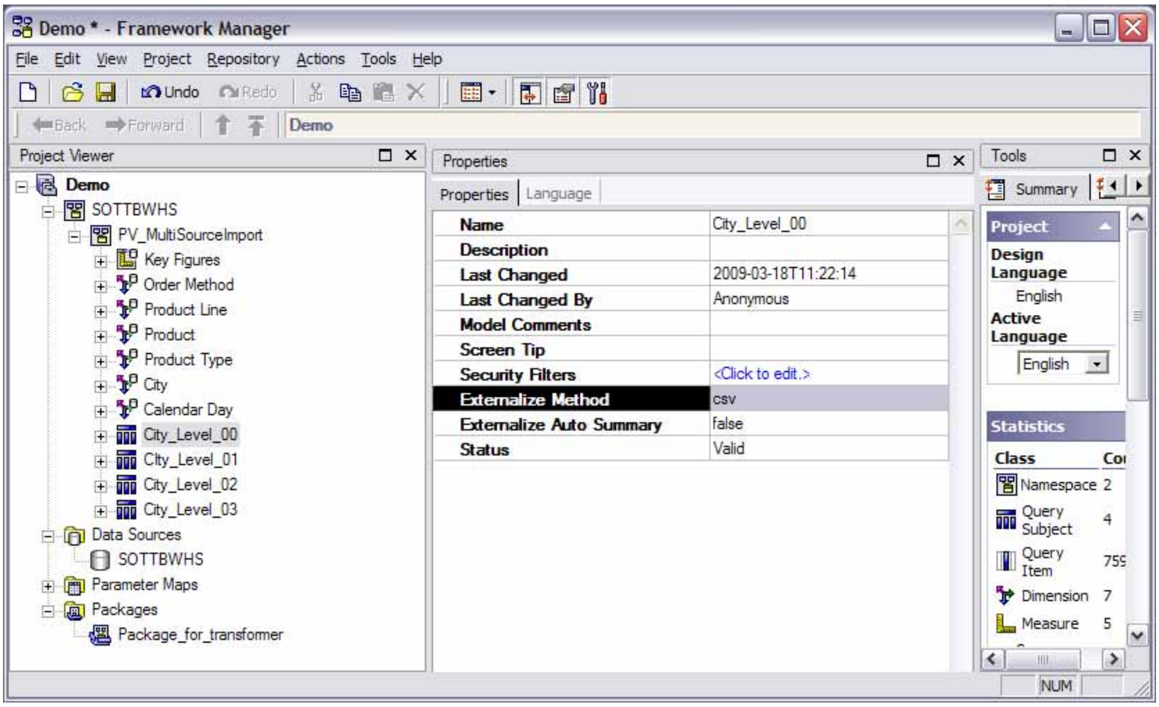
Name	Source
City Geography Level 00	City City Geography City
Dimension Unique Name	City City Geography City
Hierarchy Unique Name	City City Geography City
Level Label	City City Geography City
Level Number	City City Geography City
Level Unique Name	City City Geography City
Caption	City City Geography City
Member Description	City City Geography City
Member Unique Name	City City Geography City
Parent Unique Name	City City Geography City
_rollupType	City City Geography City
City Geography Level 00 - Key	City City Geography City
City Geography Level 00 - Me	City City Geography City

4. Once created, select the newly created query subject in the “Project Viewer” area and change the externalize method to CSV.



Name	City_Level_00
Description	
Last Changed	2009-05-12T14:47:17
Last Changed By	Anonymous
Model Comments	
Screen Tip	
Security Filters	<Click to edit..>
Externalize Method	default csv tab txt
Externalize Auto Summary	
Status	

5. Perform this task for all levels in the given hierarchy.



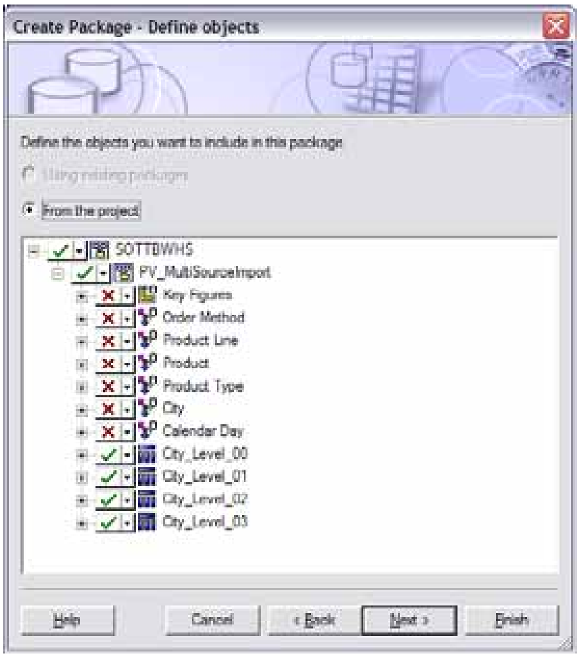
Name	City_Level_00
Description	
Last Changed	2009-03-18T11:22:14
Last Changed By	Anonymous
Model Comments	
Screen Tip	
Security Filters	<Click to edit.>
Externalize Method	csv
Externalize Auto Summary	false
Status	Valid

Design Language: English

Active Language: English

Class	Co
Namespace	2
Query Subject	4
Query Item	759
Dimension	7
Measure	5

6. Create a package including only the created four (4) query subjects.

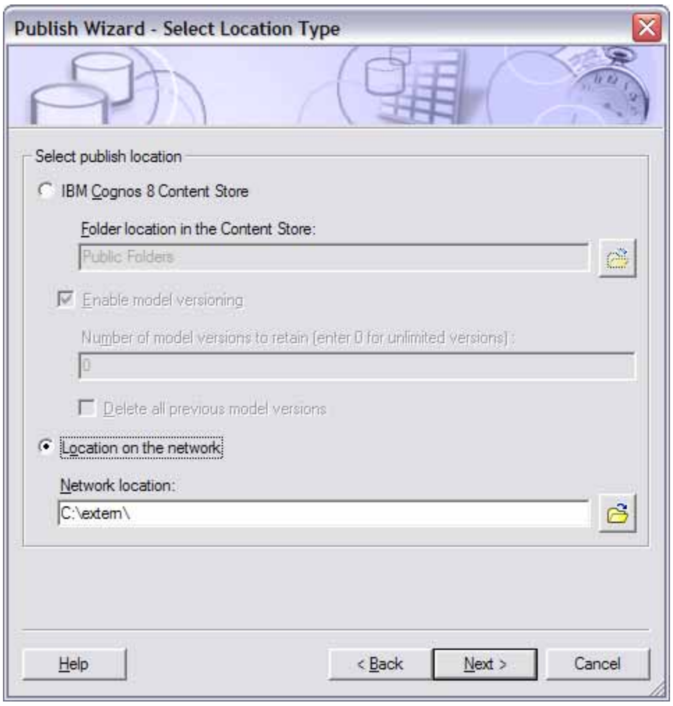


Define the objects you want to include in this package

○ Using existing packages

● From the project

7. Given you will not need to publish this to the portal, you can publish the package to a location on the network.



Publish Wizard - Select Location Type

Select publish location

○ IBM Cognos 8 Content Store

Folder location in the Content Store:

Public Folders

☑ Enable model versioning

Number of model versions to retain (enter 0 for unlimited versions) :

0

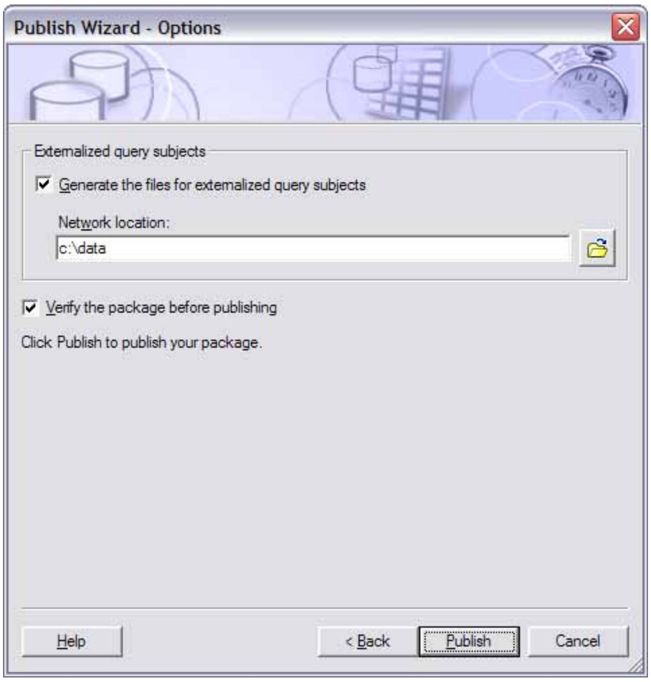
□ Delete all previous model versions

● Location on the network

Network location:

C:\extern\

8. Accept all defaults until you get to the following screen:



Publish Wizard - Options

Externalized query subjects

☑ Generate the files for externalized query subjects

Network location:

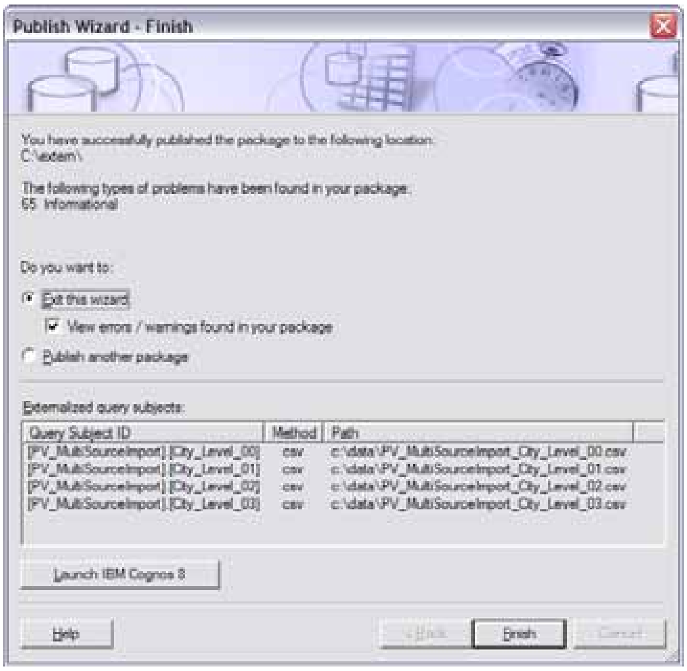
c:\data

☑ Verify the package before publishing

Click Publish to publish your package.

9. Select a location on the network\computer where you wish to create the CSV files. Now click Publish…

You will eventually see the following screen once done:



You have successfully published the package to the following location:
C:\extern\

The following types of problems have been found in your package:
65 Informational

Do you want to:

● Exit this wizard

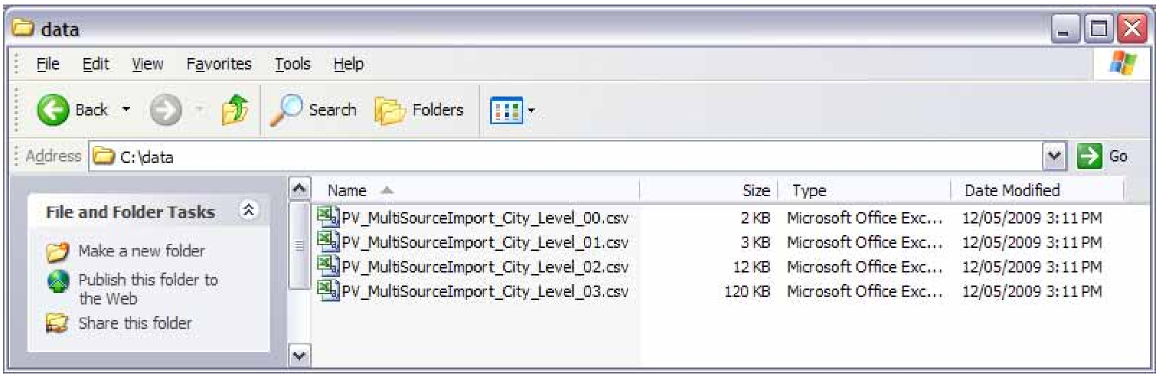
☑ View errors / warnings found in your package

○ Publish another package

Externalized query subjects:

Query Subject ID	Method	Path
[PV_MultiSourceImport].[City_Level_00]	csv	c:\data\PV_MultiSourceImport_City_Level_00.csv
[PV_MultiSourceImport].[City_Level_01]	csv	c:\data\PV_MultiSourceImport_City_Level_01.csv
[PV_MultiSourceImport].[City_Level_02]	csv	c:\data\PV_MultiSourceImport_City_Level_02.csv
[PV_MultiSourceImport].[City_Level_03]	csv	c:\data\PV_MultiSourceImport_City_Level_03.csv

Navigating to the chosen location, you should now see the CSV files created. One should be created for each level:



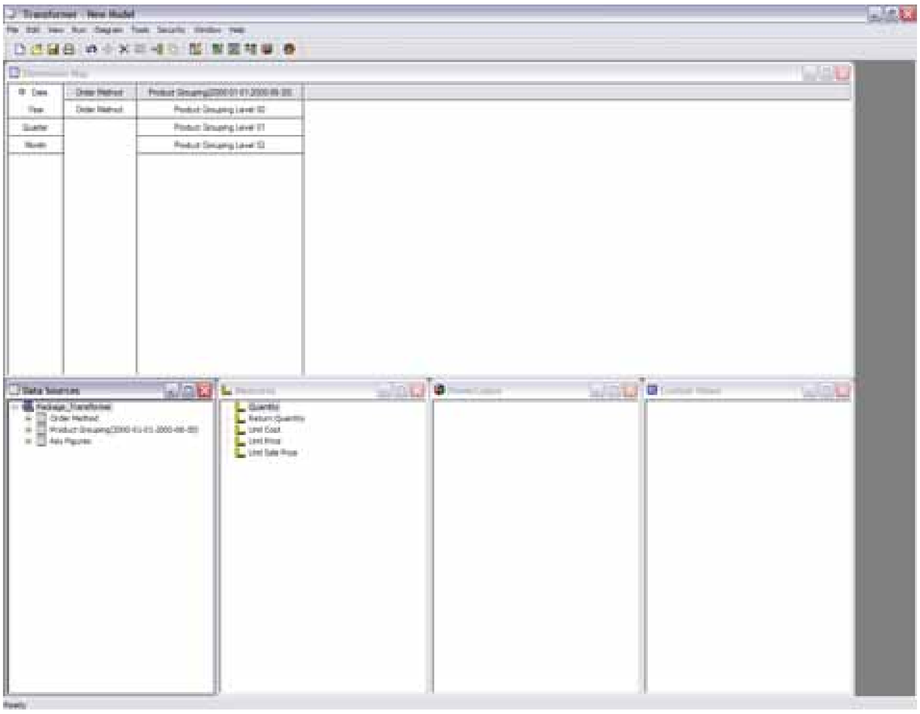
Address: C:\data

Name	Size	Type	Date Modified
PV_MultiSourceImport_City_Level_00.csv	2 KB	Microsoft Office Exc...	12/05/2009 3:11 PM
PV_MultiSourceImport_City_Level_01.csv	3 KB	Microsoft Office Exc...	12/05/2009 3:11 PM
PV_MultiSourceImport_City_Level_02.csv	12 KB	Microsoft Office Exc...	12/05/2009 3:11 PM
PV_MultiSourceImport_City_Level_03.csv	120 KB	Microsoft Office Exc...	12/05/2009 3:11 PM

6.3 Building a IBM Cognos 8 Transformer Hierarchy Using Externalized IBM Cognos 8 Framework Manager Data

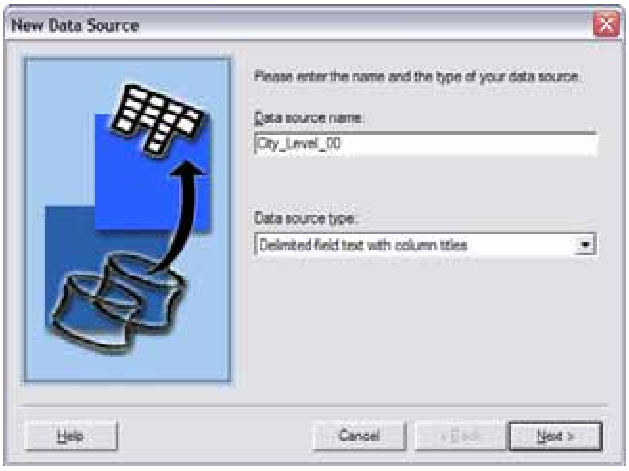
This section uses the CSV files extracted in the above Externalization section of this document to replace the automatically created City Geography Dimension obtained when creating the model from a package. Taking a close look at the CSV files created, you will notice that for each record retrieved both the “Member Unique Name” is included as well as the “Parent Unique Name”. This allows us to have a parent/child relationship we can rely on to create our hierarchy within Transformer.

Again using the model created by the package approach, we now need to open transformer and remove the City Geography Dimension and the City Geography query. Once completed your model should now look like this:



We now need to include the four CSV files as data sources for our model. To do this:

10. Right-click within the “Data Sources” window and select “insert data source”.
11. Choose “Delimited-field text with column titles” and give it any name:



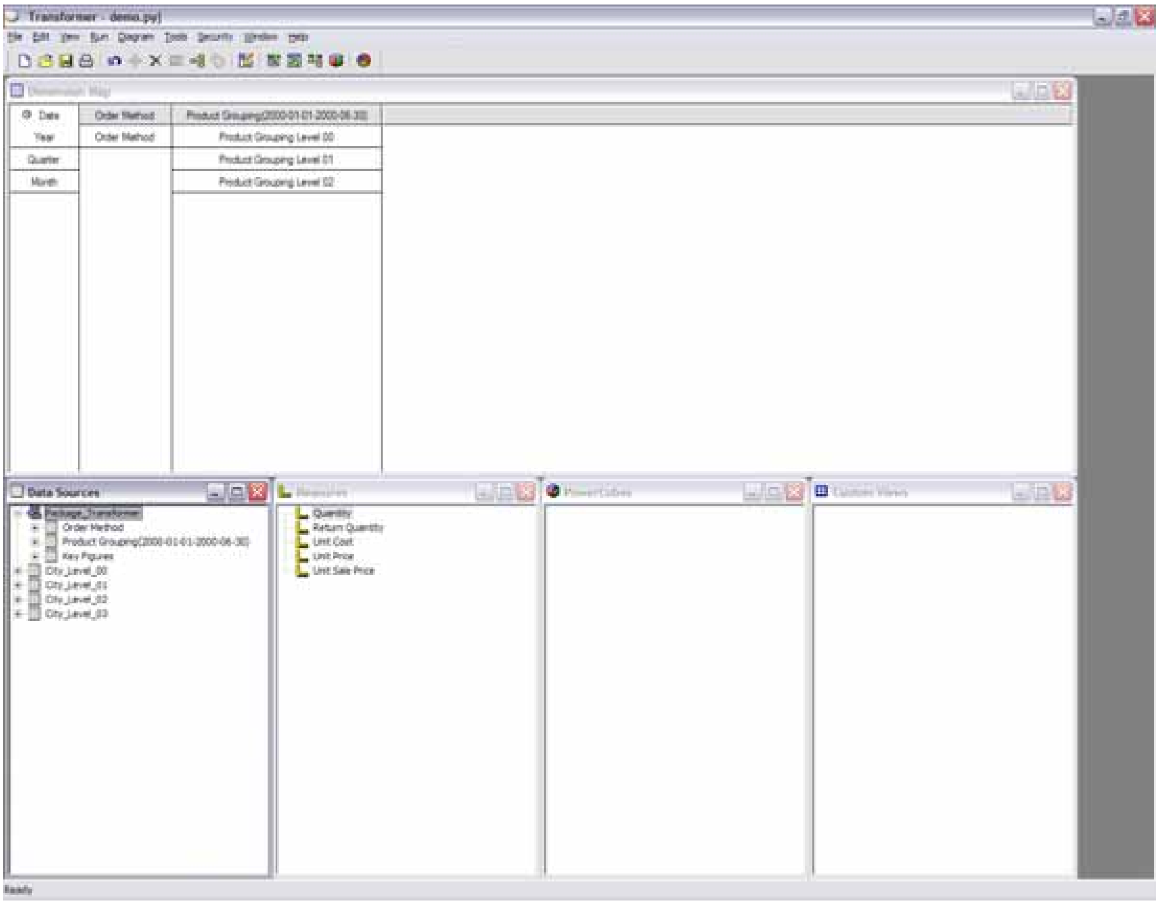
New Data Source

Please enter the name and the type of your data source.

Data source name: City_Level_00

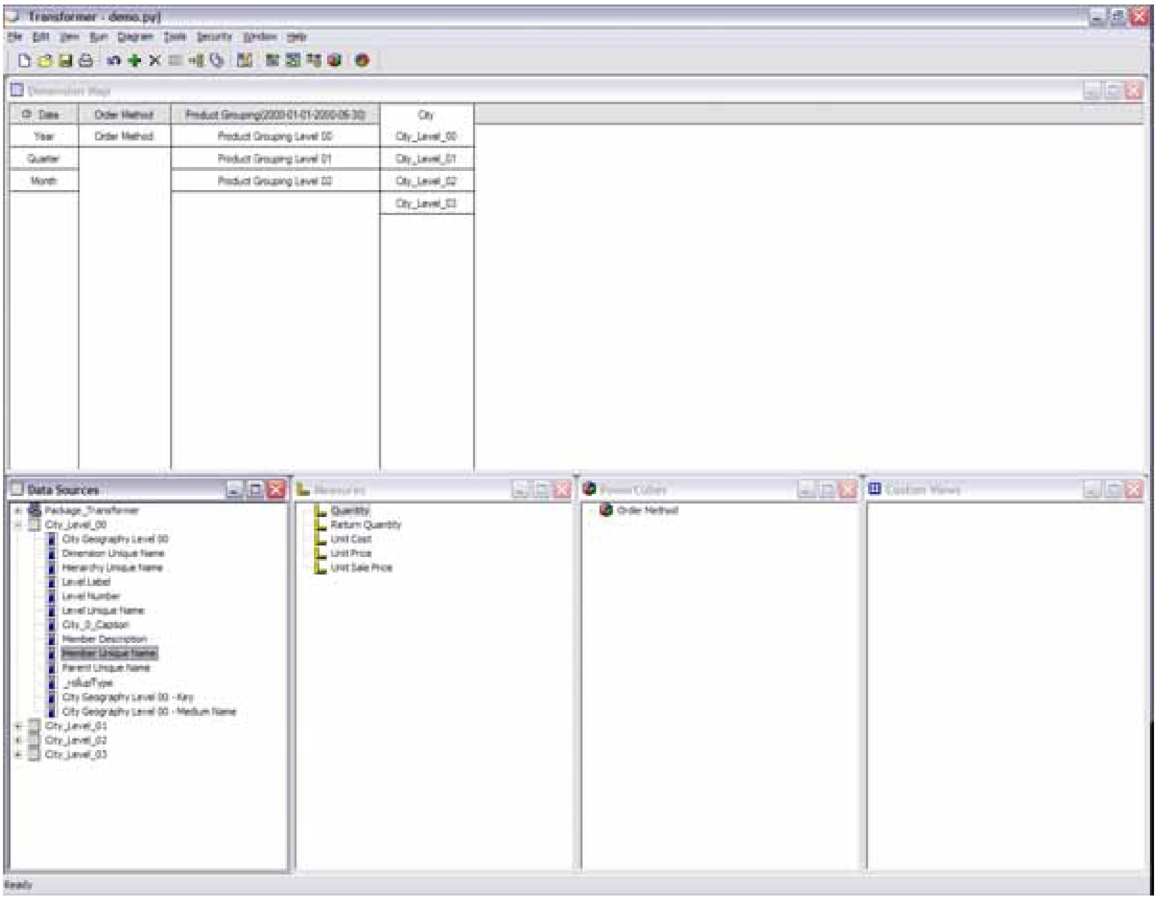
Data source type: Delimited-field text with column titles

12. Click on next, browse to the file location.
13. Click next and do not run auto-design.
14. Perform this task for every level. Once completed you should now have the following:



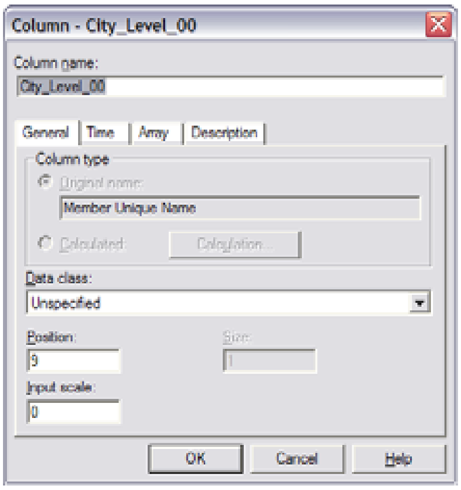
Date	Order Method	Product Grouping(2000-01-01-2000-06-30)
Year	Order Method	Product Grouping Level 00
Quarter		Product Grouping Level 01
Month		Product Grouping Level 02

You will need to define how each query relates to each other. By expanding each newly added query, you'll notice that each query contains the parent unique name along with the member unique name.



In order to establish a relationship between each level, we can rename the “Member Unique Name” column for each query to represent the actual level. For example, for query “City_Level_00”, rename “Member Unique Name” to “City_Level_00”. We do this by:

- Right-clicking on the “Member Unique Name” column.
- Select “Properties” and change the column name at the top:


Column - City_Level_00

Column name: City_Level_00

Column type: ● Original name: Member Unique Name ○ Calculated

Data class: Unspecified

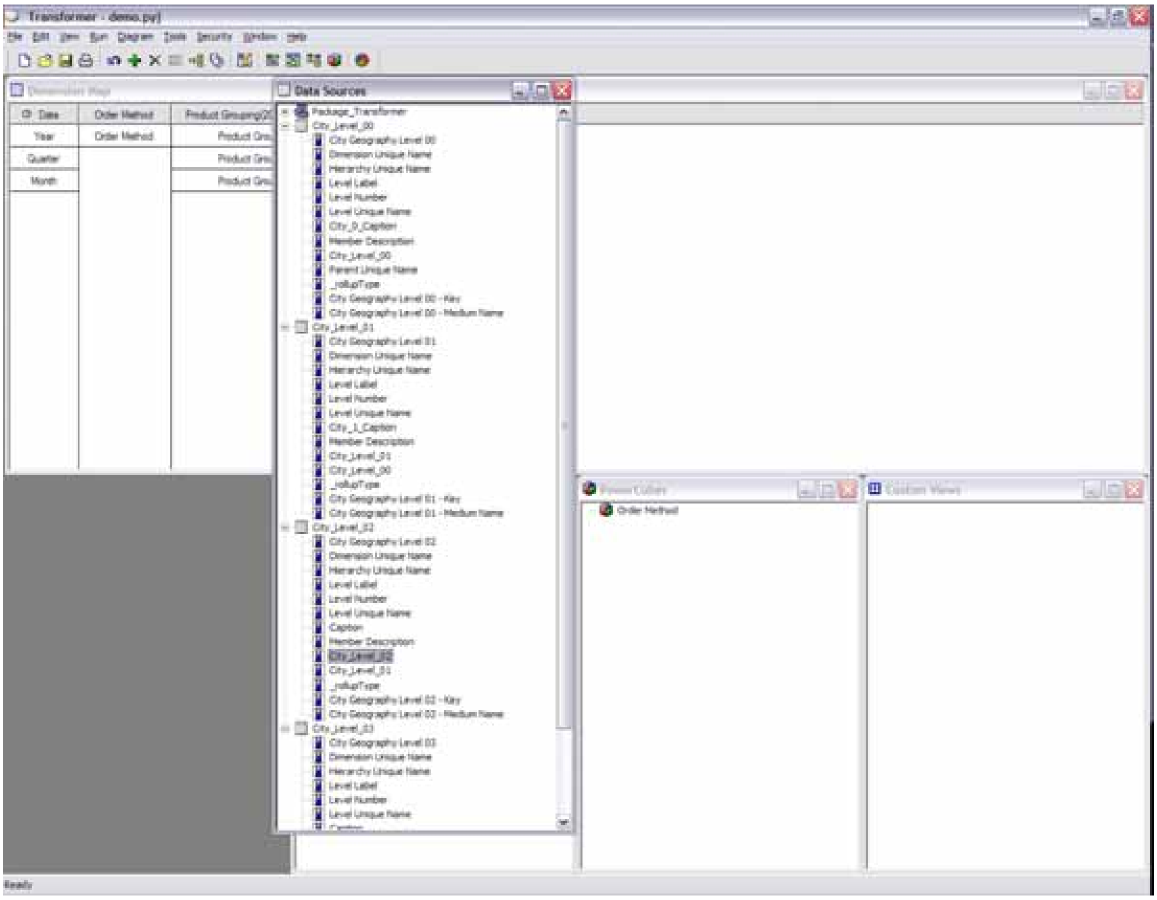
Position: 9

Size: 1

Input scale: 0

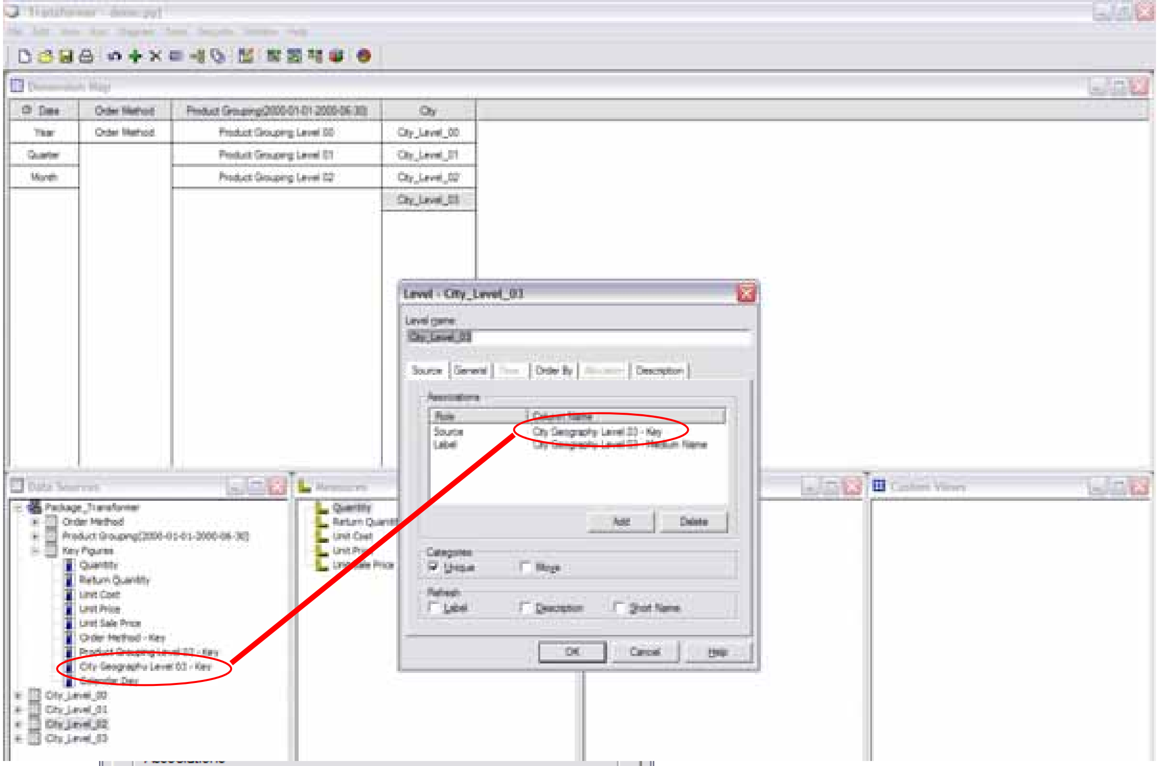
1. Now we expand query “City_Level_01” query, rename “Member Unique Name” first, and then also rename “Parent Unique Name” to “City_Level_00”.

This last step will create the relationship required between the two first queries which correspond to the top two levels. Perform these steps for each CSV query. Once done, you should end up with the following:



At this time, we can further prepare the queries prior to defining the dimension.

As each query contains the same column names (other then the ones we just modified), for each query you will need to find a column which contains the label you wish to use for reporting display purposes. The key may display “01” for the Country, however the “Medium Name” may contain “United States” which you may find more adequate for your reports. In many cases the caption can be utilized – however this depends on the design of your cube. A quick look through the CSV files should suffice to let you know which column is best suited. Once you’ve identified the appropriate column, simply rename the appropriate column as we did before for the unique names. In my case, the “Caption” column could be used for the two first levels, I needed to rename “Caption” of query “City_Level_00” to “City_0_Caption” and “Caption” of query “City_Level_01” to “City_1_Caption”. This step allowed me to differentiate one caption from the other. I did not need to perform this step for the bottom two queries, as I could use “City Geography Level 02 - Medium Name” and ” City Geography Level 03 - Medium Name” given they were named differently.

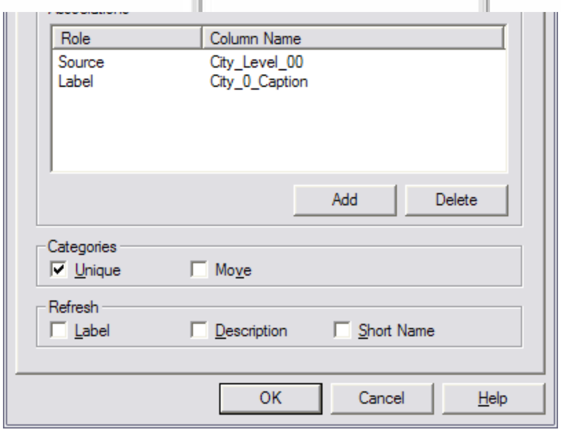

Level - City_Level_03

Level name: City_Level_03

Role	Column Name
Source	City Geography Level 03 - Key
Label	City Geography Level 03 - Medium Name

Categories: ☑ Unique ☐ Move

Refresh: ☐ Label ☐ Description ☐ Short Name


Role	Column Name
Source	City_Level_00
Label	City_0_Caption

Categories: ☑ Unique ☐ Move

Refresh: ☐ Label ☐ Description ☐ Short Name

4. Under associations, notice the Source points to “City_Level_00” column.
5. You now need to add your label, in my case I added “City_0_Caption”.
6. You also need to check the “Unique” check box.
7. Perform the same task for the top three levels.
8. The bottom level is a little different. We need to ensure that the column utilized in the fact query for this dimension is what is utilized as the source as illustrated below.

At this time, select ok and you should be able to build the cube.

7 Appendix A: Performance Case Studies

In order to achieve better performance when accessing the SAP Business Warehouse, a basic understanding of our reporting approach to this data source is recommended. The following case study will walk you through a simple example.

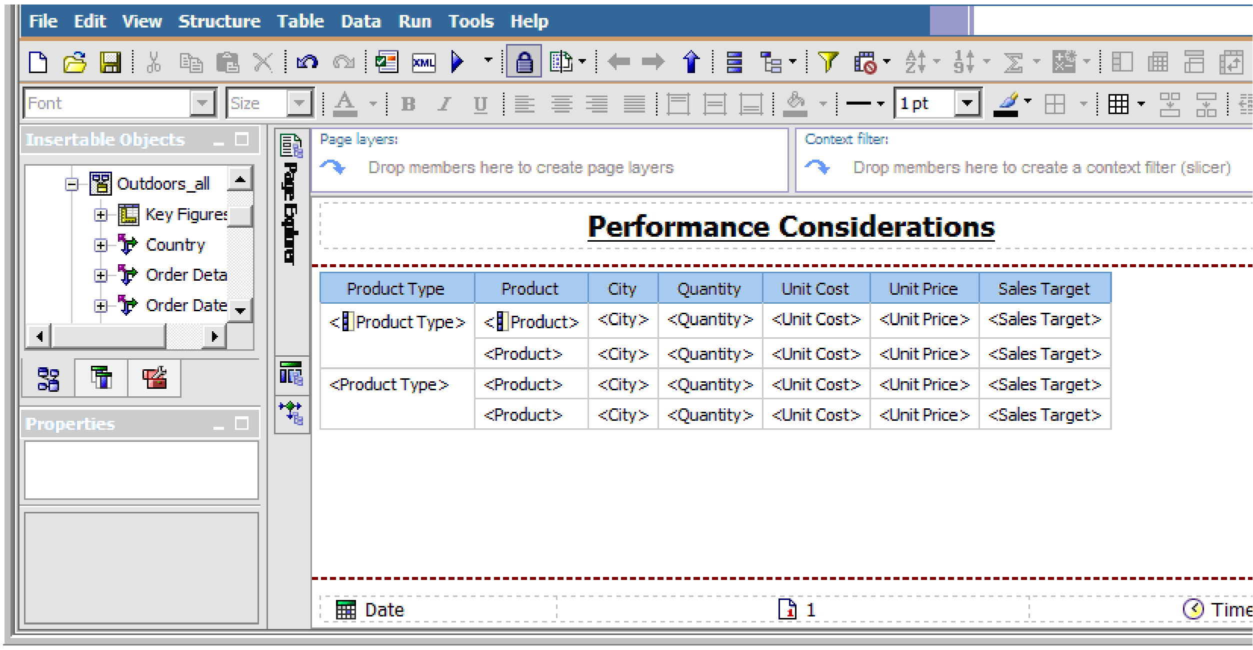
First and foremost, as highlighted in section 3 of this document, the basic methodology utilized by IBM Cognos 8 when querying the Business Warehouse is to decompose each query down to its simplest form. This method allows the creation of simple MDX sent to the SAP Business Warehouse thereby avoiding some of the pitfalls and performance degradations encountered when trying to maximize MDX performance for complicated queries.

We will tackle three basic concepts: usage of MDX for metadata retrieval, variable usage for filtering purposes, and finally reusability of metadata cache.

As a starting point, the screen capture below displays the basic report we wish to execute. The report consists of three (3) dimensions and 4 key figures. A filter was applied to restrict “Country” to United States. Taking the default configuration, let’s also assume the following pre-conditions:

Within the SAP Business Warehouse environment:

- Product Type Characteristic contains 225 distinct members (150 with associated fact values)
- Product Characteristic: 1000 distinct members (500 with associated facts)
- City Characteristic: 300 Distinct members (250 with associated facts)



Page layers:
Drop members here to create page layers

Context filter:
Drop members here to create a context filter (slicer)

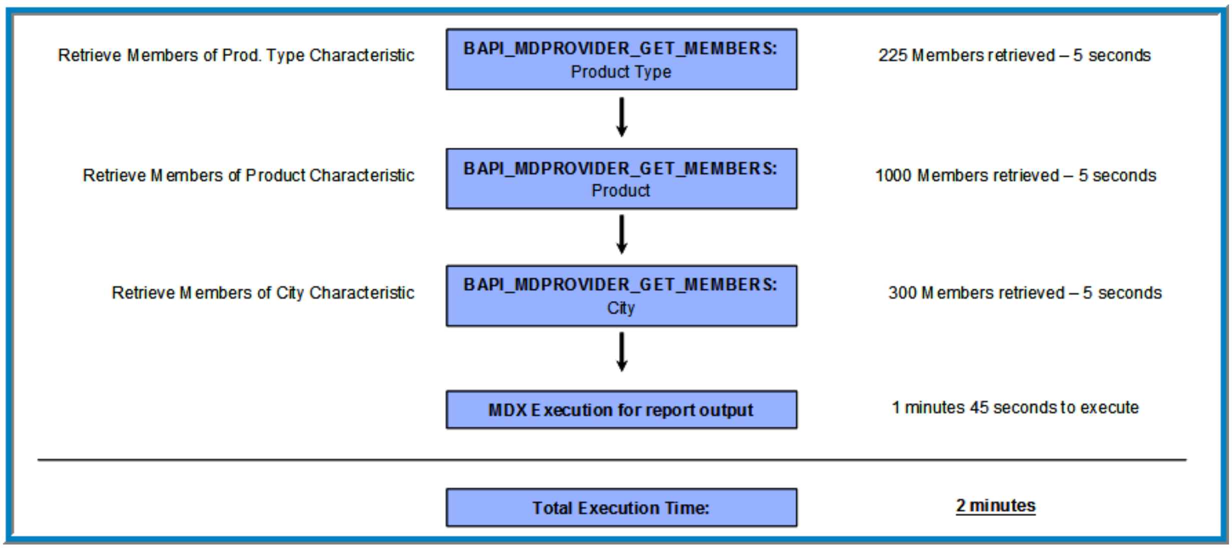
Performance Considerations

Product Type	Product	City	Quantity	Unit Cost	Unit Price	Sales Target
<Product Type>	<Product>	<City>	<Quantity>	<Unit Cost>	<Unit Price>	<Sales Target>
	<Product>	<City>	<Quantity>	<Unit Cost>	<Unit Price>	<Sales Target>
<Product Type>	<Product>	<City>	<Quantity>	<Unit Cost>	<Unit Price>	<Sales Target>
	<Product>	<City>	<Quantity>	<Unit Cost>	<Unit Price>	<Sales Target>

Date 1 Time

7.1 Scenario 1 - Base Query with no Optimization (Runtime 2 min)

As a starting point, we executed the report and its execution completed in 2 minutes. Graphically the execution of this report can be represented as such:



Retrieve Members of Prod. Type Characteristic — BAPI_MDPROVIDER_GET_MEMBERS: Product Type — 225 Members retrieved – 5 seconds

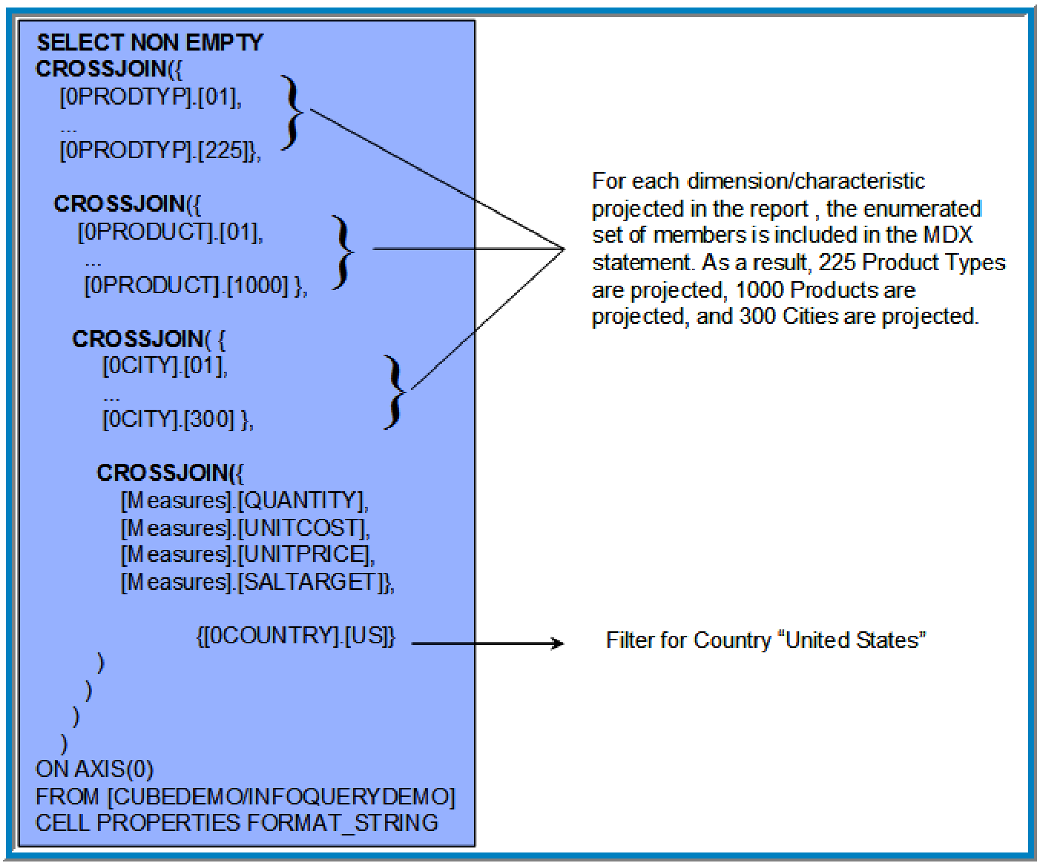
Retrieve Members of Product Characteristic — BAPI_MDPROVIDER_GET_MEMBERS: Product — 1000 Members retrieved – 5 seconds

Retrieve Members of City Characteristic — BAPI_MDPROVIDER_GET_MEMBERS: City — 300 Members retrieved – 5 seconds

MDX Execution for report output — 1 minutes 45 seconds to execute

Total Execution Time: **2 minutes**

The final MDX sent to Business Warehouse would have the following form:



```
SELECT NON EMPTY
CROSSJOIN({
  [0PRODTYP].[01],
  ...
  [0PRODTYP].[225]},

   CROSSJOIN({
     [0PRODUCT].[01],
      ...
      [0PRODUCT].[1000] },

      CROSSJOIN( {
        [0CITY].[01],
        ...
        [0CITY].[300] },

         CROSSJOIN({
           [Measures].[QUANTITY],
           [Measures].[UNITCOST],
           [Measures].[UNITPRICE],
           [Measures].[SALTARGET]},

                 {[0COUNTRY].[US]}
          )
         )
        )
       )
ON AXIS(0)
FROM [CUBEDEMO/INFOQUERYDEMO]
CELL PROPERTIES FORMAT_STRING
```

For each dimension/characteristic projected in the report , the enumerated set of members is included in the MDX statement. As a result, 225 Product Types are projected, 1000 Products are projected, and 300 Cities are projected.

Filter for Country "United States"

We now see that the bulk of the execution of the report is taken by the execution of the final MDX (1 minute 45 seconds). Each metadata BAPI calls executed using BAPI_MDPROVIDER_GET_MEMBERS is relatively fast (5 seconds in our example).

7.2 Scenario 2 - Query with UseMDX Optimization (Runtime 1min 30 sec)

The first step required to maximize this report would be to attempt to reduce the size of the final MDX sent. We can accomplish this by retrieving the members using MDX instead of BAPI_MDPROVIDER_GET_MEMBERS. This will ensure that only members with associated facts are included in the final MDX. There are two methods to accomplish this task.

The first method is more general in nature. A configuration setting exists which evaluates the cardinality of the projected dimension. If the cardinality exceeds a threshold, then MDX will be utilized to fetch the members. The cardinality for each dimension is extracted by Framework Manager during import. This setting exists in the sapbw_config.xml file and can be modified by changing the value for:

```
<parameter name="useMDXToRetrieveMembersLimit" value="500"/>
```

Please note that by default this parameter is set to 100,000 which should be reduced to a value between 200 to 5000. While this range is high, all SAP Business Warehouse environments differ and some testing may be required in order to find the correct limit which suits your environment. Setting the value to 500 as a starting point would be recommended.

The second method is more targeted and requires administrators to identify specific dimensions (characteristics), hierarchies, or even members against which you may want to make use of MDX to retrieve members. Given most large dimensions are taken care of using the first method, this would target dimensions where cardinality falls below the above threshold however have a large differences between all existing members and members actually containing posted values in various circumstances (variables for example). This setting exists in the sapbw_config.xml file and can be modified by changing the value for:

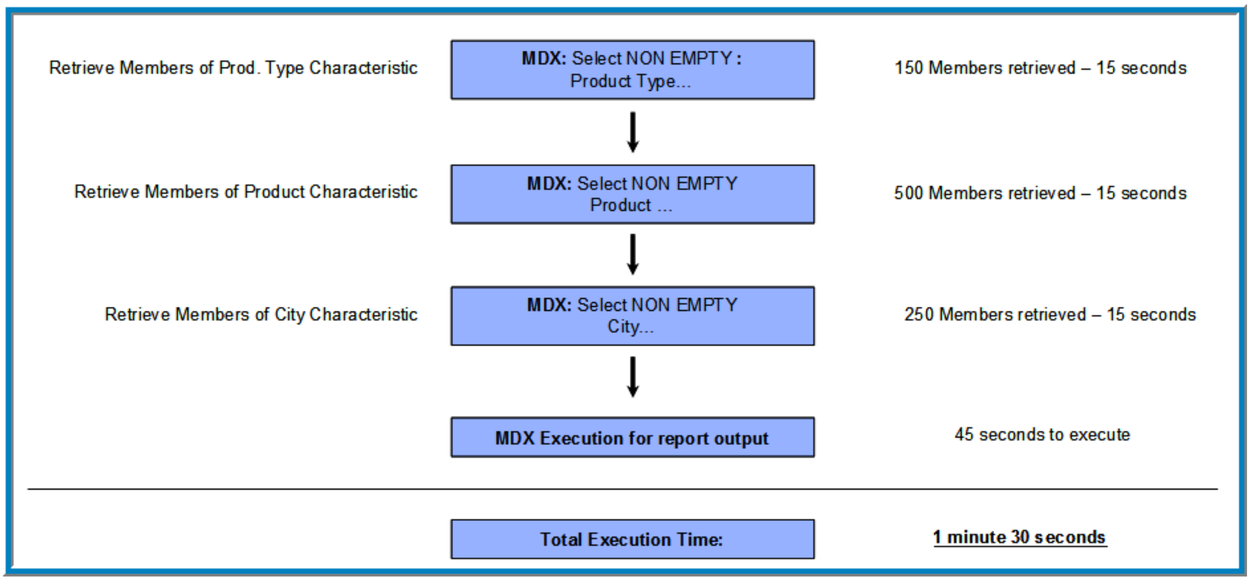
```
<parameter name="UseMDXToRetrieveMembersFor" value="[0DIMX]”/>
```

Why shouldn’t we make all metadata calls using MDX then? In most environments, MDX metadata statements tend to be more expensive and take more time to execute. A good balance between the two usually offers the best performance.

Given what we’ve learned from the parameters above, we have decided to add the following configuration setting:

```
<parameter name="useMDXToRetrieveMembersLimit" value="200"/>
```

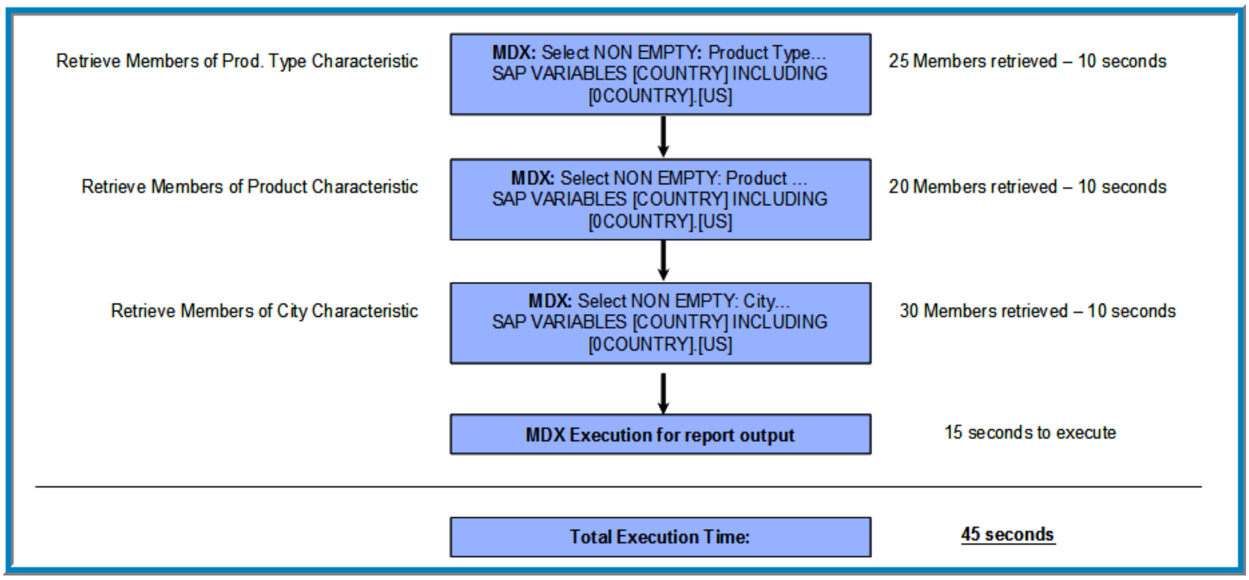
This will force the query engine to fetch the members with MDX thereby returning only members with posted data values. Find below the graphical representation of this change.


Retrieve Members of Prod. Type Characteristic
MDX: Select NON EMPTY : Product Type...
150 Members retrieved – 15 seconds
Retrieve Members of Product Characteristic
MDX: Select NON EMPTY Product ...
500 Members retrieved – 15 seconds
Retrieve Members of City Characteristic
MDX: Select NON EMPTY City...
250 Members retrieved – 15 seconds
MDX Execution for report output
45 seconds to execute
Total Execution Time:
1 minute 30 seconds

While this has increased performance by 30 seconds, there is still room for improvement.

7.3 Scenario 3 - Query with UseMDX Optimization and Business Warehouse Variable (Runtime 45 sec)

This report currently uses a local filter was defined for Country. While this filter is sent to SAP Business Warehouse in the final MDX as an extra axis, it is not considered hen retrieving members during the metadata phase. Alternatively, one can make use of BW variables instead. When combined with MDX metadata fetches, this will have the added benefit of further filtering the members returned for each characteristic. The impact on the final data MDX sent is often greatly improved.


Retrieve Members of Prod. Type Characteristic
MDX: Select NON EMPTY: Product Type... SAP VARIABLES [COUNTRY] INCLUDING [0COUNTRY].[US]
25 Members retrieved – 10 seconds
Retrieve Members of Product Characteristic
MDX: Select NON EMPTY: Product ... SAP VARIABLES [COUNTRY] INCLUDING [0COUNTRY].[US]
20 Members retrieved – 10 seconds
Retrieve Members of City Characteristic
MDX: Select NON EMPTY: City... SAP VARIABLES [COUNTRY] INCLUDING [0COUNTRY].[US]
30 Members retrieved – 10 seconds
MDX Execution for report output
15 seconds to execute
Total Execution Time:
45 seconds

7.4 Important Note on Metadata Reuse:

While MDX metadata fetch can greatly improve performance, another aspect needs to be considered - reusability. The IBM Cognos 8 query engine will attempt to reuse some of the members retrieved through metadata calls as much as possible for subsequent execution. If the cube or query is the same, if the user is the same, if the BiBus process is the same and if the subsequent execution request is performed within 5 minutes (connection timeout) then the IBM Cognos 8 engine will reuse the members in memory – however the following two rules apply:

- Member retrieved using BAPI_MDPROVIDER_GET_MEMBERS can be reused
- Member retrieved using MDX can be reused ONLY when NO variables are used

Reusability has greatly improved performance of some reports, especially with many reports targeting identical hierarchies.

While this simple example simply scratches the surface of performance considerations, a good understanding of the concept discussed here will serve as a solid foundation of all subsequent design approaches.