IBM Cognos Dynamic Cubes Version 10.2.0

User Guide



Note

Before using this information and the product it supports, read the information in "Notices" on page 121.

Product Information

This document applies to IBM Cognos Business Intelligence Version 10.2.0 and may also apply to subsequent releases. To check for newer versions of this document, visit the IBM Cognos Information Centers (http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp).

Licensed Materials - Property of IBM

© Copyright IBM Corporation 2012.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Introduction	vi i
Chapter 1. Cognos Dynamic Cubes overview	1
Chapter 2. Cognos Dynamic Cubes workflow	5
Chapter 3. Dimensional metadata and dynamic cubes	11
Dimensional metadata	11
Dimensions	11
Hierarchies	11
Parent-child hierarchies	15
Levels	16
Joins	18
Attributes	18
	19
Measures	21
	24
	21
Chapter 4. Getting started with Cognos Cube Designer	31
Introduction to Cognos Cube Designer	31
Import metadata.	33
Importing metadata from a Content Manager data source	33
Managing a project.	34
Validate a project and individual objects.	35
Chanter 5 Dimensional metadata modeling	37
Model dimensions	
Model hierarchies	
Defining a hierarchy	
Model levels	
Defining a level	
Defining a level unique key	43
Defining the member sort order	
Model parent-child hierarchies	
Defining a parent-child hierarchy	46
Browsing members	47
Chapter 6. Dynamic cube modeling	49
Model a dynamic cube	49
Defining a dynamic cube based on a relational table.	50
Defining a dynamic cube manually	51
Model measures.	51
Defining a measure-to-dimension join	54
Deploying and publishing dynamic cubes	54
Chapter 7. Advanced dynamic cube modeling	57
Calculated members	
Calculated members examples	58
Defining a calculated member	62
Model relative time dimensions	63
Defining a relative time dimension	63

Examples of level current period expressions	. 64 . 65 . 68
Selecting the design language and supported locales	. 68
Adding multiple locale names to metadata objects and dynamic cube objects	. 68
Adding support for multiple locales to members and attributes	68
Use Cognos Framework Manager to publish multilingual packages	. 60
	. 07
Chapter 8. Aggregate cube modeling	. 71
Model aggregate cubes	. 71
Defining a simple aggregate cube automatically	. 72
Defining a simple aggregate cube manually	. 73
Defining an aggregate cube with a parent-child dimension	. 74
Filtering data using an aggregate slicer	. 75
Chapter 9 Virtual cube modeling	77
Madal virtual dimensiona	. //
Model virtual dimensions	. 70
	. 79
Viewing virtual levels	. 81
Model virtual members	. 81
Model virtual measures	. 83
Chapter 10. Defining security	. 87
Security filters	87
Defining a security filter	. 07
Definiting a second function of the second	. 09
Defining a security view	. 90
Chapter 11. Cognos Dynamic Cubes administration	. 91
Roles and required capabilities	. 92
General Properties for dynamic cube data sources	. 93
Access account property for dynamic cubes	. 94
Create trusted credentials.	. 96
Create a signon	. 96
Setting the data access account	. 97
Assign users and groups to security views	. 97
Set access permissions for an entry	98
Configure a dynamic cube for the query service	. 99
Configuring query service properties	100
Trines of eaches used by dynamic cubes	102
Dypes of calles configuration settings	102
Adding durantic cube configuration settings	105
Adding dynamic cubes to the query service	105
Defense a dynamic cube from the query service.	. 105
Enable Workload log for Aggregate Advisor	. 106
Managing dynamic cubes in the query service	. 106
	. 100
Appendix A. Accessibility features	111
Accessibility features in Cognos Cube Designer	. 111
Keyboard shortcuts for Cognos Cube Designer	. 111
Appendix B. Report considerations	115
	115
	. 115
Kelative time calculated members in reports	. 117
Removal of padding members from reports	. 118
Appendix C. Troubleshooting	119
BMT-MD-6527 error when generating a dynamic cube based on a relational table	. 119

Possible overflow In-memory aggreg	in me gates fa	asure ail to	attri load	but	es.					 									. 1 . 1	.19 .20
Notices					-	 	•					•	•		-			-	12	21
Index						 													1:	25

Introduction

This document is intended for use with IBM[®] Cognos[®] Dynamic Cubes. It describes the processes required to model dimensional metadata and to create dynamic cubes to use as data sources in the Content Manager.

Audience

The following knowledge and experience can help you to use the product.

- Knowledge of OLAP concepts.
- Knowledge of your business requirements.
- An understanding of the structure of your data sources.
- Experience of installing and configuring applications.

Finding information

To find IBM Cognos product documentation on the web, including all translated documentation, access one of the IBM Cognos Information Centers (http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp). Release Notes are published directly to Information Centers, and include links to the latest technotes and APARs.

You can also read PDF versions of the product release notes and installation guides directly from IBM Cognos product disks.

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. IBM Cognos Dynamic Cubes has accessibility features. For information about these features, see the accessibility section in this document.

IBM Cognos HTML documentation has accessibility features. PDF documents are supplemental and, as such, include no added accessibility features.

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Samples disclaimer

The Great Outdoors Company, GO Sales, any variation of the Great Outdoors name, and Planning Sample depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values is coincidental. Other sample files may contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample applications. Product names referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.

Chapter 1. Cognos Dynamic Cubes overview

In a dimensional data warehouse, you model relational database tables using a star or snowflake schema. This type of data warehouse differs from a traditional OLAP model in the following ways:

- It stores information about the data in fact and dimension tables rather than in proprietary OLAP data structures.
- It describes the relationships within the data using joins between the dimension and fact tables, the collection of dimension keys in a fact table, and the different attribute columns in a dimension table.

IBM Cognos Dynamic Cubes adds an in-memory relational OLAP component to the dynamic query mode server to provide a multidimensional view of a relational data warehouse with accelerated performance. You can then perform OLAP analysis using the Cognos Dynamic Cubes server.

Cognos Dynamic Cubes differs from Cognos dimensionally-modeled relational (DMR) data sources for the following reasons:

- It provides increased scalability and the ability to share data caches between users for better performance.
- It allows you to a create a dynamic cube data source that is pre-loaded with dimensions.
- It allows for a richer set of dimensional modeling options and the explicit management of the member and data caches of a dynamic cube.

The benefits of Cognos Dynamic Cubes can be achieved only when using a dynamic cube as a data source. To use a dynamic cube as a data source, you must use the dynamic query mode.

Cognos Dynamic Cubes introduces a performance layer in the Cognos query stack to allow low-latency, high-performance OLAP analytics over large relational data warehouses. By using the power and scale of a relational database, Cognos Dynamic Cubes can provide OLAP analytics over terabytes of warehouse data.

Cognos Dynamic Cubes uses the database and data cache for scalability, and also uses a combination of caching, optimized aggregates (in-memory and in-database), and optimized SQL to achieve performance. The Cognos Dynamic Cubes solution includes the following characteristics:

- It uses simple, multi-pass SQL that is optimized for the relational database.
- It is able to minimize the movement of data between the relational database and the Cognos Dynamic Cubes engine.

This data control is achieved by caching only the data that is required and by moving appropriate calculations and filtering operations to the database. At run time, only fact data is retrieved on demand.

• It is aggregate-aware, and able to identify and use both in-memory and in-database aggregates to achieve optimal performance.

Aggregate awareness (aggregates tables that are created in the database and modeled into a dynamic cube) uses specialized log files to allow the dynamic query mode server to decompose queries to take advantage of the aggregate tables.

• It optimizes aggregates (in-memory and in-database) using workload-specific analysis.

Aggregate Advisor, part of IBM Cognos Dynamic Query Analyzer, analyzes the performance of dynamic cubes using log files and provides suggestions for improving cube performance.

• It can achieve low latency over large data volumes, such as billions of rows or more of fact data and millions of members in a dimension.

By using virtual cubes, companies can still present the complete view of the data, but need to refresh only smaller sets of data, leaving pre-cached query results for larger static sets. Users experience better performance for queries run against pre-cached results.

Evaluating your data

Before starting to model a cube, it is important to understand how your data affects the processing in IBM Cognos Cube Designer.

Referential integrity in data warehouses

Most databases today support referential integrity. However, it is typically turned off or is made declarative and instead is enforced during extract, transform, and load (ETL) processing. Erroneous modifications made to the data during or outside of the ETL process can create cases where a fact table has no matching dimension records.

Each data point in a dynamic cube is defined by a member from each dimension in the cube. If a value is required for some data point, then the SQL generated by Dynamic Cubes does not specify a filter on the table associated with a particular dimension if the member of that dimension is the All member. This allows for smaller SQL queries and also faster executing queries.

When a dimension is in scope, the join between the fact and dimension table is specified in the SQL query and the dimension is filtered by an explicit set of dimension key values. When the member of a dimension is the All member, dynamic cubes will not specify a filter for that dimension. All records are included, even records with invalid or missing dimension key values. This difference causes a discrepancy between values, depending upon which dimensions are involved in a query.

Even if your fact records have invalid or unknown dimension key values, you should validate your records before implementing Cognos Dynamic Cubes. Run an SQL query similar to the following for each dimension in a dynamic cube. This determines if there are any fact records with invalid dimension key values. Any returned data is the set of invalid dimension key values. If no data is returned, there are no referential integrity errors.

select distinct FACT.Key
from FactTable FACT
where not exists
(select *
 from DimensionTable DIM
 where DIM.Key = FACT.Key)

The SQL query can also be used as a subquery, to obtain the full set of records from the fact table.

If your fact table might contain records with invalid or unknown dimension key values, a common practice is to create a row in the dimension table to represent these dimension keys. New fact rows with invalid or unknown dimension key values can be assigned this dimension key value until the fact records and the dimension table can be updated with correct information. With this practice, records with problematic dimension key values are visible, regardless of which dimensions are involved in a report or analysis.

You should also validate snowflake dimensions.

You may have a situation where tables in a snowflake dimension are joined on a column for which the outer table did not contain values for rows in the inner table. In this case, the inner dimension table joins to the fact table, but the outer dimension table does not join to the inner dimension table.

To ensure that snow flake dimensions do not have this type of referential integrity error, run an SQL query similar to the following. In this example, the dimension is built from two tables, D1_outer and D2_inner. D2_inner is joined to the fact table. Key is the column on which the two dimension tables are joined.

select distinct INNER.Key
from D2_inner INNER
where not exists
(select *
 from D1_outer OUTER
 where OUTER.Key = INNER.Key)

Chapter 2. Cognos Dynamic Cubes workflow

IBM Cognos Dynamic Cubes brings faster, more powerful, cube performance into the IBM Cognos reporting environment. Cognos Dynamic Cubes is used to improve access to large sets of data.

Figure 1 illustrates the relationship between the main activities performed using IBM Cognos Dynamic Cubes and the corresponding tools. IBM Cognos Cube Designer provides dynamic cube design and modeling capability. The Administration Console is used to deploy and manage the cube data. The dynamic query mode (DQM) server maintains the cube data. Studio applications use the data in reporting environments. In addition, various tools, such as Dynamic Query Analyzer, are used to analyze and optimize the data as necessary.



Figure 1. Relationships between Cognos Dynamic Cubes activities and tools

The following diagram shows the five major steps in a typical process flow, showing the users who are involved at each step.



Figure 2. Cognos Dynamic Cubes typical process flow

Analyze the data

Before installing IBM Cognos Dynamic Cubes, the modeler and relational database administrator prepare for project implementation by completing the following tasks:

- Determining whether the data is a good candidate for Cognos Dynamic Cubes.
- Reviewing prerequisites to ensure correct implementation.

For more information about assessing your data and understanding prerequisites, see Chapter 1, "Cognos Dynamic Cubes overview," on page 1.

Design and model a dynamic cube

The system analyst determines high-level business requirements and evaluates cube design against reporting requirements.

The modeler creates a basic dynamic cube, adds features to satisfy the business requirements and ensures that the cube is available to IBM Cognos Administration. Within IBM Cognos Cube Designer, the modeler performs tasks such as:

- Importing relational metadata to use as the basis for dynamic cube design.
- Designing dynamic, aggregate, and virtual cubes.
- Setting cube-level security for hierarchies and measures.
- Publishing the dynamic cube.

For more information about designing and modeling dynamic cubes, see the following topics:

- "Import metadata" on page 33
- "Model a dynamic cube" on page 49
- "Calculated members" on page 57
- Chapter 8, "Aggregate cube modeling," on page 71

- Chapter 9, "Virtual cube modeling," on page 77
- Chapter 10, "Defining security," on page 87
- "Deploying and publishing dynamic cubes" on page 54

Optionally, the modeler runs Aggregate Advisor for recommendations regarding the dynamic cube design. For information about Aggregate Advisor, see the *IBM Cognos Dynamic Query Analyzer User Guide*.

Deploy and manage a dynamic cube

After dynamic cubes are published to Content Manager, the Administrator handles the initial configuration and subsequent management. Within IBM Cognos Administration, the administrators perform tasks such as:

- Setting the Access Account property in the Administration Console.
- Assigning users, groups and roles to security views.
- Assigning a server group to the dispatcher.
- Assigning a routing set to all packages associated with a dynamic cube.
- Creating a routing rule to route queries for the routing set to the server group.
- Configuring the query service and the dynamic cube for a dispatcher.
- Starting the dynamic cube for initial use.
- Refreshing the dynamic cube, as necessary.
- Stopping the dynamic cube (soft or hard stop) while the data warehouse is being updated.
- Optionally, turning on logging. Log files are required to optimize the cube.
- Clearing workload logs.

For more information about deploying and managing dynamic cubes, see Chapter 11, "Cognos Dynamic Cubes administration," on page 91 and the *IBM Cognos Administration and Security Guide*.

Run reports using dynamic cube data

The report author uses the dynamic cube as a data source in reporting applications.

Optimize a dynamic cube

To optimize individual cube performance, the administrator can monitor the metrics of the dynamic cubes, and make changes, if necessary, to the cube configuration.

To further optimize performance, the system analyst can run a series of reports that are a representative workload against the dynamic cube. The resulting workload logs are used by Aggregate Advisor to return recommendations for additional in-memory and in-database aggregates. The analyst can also examine request execution log files in the Dynamic Query Analyzer. The log files help the analyst understand where time is spent within the dynamic cube engine, the type of SQL queries that are posed, how much time is spent executing the queries, and how many rows of data are returned. For information about Aggregate Advisor, see the *IBM Cognos Dynamic Query Analyzer User Guide*.

When you save in-memory aggregate recommendations to the content store, they are loaded automatically the next time the dynamic cube is started.

For in-database aggregate recommendations, the database administrator creates the aggregate tables in the database and the modeler uses IBM Cognos Cube Designer to model and publish the dynamic cube. For more information, see Chapter 8, "Aggregate cube modeling," on page 71.

After new aggregates are published by the modeler, the administrator sets the in-memory aggregate size and restarts the dynamic cube to use new aggregates.

For detailed information, see Chapter 11, "Cognos Dynamic Cubes administration," on page 91.

Workflow summary

To prepare for and manage project implementation, there are tasks external to the IBM Cognos software and tasks performed using IBM Cognos software. The following table shows a summary of responsibilities in each step of the workflow.

Workflow	Responsibilities	Tools	Role		
Analyze, configure	Gather requirements and best practices.		Solutions architect		
	practices.				
	Prepare an overall design.				
	Perform hardware assessments.				
Configure	Determine operating system administration changes.	O/S command tools, system administration console	System administrator		
	Perform middleware installation and maintenance.				
Analyze, model	Design the database physical model.	Modeling tools, document/ presentation software	Data architect		
	Design the multi-dimensional model.	1			
Analyze, model	Gather business requirements.	Modeling tools, document/ presentation software	Business/Application consultant		
	Design the logical model.	presentation software			
	Prepare the security definition.				
Model, optimize	Design dynamic cubes.	IBM Cognos Cube Designer, IBM Cognos Dynamic	Cognos modeler		
	Define security rules and views.	Query Analyzer			

Table 1. Workflow responsibilities by role

Workflow	Responsibilities	Tools	Role
Manage, deploy	Configure and manage dynamic cubes.	Cognos Administration Console, Cognos Dynamic Query Analyzer	Cognos administrator (system)
Manage, deploy	Manage security of IBM Cognos objects, including dynamic cubes.	Cognos Administration Console	Cognos administrator (security)
Manage, deploy	Manage IBM Cognos data sources. Assign users to security views.	Cognos Administration Console	Cognos administrator (directory)
Optimize, Model	ptimize, Model Evaluate overall performance. Run Aggregate Advisor.		Cognos administrator (system)
Run	Author reports, analyses or dashboards for use by collection of users	Cognos BI client applications	Cognos report author
Configure, Model, Optimize	Implement database updates Perform database maintenance such as extract, transform and load (ETL) processes, backup and recovery.	Database administration console, ETL tools	Database administrator

Table 1. Workflow responsibilities by role (continued)

Chapter 3. Dimensional metadata and dynamic cubes

Understanding concepts relating to dimensional metadata and dynamic cubes helps you to plan and create effective dynamic cubes.

Dimensional metadata

In IBM Cognos Dynamic Cubes, dimensional metadata refers to dimensions and hierarchies. You can create commonly used dimensional metadata independent of any dynamic cubes in a project. The appropriate dimensional metadata can then be shared by one or more cubes in a project.

You can also create dimensional metadata that is connected to a specific dynamic cube.

Dimensions

In IBM Cognos Dynamic Cubes, you can create two types of dimensions: regular and parent-child.

A regular dimension is a collection of hierarchies and levels that describe one aspect of a measure, such as Customer or Product. This type of dimension can contain one or more hierarchies. A hierarchy uses levels to describe the relationship and order of dimension attributes. Related attributes and the joins that are required to group these attributes are defined in the dimension. For more information, see "Hierarchies."

A parent-child dimension contains dimension data based on a recursive relationship and is not level-based. This type of dimension can contain only a single parent-child hierarchy. For more information, see "Parent-child hierarchies" on page 15.

Cognos Dynamic Cubes also supports degenerate dimensions. A degenerate dimension is a regular dimension that is based on data in the fact table. When modeling a dynamic cube based on a degenerate dimension, you do not need to specify a measure-to-dimension join.

Hierarchies

A hierarchy uses levels to describe the relationship and order of dimension attributes. For example, a Customer dimension might contain a Region hierarchy.

For more information about attributes and levels, see "Attributes" on page 18 and "Levels" on page 16.

IBM Cognos Dynamic Cubes supports balanced, unbalanced, and ragged hierarchies. Padding members are used to balance unbalanced and ragged hierarchies, so they appear as balanced hierarchies in the IBM Cognos studios. For more information, see "Padding members" on page 14.

Multiple hierarchies

Multiple hierarchies can be defined for dimensions containing level-based hierarchies.

You create multiple hierarchies for a dimension when you want to organize dimension members in different ways. For example, in a Time dimension, you can create hierarchies for Calendar year and Fiscal year.

Because dimension members in separate hierarchies can be used to represent the same entity, each hierarchy should contain the same lowest level members. For example, in a Time dimension, the Calendar hierarchy might have Year, Month, and Day levels. The Fiscal hierarchy might have Year, Quarter, and Day levels. The lowest level in both dimensions is the Day level.

Hierarchies that are modeled using a shared level can be optimized during query execution to remove non-intersecting values. To do this, you must ensure the **Remove non-existent tuples** property is set in a dynamic cube. For more information, see "Model a dynamic cube" on page 49.

Balanced hierarchies

In a balanced hierarchy, the branches of the hierarchy all descend to the same level. The parent of every member comes from the next highest level.

A balanced hierarchy can be used to represent time where the meaning and depth of each level, such as Year, Quarter, and Month, is consistent. They are consistent because each level represents the same type of information, and each level is logically equivalent. The following diagram shows an example of a balanced time hierarchy.



Figure 3. Example of a balanced hierarchy

Unbalanced hierarchies

Unbalanced hierarchies include levels that are logically equivalent, but each branch of the hierarchy can descend to a different level. In other words, an unbalanced hierarchy contains leaf members at more that one level. The parent of every member comes from the level immediately above.

An example of an unbalanced hierarchy is the following organization chart, which shows reporting relationships between employees in an organization. The levels within the organizational structure are unbalanced, with some branches in the hierarchy having more levels than others.



Figure 4. Example of an unbalanced hierarchy

IBM Cognos Dynamic Cubes inserts padding members to balance such hierarchies. For more information, see "Padding members" on page 14.

Ragged hierarchies

In a ragged hierarchy, the parent of at least one member does not come from the level immediately above, but a level higher up.

The following diagram shows a Geography hierarchy with Continent, Region, State, and City levels defined. One branch has North America as the continent, Canada as the region, Manitoba as the state, and Winnipeg as the city. Another branch has Europe as the continent, Greece as the region, and Athens as the city, but has no entry for the state level because this level is not applicable. The parent of Athens is at the region level rather than the state level, creating a ragged hierarchy.



Figure 5. Example of a ragged hierarchy

IBM Cognos Dynamic Cubes inserts padding members to balance such hierarchies. For more information, see "Padding members" on page 14.

Padding members

IBM Cognos Dynamic Cubes inserts padding members to balance unbalanced and ragged hierarchies. Padding members do not represent actual dimension members; they are visible only for navigational and performance reasons.

You can reference a padding member in an expression in the same way as any other hierarchy member.

Padding members can include a blank caption or the same caption as the parent. The following diagram illustrates a ragged hierarchy with a padding member included in the Europe branch. A blank caption was used as the caption for the padding member.



Figure 6. Example of a ragged hierarchy with blank padding member

In the IBM Cognos studios, the metadata for this hierarchy with blank captions would show a level without a caption, as in the following example:

North America |--Canada |--Manitoba |--Winnipeg Europe |--Greece |--|--Athens

Figure 7. Example metadata showing blank padding member

The metadata for the same hierarchy using parent captions would show a level that uses the same caption as the parent, as in the following example:

North America |--Canada |--Manitoba |--Winnipeg Europe |--Greece |--Greece |--Athens

Figure 8. Example metadata showing parent padding member

A dimensional member can have only one child padding member.

The use of padding members can result in skewed calculations related to the members of a hierarchy level. For information on removing skewed data from reports, see "Removal of padding members from reports" on page 118.

Extraneous padding members

An extraneous padding member is used to provide a value for a non-leaf hierarchy member. This type of padding member is not required to balance a hierarchy.

To provide easier navigation of a hierarchy with extraneous padding members, you can remove the path of these members.

If padding members do not occur in a hierarchy, and no paths are removed, the hierarchy remains a rollup hierarchy; in other words, a member's value can be derived from aggregating the values of its immediate children. If a regular hierarchy has extraneous paths removed, the entire dimension is identified as non-rollup. This disables certain query optimizations in the dynamic query mode query planner and causes certain aggregate values to be returned as Null or n/a. This typically occurs when a detail filter is applied to a query at a level below the lowest projected level in a report, or if the context filter (slicer) in a report contains multiple members from a single hierarchy.

In a hierarchy containing a non-leaf member with padding members, the padding members are assigned a value of Null for all measures.

To remove extraneous members, you must set the **Show Extraneous Padding Members** property in a hierarchy. For more information about setting this property, see "Model hierarchies" on page 39.

Parent-child hierarchies

A parent-child hierarchy contains relational dimension tables based on a recursive relationship for which there are no predefined levels. For example, an Employee parent-child hierarchy may specify Supervisor as the parent member, and Employee as the child member. The relationships within the data determine what is visible to report users in the IBM Cognos studios, and you can drill down from member to member according to the defined relationships.

Cognos Dynamic Cubes supports parent-child hierarchies.

Data members

A fact table can contain data for non-leaf members as well as leaf members. Data for non-leaf members is derived from aggregating data from leaf members. By

default, non-leaf data members are hidden in a parent-child hierarchy, but you can choose to make them visible. The value for a non-leaf member is the aggregation of all its descendants and its own value.

In the following example, a fact table contains data values for two cities: San Jose (15), and Oakland (20), and one state: California (100). The city values roll up to the state.

State/City	Sales
San Jose	15
Oakland	20
California	135

Table 2. Example fact table with non-leaf data members hidden

In the corresponding hierarchy structure, California (the parent) is a non-leaf member to which a separate data value (100) is assigned. The total data value for California is 135 because the hidden data value for California is included.

The next table illustrates the same fact table, but with non-leaf data members shown. Here you can see that there is an additional child value for California (100). The total rolled up sales value for California remains 135. The data member highlights the contribution of a member to the aggregated value.

Table 3. Example fact table with non-leaf data member shown

State/City	Sales
San Jose	15
Oakland	20
California	100
California	135

If a parent-child hierarchy has non-leaf data members, the entire dimension is identified as non-rollup. This prevents the query engine from assuming the value of a parent is the rollup of its children.

When modeling a dynamic cube, it is important to consider the presentation of a hierarchy against the impact it can have on reports/analyses posed against the hierarchy (and its encompassing dimension and related hierarchies).

To show or hide non-leaf data members, you must set the **Show Data Members** property in a parent-child hierarchy. For more information about setting this property, see "Model parent-child hierarchies" on page 44.

Levels

A level is a collection of attributes related to one aspect of a hierarchy. For example, a Region hierarchy can contain States and City levels.

For more information about attributes, see "Attributes" on page 18.

You can define an All level at the highest level of a hierarchy. An All level contains a single member that aggregates data from all members in the child levels of the

hierarchy. For example, you can include an All level in a Region hierarchy that aggregates data for all cities, in all states, in all regions.

Important: There are many ways to model a hierarchy using levels. Whether you follow best practice or different modeling techniques, it is important that you define each level so that the level key attributes uniquely identify the values in that level.

Best practice modeling

Both star and snowflake schemas can be used to implement best practice modeling. For example, in a star schema the relational data for each dimension is stored in a single dimension table that contains ID columns for each of the levels in the dimension, and each ID column uniquely identifies the values in the level. You might have a single dimension table for the Region dimension that contains the following columns:

Table 4. Example of a single dimension table using best practice modeling

Columns in a best practice Region dimension table						
City ID (Primary key)						
City name						
City mayor						
State ID						
State name						
State governor						
Region ID						
Region name						

Alternative modeling

If you do not have unique ID data columns for each level in your hierarchy, you must be careful when you define the level key attributes for each level. For example, you might have a single dimension table for the Region dimension that contains the following columns:

Table 5.	Example	of a	single	dimension	table	using	alternative	modeling
			<u> </u>					

Columns in an alternative Region dimension table
City ID (Primary key)
City name
City mayor
State name
State governor
Region name

You can create a hierarchy that contains Region, State, and City levels, like in the best practice modeling example. However, you must carefully define the level key attributes to ensure that each row in the level can be uniquely defined. For example, City name does not uniquely define the City level because there are cities with the same name in the United States and in England. The only way to uniquely define the City level is with the combination of the Region name, State name, and City name attributes, as shown in the following table.

Table 6. Example of unique level key attributes using multiple columns

Level	Level key attributes	Level related attributes
Region	Region name	
State	Region name, State name	State governor
City	Region name, State name, City name	City major

Joins

A join combines columns from two relational tables using an operator to compare the columns. A join uses attributes that reference columns in the tables being joined.

The simplest form of a join uses two attributes: one that maps to a column in the first table and one that maps to a column in the second table. You also specify an operator to indicate how the columns are compared. For example, "Time ID = time_id".

A join can also model composite joins where two or more columns from the first table are joined to the same number of columns in the second table. A composite join uses pairs of attributes to map corresponding columns together. Each pair of attributes has an operator that indicates how that pair of columns is compared. For example, "Customer Number = customer_number AND Store Number = store_number".

A join also has a type and cardinality. The join types map to relational join types. Joins are primarily used to join the cube dimensions to the relational tables. Joins can also be used to join dimension tables together in snowflake schema.

The most common type of join is the one-to-many equality join.

Attributes

An attribute is an item used to describe part of a level. For example, a Product level can have a Color attribute. An attribute contains an expression that can be a simple mapping to a data source column or a more complex expression. Complex expressions can combine multiple columns or attributes. They can use functions that are supported against a relational data source, including user-defined functions, if necessary.

When modeling levels in IBM Cognos Cube Designer, there are some special attributes you can define:

- **Member caption** does not appear as a separate attribute of a level; it is used only as the caption for hierarchy members.
- **Member description** appears as a separate attribute with the name *level name* description.
- Level unique key appears as a separate attribute with the name level name key.

When additional attributes are used in an expression, they cannot form attribute reference loops. For example, if Attribute A references Attribute B, then Attribute B cannot reference Attribute A.

Attribute names must be unique from the names of all other attributes in a dimension.

Dynamic cubes

A dynamic cube represents a dimensional view of a star or snowflake schema. It is based on a single fact table and defines the relationships between dimensions and measures.

To model a basic dynamic cube, you must ensure that it contains the following items:

- A measure dimension that contains at least one measure
- At least one dimension
- At least one hierarchy and associated levels defined for each dimension
- Mappings between the measures and dimensions
- Attributes that reference table columns either directly, by expressions, or by an expression that is a constant value

Measures are used to aggregate data from a fact table using specified dimensions. They describe data calculations using columns in a relational table. The following diagram shows how measures relate to relational data.



Figure 9. The relationship between measures and relational data

Dimensions are connected to a measure using joins. A hierarchy provides a way to calculate and navigate a dimension. It stores information about how the levels within a dimension are related to each other, and how they are structured. Each dimension has one or more hierarchies that contain levels with sets of related attributes. The following diagram shows how dimensions are built from relational tables.



Figure 10. The relationship between dimensions in a project and the source relational tables

In a star schema, joins are used to connect tables to create a dimension or a measure. Joins can also connect a measure dimension to specific dimensions. The dimensions reference their corresponding hierarchies, levels, attributes, and related joins. A measure dimension references its measures, attributes, and related joins. In a snowflake schema, joins can also connect tables between dimensions. The following diagram shows how the items fit together in a dynamic cube and map to a relational snowflake schema.



Figure 11. The mapping of items in a dynamic cube to the relational snowflake schema

Measures

In IBM Cognos Dynamic Cubes, you can define regular measures and calculated measures.

Regular measures are mapped directly to a database column of numerical data or defined by an expression. If defined by an expression, the expression is constructed from relational metadata and cannot include dimensional constructs and functions.

Calculated measures are computed in the context of a dynamic cube and computed in the dynamic query server. The expression is constructed from cube metadata and uses dimensional constructs and functions. Dimensional expressions are required when it is necessary to traverse hierarchical relationships or compute complex calculations which are difficult or impossible with relational expressions. With dimensional expressions, you have the ability to access parent/child relationships, to calculate parallel periods, to use set operations and to define an expressions which are evaluated based on its context within a query. There are some behavior similarities between calculated measures and calculated members. For information about calculated members, see "Calculated members" on page 57.

In Cognos Dynamic Cubes, a measure dimension, containing a set of measures, is used in a dynamic cube as the center of a star schema. The physical grouping of measures into a single fact table implies that they share one area of interest. Each measure references the attributes that are used in measure-to-dimension joins. Each measure also references the attributes and joins that are used to map the additional measures across multiple database tables. The value of a measure is meaningful only within the context of the dimensions in a cube. For example, a revenue of 300 has no meaning on its own, but does have meaning in the context of dimensions, such as Region and Time. For example, the revenue for New York in January is 300. Common examples of measures are Revenue, Cost, and Profit.

Simple arithmetic expressions can often be evaluated either by the relational database or in the context of the cube. If a measure expression can be evaluated in either context, it may be preferable to choose a relational expression. Relational databases usually have access to wider range of functions and may be more efficient. If a database is constrained in terms of resources, an alternative is to use calculated measures.

Aggregation rules

Each measure has a regular aggregation type. Aggregation rules can be used in addition to the regular aggregate. They specify how semi-aggregate measures are aggregated with respect to information from the dimension. A semi-aggregate measure has a different aggregation rule defined for at least one dimension. For example, if you want to determine the average inventory for products in a quarter, you would roll up dimensions such as Products as Sum, then aggregate the Time dimension as Average.

Aggregation rules can be aggregates such as Sum, Maximum, and Minimum. You can also specify a **Calculated** or **Custom** aggregation rule.

When you use a **Calculated** aggregation rule, IBM Cognos Dynamic Cubes first aggregates each measure in the expression using its aggregation rule. Then, it uses the values of the aggregated measures to calculate the expression.

Use Sum and Count aggregates rather than Average where possible. You can also use simple calculations by selecting a measure and assigning a rule, such as Average.

Location	Time	Sales	Average Returns
USA	Q1	10	2
USA	Q2	30	4
USA	Q3	50	6

Table 7. Sample data for calculated aggregation rule example

Sales is defined with an aggregation rule of Sum. Average Returns is defined with an aggregation rule of Average.

In this example, the calculated measure, Measure A, is defined by the expression (Sales - Average Returns).

If Measure A is assigned an aggregation rule of Sum, its value is computed as follows if grouping by distinct values of Location.

10 - 2 = 8 30 - 4 = 26 50 - 6 = 44Measure A 8+ 26 + 44 = 78

If Measure A is assigned an aggregation rule of Calculated, its value is computed as follows if grouping by distinct values of Location.

Sales 10 + 30 + 50 = 90 Average Returns (2 + 4 + 6) / 3) = 4 Measure A 90 - 4 = 86

The **Custom** aggregation rule indicates that the value of the measure is computed by an external business process. Custom aggregate rules are a specialized form of non-distributive measures that do not roll up. Values must exist in the measure or aggregate tables at the precise level of aggregation required for a query, otherwise the values are shown as Null. You can customize measure values using advanced business logic and make those values available in IBM Cognos Business Intelligence.

For example, a physician writes a total of 100 prescriptions per year. Because the physician has offices in Florida and New York, rolling up the two territories to a higher level would result in double counting (200 prescriptions), which is incorrect. Custom business rules are applied to prevent double counting in reports. The results of these business rules would be included in a custom aggregate table.

Non-distributive measures

Non-distributive measures must always be aggregated from the detail fact table and cannot be aggregated from one level to the next.

A non-distributive measure is a measure defined with a non-distributive aggregation rule such as:

- Count Distinct
- Average
- Standard Deviation
- Variance
- Median

Aggregate tables can be used only if they are calculated from the exact group of levels of the SQL query. If none of the aggregate tables exactly match the required roll ups, the aggregate value must be computed from the fact table. As a result, higher level aggregations of non-distributive measures against a large fact table can take longer to calculate than measures which can take advantage of external aggregate tables

A dynamic cube stores the values of non-distributive measures in its data cache for later use.

When calculating summary values in a query, non-distributive measures require a separate SQL query for each summary. These summary values are query-specific and are not stored in the data cache.

For a cross tab report with row/column summaries, each summary requires a separate SQL query which, depending upon the underlying database, can have an impact on query performance.

Unlike non-distributive measures, distributive measures can always be aggregated from one level to the next. For example, the sum of Sales for a quarter can be calculated by summing monthly sales data.

Virtual cubes

In IBM Cognos Dynamic Cubes, a virtual cube consists of two merged cubes. You can merge cubes by using the following combinations:

- Merge two source cubes.
- Merge two virtual cubes.
- Merge one source cube with one virtual cube.

By combining two virtual cubes, or one source cube with a virtual cube, you can merge more than two cubes into a single virtual cube.

Some advantages of using virtual cubes include the following points:

- Virtual cubes use less memory than physical cubes.
- There is reduced cube refresh latency.
- You can add volatile information to a lookup cube.
- You can join cubes to present consolidated data and provide more sophisticated calculations.
- Each source cube can be derived from a separate data source.

A virtual cube must contain the following objects:

- A virtual measure dimension that contains one or more virtual measures.
- At least one virtual dimension that contains one or more virtual hierarchies.

It can also contain virtual calculated measures, and virtual calculated members.

When you create a virtual cube, the following objects are added, if they exist in at least one source cube:

- Dimensions
- Hierarchies
- Measures
- Levels
- Members

Virtual dimensions and hierarchies

Any dimensions and hierarchies with identical names in the source cubes are known as conformed dimensions and conformed hierarchies. These objects are added to the virtual cube as merged virtual dimensions and virtual hierarchies.

For example, two source cubes with a Time dimension are merged into a virtual dimension also named Time.

Any dimensions and hierarchies that do not have identical names, or that exist in only one of the source cubes, are known as non-conformed dimensions and non-conformed hierarchies. These objects are added to the virtual cube as new virtual dimensions and virtual hierarchies.

For example, if source cube 1 contains a Sales Q3 hierarchy, and source cube 2 contains a Sales Q4 hierarchy, the dimensions are not merged because the names do not match. Instead, two virtual hierarchies, Sales Q3 and Sales Q4, are added to the virtual cube.

If a virtual cube contains a non-conformed hierarchy, the virtual cube queries both source cubes to retrieve data only if one of the following conditions is met:

- The non-conformed hierarchy is deleted from the virtual cube.
- The virtual hierarchy includes an All member and the query includes this member.

This can occur if the All member is referenced explicitly in he query or if the All member is the default member.

If neither of these conditions is met, the virtual cube queries only the source cube with the non-conformed hierarchy, and never the second source cube.

Virtual measures

Any measures with identical names in the source cubes are added to the virtual cube as merged virtual measures. Any measures that do not have identical names, or that exist in only one of the source cubes, are added to the virtual cube as new virtual measures.

Important: It is possible to merge measures only when the regular aggregate is one of the following: Sum, Maximum, Minimum, or Count. It is not possible to merge non-distributive measures or a distributive measure with an aggregation rule applied.

When merging measures from two source cubes, if the data format both measures do not match, the data format of the merged virtual measure is set to * or unknown. For example, if a measure in source cube 1 has a US currency data format, and a measure in source cube 2 has a UK currency data format, the data format cannot be merged.

Virtual levels

Source cubes containing identical levels in a hierarchy (same number of levels and identical names), are merged as virtual levels. If levels in the source cubes are not identical, level names from the first source cube are used as the names of the virtual levels. If one source cube contains more hierarchy levels than the second source cube, the extra levels are added as the lowest levels of the virtual hierarchy.

For example, source cube 1 contains a Time hierarchy with Year, Quarter, and Month levels. Source cube 2 also has a Time hierarchy with Year, Month, Day, and Time levels. When they are merged, a Time virtual hierarchy is created with Year, Quarter, and Month, and Time virtual levels.

Virtual members

For a virtual hierarchy that is merged from two conformed dimensions, all hierarchy members from the source cubes are available as virtual members. If the level key for each source member is identical, members are added to the virtual cube as merged virtual members. Any members that do not have matching level keys are added to the virtual cube as new virtual members.

Tip: To browse virtual members, ensure that each source cube is deployed as data source to the content store and started.

Calculated measures and calculated members

Calculated measures and calculated members from source cubes are not added to a virtual cube. To use calculated measures or members from source cubes, you must manually define them in the virtual cube.

For more information, see "Calculated members" on page 57.

Aggregate cubes

Aggregate cubes are unavailable in a virtual cube because a virtual cube can retrieve data only from source cubes, not by querying a data source.

Support for multiple locales

If source cubes include support for multiple locales, a virtual cube also has multiple locale support.

A virtual cube automatically supports all locales defined in the source cubes. For example, in source cube 1, English and French are defined as supported locales. In source cube 2, English and Japanese are defined as supported locales. In the virtual cube, English, French and Japanese are included as supported locales.

A virtual cube also supports the use of multilingual names and captions for a virtual cube, virtual dimensions, virtual hierarchies, virtual levels, and virtual measures. However, with the exception of the All member caption, multilingual names and captions from source cubes are not automatically added to a virtual cube. To use multilingual names and captions from source cubes, you must manually define them in the virtual cube.

Manual merging of source objects

It is possible to manually merge objects in a virtual cube that could not be merged automatically. For example, source cube 1 contains a Time dimension and source cube 2 contains a Fiscal Time dimension. They are not merged, so two virtual dimensions Time and Fiscal Time are added to the virtual cube. If both dimensions contain the same structure and data, you could manually merge them into one virtual dimension named Time. You could then delete the redundant Fiscal Time virtual dimension.

You cannot reference a source object more than once in a virtual cube. For example, if the Time source hierarchy is used in the Time virtual hierarchy, it cannot also be used in the Fiscal Time virtual dimension.

Virtual cube scenarios

Common scenarios for using virtual cubes are described here. You can combine these scenarios based on your specific needs.

Cubes with partitioned data

Sales information for a large region is stored in two cubes. Fact data for each cube can originate from a single fact table or two separate fact tables. One cube, WestSales, stores sales information for the west region, and the other cube, EastSales, stores sales information for the east region. WestSales and EastSales have the same structure. To provide a combined view of the sales data, you can define a virtual cube AllSales to merge the two regional cubes.

Cubes with pre-cached historical data and current data

Sales information is stored in a single cube called AllSales. The cache of this large cube must be rebuilt frequently to reflect the updates in the database. The rebuilding process usually takes a long time.

To address this problem, you can split AllSales into two cubes: one to record the historical sales information (HistoricSales), and another to record the daily sales information for the current month (CurrentMonthSales). You can then define a virtual cube called VirtualSales to join these two cubes. By reorganizing the cubes this way, performance is improved in the following ways:

- Because you refresh data only for CurrentMonthSales, cube refreshing performance is improved.
- Because query results from HistoricSales are pre-cached, and CurrentMonthSales is small in size, performance for queries run against the sales data of the entire time period is improved.
- Because of the smaller size of CurrentMonthSales, performance for queries run against the sales data of the current month is improved.

Cubes with shared dimensions

Sales information is stored in a single cube called GlobalSales. You need to convert some sales figures into other currencies. You could add exchange rates to this cube, but the cube might contain redundant data and would be hard to maintain.

Instead, you can create a cube called ExchangeCurrency to store the exchange rates, and define a virtual cube SalesConversion to perform currency conversion for the sales data. GlobalSales and ExchangeCurrency share some dimensions but do not have the same structure.

Aggregate cubes

In IBM Cognos Cube Designer, you can model aggregate cubes within a dynamic cube when the imported data source for a dynamic cube contains fact tables with pre-aggregated data.

IBM Cognos Dynamic Cubes supports the use of aggregate cubes created in a dynamic cube and rewrites queries to use the underlying aggregate tables whenever possible. For information about aggregate cube modeling, see Chapter 8, "Aggregate cube modeling," on page 71.

Aggregate tables

Although it is a best practice to store the lowest level of data in a detail fact table in a data warehouse, selected data can be summarized in a separate table known as an aggregate table. An aggregate table contains detail fact data that is aggregated at a higher level relative to one or more of the dimensions associated with the data.

Using aggregates is critical to achieving performance over large scales for the following reasons:

- It allows you to use pre-calculated data from a data warehouse.
- It decreases the amount of data required to be accessed from the data warehouse.

Some database vendors use special table types for aggregate tables. For example, IBM DB2[®] uses Materialized Query Tables (MQTs) and Oracle uses Materialized Views. The relational database understands that these special tables are aggregates and routes to them for performance if the database can determine they are applicable and faster. The aggregate awareness feature in Cognos Dynamic Cubes can also use these tables so that a dynamic cube routes to these aggregates tables rather than relying on the database to do the routing.

To increase performance, more than one aggregate table can be necessary in a schema. However, if an aggregate table summarizes data at too high a level within one or more hierarchies, the aggregates can be applicable to a only a few queries. In addition, if many dimensions are used, it can be difficult to design frequently used aggregate tables.

When creating aggregate tables, refer to the documentation for your database for information about creating a data warehouse, in particular indexing your data, and co-locating fact and dimension tables. Cognos Dynamic Cubes supports these concepts:

- Sharing common dimension tables if fact and aggregate tables are co-located in the same storage space.
- Use of separate dimension tables for aggregate tables (co-locating dimension and fact data).
- Inclusion of dimension level keys entirely within an aggregate table to avoid joins to dimension tables.
- Partitioning of data.

In-database aggregates

In-database aggregates are aggregate tables that a database administrator can create and apply to the database. After the database has been updated, a modeler must model an aggregate cube for each created aggregate table in the database and redeploy the dynamic cube to the content store.

In-memory aggregates

In-memory aggregates are aggregate tables that can be applied by the IBM Cognos Business Intelligence server the next time the cube is started. These aggregates are stored in the content store.
Aggregate Advisor

Aggregate Advisor is an external tool, available with IBM Cognos Dynamic Query Analyzer, that can analyze the underlying model in a dynamic cube data source and recommend which aggregates to create. These aggregates can be created both in-database and in-memory.

Aggregate Advisor can also reference a workload log file that allows it to suggest aggregate tables (in-database or in-memory) that correspond directly to the reports contained in the log file.

For more information on using Aggregate Advisor, see the *IBM Cognos Dynamic Query Analyzer User Guide*.

Chapter 4. Getting started with Cognos Cube Designer

IBM Cognos Cube Designer is the modeling tool provided with IBM Cognos Dynamic Cubes. You use it to build dynamic cubes and publish them for use in the IBM Cognos studios.

To get started, you import metadata from a relational database. Using the metadata, you model dynamic cubes and save the cube definitions in a project. After you publish the cubes, they are listed as data sources in Content Manager and their related packages are available to report authors.

Introduction to Cognos Cube Designer

IBM Cognos Cube Designer is the application that you use to model dimensional metadata and dynamic cubes. The **Data Source Explorer** tree, the **Project Explorer** tree, the object editors, and the **Properties** pane are the main parts of the Cognos Cube Designer user interface.

Getting Started page

The **Getting Started** page is shown when you start Cognos Cube Designer. You can also display this page at any time by clicking **Show the Getting Started Page** from the **Help** menu.

You can perform the following tasks:

- Click **Create New from Metadata** to import metadata into a new project. For more information, see "Import metadata" on page 33.
- Click **Create New Blank Project** to create a project.

For more information, see "Managing a project" on page 34.

Click **Open Existing** to open a project.
 For more information, see "Managing a project" on page 34.

Data Source Explorer

The **Data Source Explorer** shows the metadata imported from relational data sources. You can view the columns, keys, and joins by expanding a table in the **Data Source Explorer** tree.

You can perform the following tasks:

- Right-click a table and select **Explore Metadata** to view a graphical representation of the metadata in the **Relational Explorer Diagram** tab. You can view the columns in a table, the primary key and foreign keys, and its joins to other tables.
- Right-click a table and select **View Data** to view sample data from the data source in the **Tabular Data** tab.

Data is retrieved from the data source and shown in IBM Cognos Viewer.

• Right-click a fact table and select Generate, Cube to create a dynamic cube.

Use this option to create a dynamic cube based on a fact table in the data source. The cube, including all required dimensional metadata, is added to the project in the **Project Explorer**. For more information about creating cubes, see "Model a dynamic cube" on page 49.

Relational Explorer

The **Relational Explorer Diagram** shows a graphical view of your data source metadata. Use the **Relational Explorer Diagram** to explore your metadata and view the relationships between objects.

Tip: When this tab is visible, you can drag tables from the **Data Source Explorer** tree to explore them.

Project Explorer

The **Project Explorer** shows all the dimensional metadata definitions and dynamic cube definitions included in a project. Use the **Project Explorer** tree to add objects to your dynamic cubes, access the object editors, and publish your cubes.

You can perform the following tasks:

Model dimensions and hierarchies.

For more information, see Chapter 5, "Dimensional metadata modeling," on page 37.

Model dynamic cubes

For more information, see "Model a dynamic cube" on page 49.

• Right-click and select **Validate** to validate an entire project or an individual object.

For more information about validation, see "Validate a project and individual objects" on page 35.

• Right-click a cube and select **Publish** to deploy the cube and, optionally, publish a package to be used by report authors.

For more information about publishing, see "Deploying and publishing dynamic cubes" on page 54.

Tip: When you add a dynamic cube to a project, the data source on which it is based is added to the **Data Sources** folder in the **Project Explorer** tree. You can view the database catalog and schema referenced by the data source in the **Properties** tab.

Functions tab

From the **Functions** tab (1), you have access to the operators, summaries, constants and functions that you use in expressions.

Object editors

There is an editor available for each object. When an editor tab is visible, you can also access other functionality related to the object. For example, when viewing the cube editor, you have access to the **Aggregates**, **Security**, and **Implementation** tabs.

To access an editor and its related tabs, right-click the object in the **Project Explorer** tree and select **Open Editor**.

Tip: To keep multiple editor tabs accessible, right-click the tab and select **Pin**. Because some of the editor windows are similar in appearance, verify your edit location on the tab.

Implementation tab

The **Implementation** tab shows a physical diagram of the current object. For example, to view the implementation of an entire cube, right-click the cube in the **Project Explorer** tree tab, select **Open Editor**, and then select the **Implementation** tab. For some objects, you can also add or edit relationships between the cube objects. Select an object and click to use menus to explore the diagram.

Object properties

On the **Properties** tab, you can view and edit the properties of an object.

To access the properties of an object, select the object in the **Project Explorer** tree. For more information about object properties, see Chapter 5, "Dimensional metadata modeling," on page 37 and "Model a dynamic cube" on page 49.

Validation issues

The **Issues** tab shows errors and warnings for objects that must be fixed to validate them.

You can view validation issues for all objects in a project or for an individual object. Select the project or object in the **Project Explorer** tree and then click the **Issues** tab. For more information about validating objects, see "Validate a project and individual objects" on page 35.

Import metadata

You import metadata to use as the basis for modeling dimensional metadata and dynamic cubes.

Remember: You must ensure that the data source from which you import metadata supports the dynamic query mode.

You can import metadata from a Content Manager data source. This allows you to model dynamic cubes based on a relational data source that is defined in IBM Cognos Business Intelligence.

Tip: If you want to browse hierarchy members while modeling dynamic cubes, before importing the metadata, check whether your administrator created a development or test data source connection that contains a subset of the metadata. Using a smaller volume of metadata can speed up the modeling process.

Importing metadata from a Content Manager data source

If you want to model dimensional metadata and dynamic cubes based on a relational database, you import the metadata from a Content Manager data source.

You import metadata from one data source at a time. If you want to import from more than one data source, you must perform separate imports.

A separate file is created for each data source from which you import metadata. These files are stored in the *c10_location*\data directory.

A dynamic cube is modeled using a single data source only. A project can contain many dynamic cubes, and if you have imported multiple data sources, each dynamic cube can be derived from a separate data source.

Before you begin

Before importing metadata from a Content Manager data source, check the following prerequisites:

- The data source contains a star or snowflake schema.
- The data source connection to the database uses a JDBC driver. This is required by dynamic query mode.
- The data source is defined in IBM Cognos Business Intelligence. If it does not exist, you must first create it.

For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

You can create a connection to the following databases using a JDBC driver: IBM DB2, Microsoft SQL Server, Netezza[®], Oracle, and Teradata.

Procedure

1. From the Start menu, click Programs, IBM Cognos 10, IBM Cognos Cube Designer.

You can also start the Cognos Cube Designer from IBM Cognos Framework Manager. From the **Tools** menu, select **Run Cube Designer**.

- 2. From the toolbar, click **Get Metadata**.
- 3. Click Browse Content Manager Datasource.
- Select the database schema from which to import data, and then click OK. The imported metadata is shown as a list of database tables in the Data Source Explorer tree.

Tip: If your project contains more than one imported data source, each data source is shown in a separate panel.

You can now model dimensional metadata and dynamic cubes.

5. When you finish working, click Save 🔚 .

Managing a project

Dynamic cube definitions are saved in a project. This section describes how to open, edit, and save an existing project.

Tip: It is good practice to save a project at regular intervals.

Procedure

- 1. From the toolbar, click **Open** 🛅 .
- 2. Select the project file (.fmd).

- 3. Click OK.
- 4. Edit individual objects as required.

For more information, see Chapter 5, "Dimensional metadata modeling," on page 37 and "Model a dynamic cube" on page 49.

5. When you finish, click Save 🔚 .

Validate a project and individual objects

IBM Cognos Cube Designer automatically validates individual objects as you design them. Modeling issues are identified in the **Project Explorer** with icons shown next to objects that are causing the issues:

- Errors are indicated by a red cross.
- Warnings are indicated by a yellow triangle.

The **Issues** tab shows a list of all the issues related to a selected object. You can click an issue for more details. If a solution is provided, you can resolve the issue by selecting the solution and clicking **OK**. You can also click **Invoke Editor** to access the object editor.

You can validate an entire project or an individual object at any time. Validate frequently and resolve issues as they are reported. If you attempt to model a large cube without validating as you go, you may have a long list of issues to resolve.

You can validate each object as you create it by right-clicking it in the **Project Explorer** and selecting **Validate**.

You cannot deploy a dynamic cube that contains errors. It is possible to deploy a valid cube when the project contains unrelated objects that are not valid.

Chapter 5. Dimensional metadata modeling

You use IBM Cognos Cube Designer to model dimensions, hierarchies and levels.

Model dimensions

With IBM Cognos Cube Designer, you can model commonly used dimensions at the project level and reference them in one or more dynamic cubes. You can also model dimensions within a specific cube.

The following table lists the properties that you can set when modeling a dimension.

Property	Description	
Name	The dimension name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.	
Comment	A comment or description of the dimension. Comments are not visible in the IBM Cognos studios.	
Default Hierarchy	The hierarchy to use when no hierarchy been specified for a dimension used in an expression.	
	Applies only when multiple hierarchies are defined for a dimension.	
Multilingual Support	Disabled (default) - Specifies that members do not have multiple locale support.	
	By Column - Specifies that members support multiple locals.	
	For more information about multiple locales, see "Multiple locales" on page 68	
Dimension Type	Regular (default) - Identifies a regular dimension.	
	Time - Identifies a time dimension. For more information about relative time dimensions, see "Defining a relative time dimension" on page 63.	

Table 8. Properties of a dimension

Defining a dimension

In IBM Cognos Cube Designer, you can model commonly used dimensions at the project level and reference them in one or more dynamic cubes. You can also model dimensions within a specific cube.

When you add a dimension, it contains an initial set of objects that you need to complete the dimension. When you validate the dimension, you can use information from the **Issues** tab to help you complete the dimension definition.

Procedure

1. Select the location from which you want create the dimension:

- To create a shared dimension at the project level, select **Model** from the **Project Explorer** tree.
- To create a dimension within a dynamic cube, select the cube from the **Project Explorer** tree.

Tip: Use folders and namespaces to organize objects. Using folders and namespaces makes it easier for you to locate objects and view the structure of a project in the **Project Explorer**.

- 2. Click **New Dimension !**. The dimension contains a set of initial objects you can use to complete the dimension.
- 3. To create additional hierarchies, click New Hierarchy 🔛 .
- 4. To create additional levels, click New Level 🔛 .
- 5. On the **Properties** pane, set the default hierarchy.
- 6. To access the dimension editor, right-click a dimension from the **Project Explorer** tree and select **Open Editor**.
- 7. Change the order of the levels by clicking **Move Up 1** and **Move Down U**.

What to do next

To complete the dimension, you must complete the definition of each hierarchy and level that belongs to the dimension. For more information, see "Defining a hierarchy" on page 40 and "Defining a level" on page 42.

Tip: Right-click a relational table and select **Explore Metadata**. You can use the **Relational Explorer Diagram** to help you understand the structure of the metadata used to design the hierarchies and levels.

When you finish modeling a dimension, you can perform the following tasks:

- Browse members from the data source. For more information, see "Browsing members" on page 47.
- Add a shared dimension to a dynamic cube by dragging and dropping it onto the dynamic cube in the **Project Explorer** tree.

Related tasks:

"Defining a hierarchy" on page 40

In IBM Cognos Cube Designer, a single level-based hierarchy is automatically added when you create a dimension. You can also create multiple level-based hierarchies in a dimension.

"Defining a level" on page 42

In IBM Cognos Cube Designer, you define levels to model the relationships in a hierarchy.

"Defining a parent-child hierarchy" on page 46

In IBM Cognos Cube Designer, you can model commonly used parent-child hierarchies at the project level and reference them in one or more dynamic cubes. You can also model parent-child hierarchies within a specific dynamic cube.

Model hierarchies

IBM Cognos Dynamic Cubes supports level-based hierarchies and parent-child hierarchies. A single level-based hierarchy is automatically added when you create a dimension. You can also create multiple level-based hierarchies in a dimension.

For more information, see "Dimensions" on page 11 and "Hierarchies" on page 11.

Complete the hierarchy definition using the properties listed in the following table:

Property	Description	
Name	The hierarchy name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.	
Comment	A comment or description of the hierarchy. Comments are not visible in the IBM Cognos studios.	
Multiple root members	False (default) - the hierarchy uses a single root member at the top of the hierarchy. Selecting this option creates the All level at the top of the hierarchy. You can change the default caption of the top level by editing the Root Caption property.	
	True - the hierarchy contains multiple root members. Selecting this option deletes the All level that is automatically created at the top of the hierarchy.	
	If a hierarchy is single root, Cognos Cube Designer generates the root member. Because all members must belong to a level, the root member is in the All level.	
Add Relative Time Members	False (default) - If the hierarchy belongs to a Time dimension, relative time members are not added to the hierarchy.	
	True - If the hierarchy belongs to a Time dimension, relative time members are added to the hierarchy.	
	For more information, see "Defining a relative time dimension" on page 63.	

Table 9. Properties of a hierarchy

Property	Description	
Default Member	The member value to use when evaluating member expressions, where no value is specified for a hierarchy.	
	If the default member is empty, the root member of the hierarchy is used.	
	To set a default member, drag the required member from the Members folder in the Project Explorer tree.	
Root Caption	The caption of the root member at the top of the hierarchy shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the caption in all supported languages.	
Parent-Child	False - Indicates that the hierarchy does not use a parent-child structure.	
	This property cannot be edited.	
Show Extraneous Padding Members	False (default) - collapse multiple paths of padding members under a single member into a single path.	
	True - show multiple paths of padding members for a single member.	
	For more information, see "Padding members" on page 14.	
Caption of Padding Members	The caption to use for padding members in the hierarchy.	
	Empty (default) - use a Null caption.	
	Parent's caption - use the caption of the parent.	
	For more information, see "Padding members" on page 14.	

Table 9. Properties of a hierarchy (continued)

Defining a hierarchy

In IBM Cognos Cube Designer, a single level-based hierarchy is automatically added when you create a dimension. You can also create multiple level-based hierarchies in a dimension.

Procedure

- 1. From the Project Explorer tree, select the dimension you want to work with.
 - To create a new hierarchy, click New Hierarchy
 - To access the hierarchy editor, right-click a hierarchy that belongs to the dimension and select **Open Editor**.
- 2. Complete or modify the hierarchy definition using the **Properties** tab. Identify the **Default Member** and **Root Caption** if required.
- 3. Set the Show Extraneous Padding Members and Caption of Padding Members if required.

For more information, see "Padding members" on page 14.

- 4. If an **All** level is not required, change the **Multiple Root Members** property to **false**.
- **5**. To add levels to the hierarchy, drag levels from the **Levels** folder to the hierarchy.

Model levels

In IBM Cognos Cube Designer, each level in a dimension is defined by creating attributes, mapping those attributes to the relational database source and identifying which attributes are level keys.

When you create a hierarchy, an All level is created at the top of the hierarchy. An All level contains a single member that aggregates data from all members in the lower levels of the hierarchy. For example, an All level in a Region hierarchy aggregates data for all cities, in all states, in all regions.

Complete the level definition using the properties listed in the following table:

Property	Description	
Name	The level name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.	
Comment	A comment or description of the level. Comments are not visible in the IBM Cognos studios.	
Level Type	Identifies whether the level is regular or time-based. Default: Regular	
Current Period	An expression used to define the current period in a time-based level. The value of the expression is compared to the value of the level key attribute at the level.	

Table 10. Properties of a level

Complete the definition of the level attributes using the properties listed in the following table:

Table 11. Properties of an attribute

Property	Description
Name	The attribute name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Comment	A comment or description of the attribute. Comments are not visible in the IBM Cognos studios.
Expression	This property is available only for attributes created in the Cognos Cube Designer.
Column Name	The name of the associated column in the relational database. If the Multilingual property is true, this value can be set. For more information, see "Adding support for multiple locales to members and attributes" on page 68.

Table 11.	Properties	of an	attribute	(continued)
-----------	------------	-------	-----------	-------------

Property	Description
Visible	Controls whether the object is visible in the published package.
	Non-visible objects are typically used to represent intermediate values. These objects are not intended to be used for direct reporting. However a non-visible object is always present in the published package because the object might be needed by other objects in a dynamic cube.
	Non-visible objects are not displayed in the metadata browser and are removed from the output of reports which contain references to them. For example, a report that references a non-visible object does not include output from that measure.
	Default: True
Data Type	The data type of the associated column in the relational database.
	This property cannot be edited.
Precision	The precision of the associated column in the relational database.
	This property cannot be edited.
Scale	The scale of the associated column in the relational database
	This property cannot be edited.
Multilingual	This property appears only if support has been enabled for multiple locales in the dimension. For more information, see "Multiple locales" on page 68.
	False (default) - This attribute does not support multiple locales.
	True - This attribute supports multiple locales.

The **Level Unique Key** consists of one or more attributes whose values uniquely identify each instance of a level. For more information, see "Defining a level unique key" on page 43.

Member Sort consists of one or more attributes that provide information about the ordering of the members within a level. For more information, see "Defining the member sort order" on page 44.

Defining a level

In IBM Cognos Cube Designer, you define levels to model the relationships in a hierarchy.

For each level, you assign or create attributes, map them to the relational data source, identify level keys and, optionally, define a sort order. You can also hide the attribute in the published package if required.

Procedure

- 1. From the **Project Explorer** tree, select a dimension, and click **New level**
- 2. To access the level editor, right-click the level in the **Project Explorer** tree, and select **Open Editor**.
- 3. To create an attribute, click New Attribute 🛄 .

Tip: To give the new attribute a more meaningful name, right-click it and select **Rename**.

4. To map a table column to the new attribute, select the required column from the **Data Source Explorer** tree and drop it onto the **Mapping** column.

Tip: You can also create attributes by dropping table columns to the **Attribute** column.

- Select the attributes assigned to Member Caption and, if required, Member Description. For more information about these special attributes, see "Attributes" on page 18.
- 6. You can define the Level Unique Key one of two ways:
 - If the level unique key is a single attribute, select the **Level Unique Key** check box for the attribute.
 - If the level unique key is a composite key, click **Level Key** . For more information, see "Defining a level unique key."
- 7. If required, specify the member sort order. For more information, see "Defining the member sort order" on page 44.
- 8. To hide an attribute in the published package, change the **Visible** property to false.
- **9**. To assign the level to a hierarchy, select the level and drop it onto the hierarchy in the **Project Explorer** tree.

Tip: You can also assign levels by dropping them into the hierarchy editor.

10. Expand the hierarchy in the **Project Explorer** tree and, if necessary, modify the order of the levels as they appear under the hierarchy.

Defining a level unique key

The **Level Unique Key** consists of one or more attributes whose values uniquely identify each instance of the level.

A level key is meant to uniquely identify each of the members within a level. The first level key shown in the **Level Key** window and denoted with the business key

icon \square . If a level key does not uniquely identify members within a level, then attributes from the current level or parent levels must be used to uniquely identify the members within the level

For example, a City level can use a unique ID as its level key attribute. City names are not unique, so you cannot use the city name attribute as a level unique key. You can include the set of Region name, State name, and City name attributes as a composite level unique key because the three attributes together uniquely define a city.

Level keys in SQL statements retrieve data values from the database and the corresponding columns are used as the basis for grouping, joining, and filtering. For optimal performance, use an attribute with an integer data type as the level key. Avoid character and text fields. There can be a performance difference between an integer level key and any other numeric type depending on the database system in use. For more information, see "Levels" on page 16.

If the level unique key is a single attribute, select the **Level Unique Key** check box for the attribute.

If there are multiple level key attributes, the first attribute must be the level key for the level. You may have to reorder the attributes to ensure the appropriate attribute is defined as the level key.

Procedure

- 1. To define a composite level unique key, right-click a level in the **Project Explorer** tree, and select **Open Editor**.
- 2. Click Level Key 🔛 .
- 3. Select the attributes that together uniquely identify the level.
- 4. Change the order of the attributes by clicking **Move Up 1** and **Move Down**

• The first attribute shown in the **Level Key** window must be the level key for the level.

Defining the member sort order

By default, hierarchy members are shown in the order in which they are loaded into a dynamic cube.

You can select one or more attributes that define the sort order of the members within a level. For example, a Month level might have Month ID as a key attribute, Month Name as the caption attribute, and Month Number as an ordering attribute. Month Number is specified as the ordering attribute because Month Number sorts the months by calendar order whereas Month Name sorts the months alphabetically.

Procedure

- 1. Right-click a level in the Project Explorer tree, and select Open Editor.
- 2. Click Member Sort **2**.
- 3. Select the required attributes from the **Attribute** column and click **Add** to add them to the **Sorting** column.

You can change the sort order by selecting an attribute and clicking **Move Up** and **Move Down** .

- 4. To change the direction of sorting for an attribute, click the **Direction** column and select the required option.
- 5. Click OK.

Model parent-child hierarchies

In IBM Cognos Cube Designer, you model a parent-child hierarchy when the dimension data is based on a recursive relationship and it is not level-based.

For more information, see "Parent-child hierarchies" on page 15.

To model a parent-child hierarchy, you create attributes, map them to the relational data source and identify which attributes represent the parent key and child key. The child key also acts as the member key.

The top level member in a parent-child hierarchy is determined as the member whose parent is Null.

You define a parent-child hierarchy within a parent-child dimension. Be aware of the following constraints:

- A dimension containing a parent-child hierarchy cannot include any other hierarchies.
- The attributes used for the parent key and member key cannot be composite keys.
- A parent-child hierarchy member cannot contain multiple parents. If the imported data source contains hierarchy members with multiple parents, you can use surrogate keys in the data source to overcome this issue.

To access the parent-child dimension properties, double-click a parent-child dimension from the **Project Explorer** tree.

Complete the parent-child dimension definition using the properties listed in the following table:

Property	Description	
Name	The dimension name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.	
Comment	A comment or description of the dimension. Comments are not visible in the IBM Cognos studios.	
Default Hierarchy	The parent-child hierarchy defined within the dimension. This property cannot be edited.	
Multilingual Support	Disabled (default) - Specifies that members do not have multiple locale support. By Column - Specifies that members support multiple locales.	
	For more information about multiple locales, see "Multiple locales" on page 68.	

Table 12. Properties of a parent-child dimension

To access the parent-child hierarchy properties, double-click a parent-child hierarchy from the **Project Explorer** tree.

Complete the parent-child hierarchy definition using the properties listed in the following table:

Table 13. Properties of a parent-child hierarchy

Property	Description
Name	The parent-child hierarchy name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Comment	A comment or description of the parent-child hierarchy. Comments are not visible in the IBM Cognos studios.

Property	Description
Default Member	The member value to use when evaluating member expressions, where no value is specified for a hierarchy.
	If the default member is empty, the root member of the hierarchy is used.
	To set a default member, drag the required member from the Members folder in the Project Explorer tree.
Root Caption	The caption of the root member shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the caption in all supported languages.
Parent-Child	True - Indicates that the hierarchy uses a parent-child structure.
	This property cannot be edited.
Show Data Members	True (default) - show data members for non-leaf members in the hierarchy.
	False - hide data members for non-leaf members in the hierarchy.
	For more information, see "Data members" on page 15.
Caption of Data Members	The caption to use for data members in the hierarchy.
	Empty (default) - use a Null caption.
	Parent's caption - use the caption of the parent.

Table 13. Properties of a parent-child hierarchy (continued)

To access properties of an attribute, select the attribute in the **Attribute** column in the parent-child hierarchy editor. For more information about attribute properties, see "Model levels" on page 41.

Defining a parent-child hierarchy

In IBM Cognos Cube Designer, you can model commonly used parent-child hierarchies at the project level and reference them in one or more dynamic cubes. You can also model parent-child hierarchies within a specific dynamic cube.

Procedure

- 1. Select the location from which you want create the parent-child hierarchy:
 - To create a shared parent-child hierarchy at the project level, select **Model** from the **Project Explorer** tree.
 - To create a parent-child hierarchy within a dynamic cube, select the cube from the **Project Explorer** tree.
- 2. Click New Parent-Child Dimension 🚺.

A new parent-child dimension is created with a parent-child hierarchy.

- 3. Edit the dimension properties in the parent-child Properties pane.
- 4. Open the parent-child hierarchy editor.
- 5. From the **Project Explorer** tree, drag table columns to the **Attribute** column to create the hierarchy attributes.
- 6. Select the attributes assigned to the Parent key and Child key. These attributes are mandatory.

7. Select the attributes assigned to the Member Caption, and Member Description.

The Member Caption attribute is mandatory.

- 8. If required, specify the member sort order. For more information, see "Defining the member sort order" on page 44.
- **9**. Complete the parent-child hierarchy definition using parent-child hierarchy editor **Properties** pane.
- 10. If required, edit the properties of the attributes using the attribute editor **Properties** pane.

Browsing members

When you finish modeling a dimension containing a regular hierarchy or parent-child hierarchy, you can browse the dimension members from the data source.

Tip: A dimension must be valid before you can browse its members. If the dimension that you want to browse is contained in a dynamic cube, the cube must also be valid.

When viewing members in the Cognos Cube Designer, relative time members do not reflect the current period expressions defined in a project, but the members can be used in other expressions if desired. Current period expressions are used when the cube is started.

Procedure

- 1. From the **Project Explorer** tree, select the hierarchy for which you want to browse members.
- 2. Expand the Members folder.

The parent level dimension members are shown.

Note: Depending on the volume of metadata included in the data source, it can be time consuming to browse the full list of members. You can cancel browsing by pressing **Escape**.

3. Expand a member to view its child members.

Repeat this step to view further child members.

- 4. If you make changes to a dimension or hierarchy, you must refresh the list of members to browse.
 - To refresh members for all hierarchies in a dimension, right-click the dimension and select **Refresh Members**.
 - To refresh members in a specific hierarchy, right-click the **Members** folder and select **Refresh**.

Chapter 6. Dynamic cube modeling

With IBM Cognos Dynamic Cubes, you design and prepare dynamic cubes for use as data sources in the IBM Cognos studios.

The process to create dynamic cubes includes the following tasks:

• In IBM Cognos Administration, create a JDBC data source connection to your relational database.

For more information, see the "Create a data source" topic in the *IBM Cognos Administration and Security Guide*.

- In Cognos Cube Designer, import the metadata to use for modeling dynamic cubes.
- In Cognos Cube Designer, model the dimensional metadata.
- In Cognos Cube Designer, model the dynamic cubes.
- In Cognos Cube Designer, deploy individual dynamic cubes as OLAP data sources to Content Manager in IBM Cognos Business Intelligence.
- In Cognos Cube Designer, publish a package that includes a deployed cube. It is also possible to manually publish a package using IBM Cognos Framework Manager. You might publish manually, for example, if you want to create a package containing more than one dynamic cube. For more information about creating and publishing packages, see the *IBM Cognos Framework Manager User Guide*.
- In IBM Cognos Administration, configure the deployed cube for use as a data source by the Query Service.
- In IBM Cognos Administration, start the dynamic cube.

Model a dynamic cube

In IBM Cognos Cube Designer, you can define a dynamic cube manually or generate a dynamic cube based on a table in your relational database.

A basic dynamic cube contains the following items:

- A measure dimension that contains at least one measure
- At least one dimension
- · At least one hierarchy and associated levels defined for each dimension
- Mappings between the measure and the dimensions
- Attributes that reference table columns either directly, by expressions, or by an expression that is a constant value

For more information, see "Dynamic cubes" on page 19.

When modeling a dynamic cube, the relationship between a measure and a dimension must be defined for each dimension in the cube. This relationship is defined by a measure-to-dimension join. For more information, see "Defining a measure-to-dimension join" on page 54.

Table 14. Properties of a dynamic cube

Property	Description	
Name	The name of the dynamic cube. This is also used as the name of the data source that represents the cube. If the project supports multiple locales, there can be versions of the name in all supported languages. Tip: When creating a Framework Manager package for the dynamic cube, select this name from the list of data sources.	
Comment	A comment or description of the dynamic cube. Comments are not visible in the IBM Cognos studios.	
Remove non-existent tuples	 True (default) - remove tuples from the cross join set that cannot contain any data. False - do not remove tuples from the cross join set. Applies when a dimension has multiple hierarchies and a report contains the cross join of two or more of those hierarchies. With this feature enabled, only those tuples for which data may exist are retained from the cross join, improving report efficiency. A cross join of hierarchies from the same dimension may contain tuples for which no data can possibly exist. For example, in a Time dimension with two hierarchies, the cross join of [2011 Q1] and [2011 Aug] would be removed since [2011 Q1] and [2011 Aug] do not share a common month. 	

Related tasks:

"Defining a dynamic cube manually" on page 51

Because IBM Cognos Cube Designer needs the information provided by foreign keys to determine relationships, only fact tables with foreign keys can be used to generate a dynamic cube. If your database does not use referential integrity, you can manually define a dynamic cube to meet your requirements.

"Defining a dynamic cube based on a relational table"

When you create a dynamic cube based on a relational table, IBM Cognos Cube Designer makes design assumptions based on the relational metadata. In one step, Cognos Cube Designer creates a basic cube structure - a measure dimension containing measures, a set of dimensions, and appropriate mappings to the database. To complete and validate the dynamic cube definition, you manually resolve some issues.

Defining a dynamic cube based on a relational table

When you create a dynamic cube based on a relational table, IBM Cognos Cube Designer makes design assumptions based on the relational metadata. In one step, Cognos Cube Designer creates a basic cube structure - a measure dimension containing measures, a set of dimensions, and appropriate mappings to the database. To complete and validate the dynamic cube definition, you manually resolve some issues.

Before you begin

Because Cognos Cube Designer needs the information provided by foreign keys to determine relationships, only fact tables with foreign keys can be used to generate a dynamic cube. If your database does not use referential integrity, you can manually define a dynamic cube to meet your requirements. For more information, see "Defining a dynamic cube manually" on page 51.

Procedure

- 1. Select a fact table in the **Data Source Explorer**.
- 2. Right-click and select Generate, Cube.

What to do next

Any object causing a modeling issue or requiring further design is identified in the **Project Explorer** and an icon appears next to the object. On the **Issues** tab, you might be presented with actions required to resolve these issues and validate the dynamic cube.

Defining a dynamic cube manually

Because IBM Cognos Cube Designer needs the information provided by foreign keys to determine relationships, only fact tables with foreign keys can be used to generate a dynamic cube. If your database does not use referential integrity, you can manually define a dynamic cube to meet your requirements.

Any object causing a modeling issue or requiring further design is identified in the **Project Explorer** and an icon appears next to the object. You can validate an entire project or an individual object at any time. An effective practice is to validate each object as you create it. Right-click an object in the **Project Explorer** tree and select **Validate**.

Procedure

- 1. Select a namespace in the **Project Explorer** tree.
- 2. Click New Cube .

What to do next

A measure dimension is automatically created. To complete your dynamic cube, define your measures, dimensions, hierarchies, levels and joins.

Model measures

Using IBM Cognos Cube Designer, you can define a measure manually or you can generate a measure based on a column in your relational database. A dynamic cube contains one measure dimension.

For more information, see "Measures" on page 21.

Property	Description
Name	The measure dimension name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Comment	A comment or description of the measure dimension. Comments are not available to the studio users.
Default Measure	During report processing, if no measure is defined for evaluation of a value expression, the default measure is used. The default measure can be a regular or calculated measure. Default: the first measure added to the dynamic cube.

Table 15. Properties of a measure dimension

Table 16. Properties of a measure item

Property	Description		
Name	The measure name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.		
Comment	A comment or description of the measure. Comments are not available to the studio users.		
Expression	The expression can refer to measures in the dynamic cube.		
	The expression cannot contain multidimensional dynamic query mode constructs.		
	This property is only available for measure items that were created in Cognos Cube Designer.		
Column Name	The name of the associated column in the relational database.		
	This property cannot be edited.		
Visible	Controls whether the object is visible in the published package.		
	Non-visible measures are typically used to represent intermediate values in the construction of a complex calculated measure. These measures are not intended to be used for direct reporting. However, a non-visible measure is always present in the published package because the measure might be needed by other objects in a dynamic cube.		
	Non-visible measures do not display in the metadata browser and are removed from the output of reports which contain references to them. For example, a report that references a non-visible measure does not include output from that measure.		
	The default measure cannot be hidden.		
	Default: True		
Data Type	The data type of the associated column in the relational database.		
	This property cannot be edited.		
Precision	The precision of the associated column in the relational database.		
	This property cannot be edited.		
Scale	The scale of the associated column in the relational database.		
	This property cannot be edited.		
Regular Aggregate	The primary method used to aggregate data for the measure.Default: Sum		
Data Format	Set the default data properties for each type of data.		

Defining a measure based on a relational column

In IBM Cognos Cube Designer, you can define a measure based on a relational column. To create measures, add a cube and then create measures in the measure dimension folder under the cube.

For information about creating calculated measures, see "Calculated members" on page 57.

Procedure

- 1. From the **Project Explorer** tree, expand your cube.
- 2. Right-click the measure dimension 🔛 and select **Open Editor**.
- **3**. From the **Data Source Explorer**, drop a table column onto the **Editor** pane.

The mapping to the associated column is automatically created. The **Property** fields are initialized from the table column values.

Defining a measure manually

In IBM Cognos Cube Designer, you can define a measure manually by creating a mapping to a database column or to an expression. To create measures, add a cube and then create measures in the measure dimension folder under the cube.

For information about creating calculated measures, see "Calculated members" on page 57.

Procedure

- 1. From the Project Explorer tree, expand your cube.
- 2. Right-click the measure dimension 🗾 and select **Open Editor**.
- 3. Click **New Measure** Late to add a blank measure.
- 4. To give the new measure a more meaningful name, right-click the new measure and select **Rename**.
- 5. You can complete the measure one of two ways:
 - To map the measure to a table column, drag a table column from the **Data Source Explorer** onto the **Mapping** field.
 - To map the measure to an expression, define an expression in **Expression** property on the **Properties** pane.

Defining aggregation rules

Each measure has a regular aggregation type. The **Regular Aggregate** property identifies the type of aggregation that is applied to the measure. Aggregation rules can be used in addition to the regular aggregate. They specify how semi-aggregate measures are aggregated with respect to information from the dimension.

When you import metadata, IBM Cognos Cube Designer assigns values to the **Data Type**, **Precision**, **Scale** and **Regular Aggregate** properties based on the relational object. For cube measures, you can define aggregation rules for each related dimension.

Aggregate rules are applied in this order:

- 1. The **Regular Aggregate** property is applied to dimensions that are included in the query but do not have assigned **Aggregation Rules**.
- 2. The **Aggregation Rules** are then applied to their specified dimensions, in the order that you specified the rules.
- 3. The report-level aggregation that is specified in the query.

For more information about measures and aggregation rules, see "Measures" on page 21.

Procedure

- 1. Select the Aggregation Rules tab.
- 2. Select a measure in the Measures pane.
- **3**. Select a related dimension from the **Dimension** column.
- 4. Click Include to activate the aggregation rule for the dimension.
- 5. From the **Aggregation Rule** drop-down list, select the aggregation rule to be used for the selected dimension.
- 6. When you have finished adding aggregation rules for the dimension, use **Up**, **Down**, **Top** and **Bottom** to specify the order to apply the aggregation rules.

Defining a measure-to-dimension join

You can define a measure-to-dimension join in a dynamic cube when the level of a join does not match the level of the fact table. You must define the correct measure-to-dimension join to avoid double counting data from the fact table.

For example, a fact table can contain data at the Day level, but it can be joined to the Time hierarchy at the Week level. If the measure-to-dimension join is not defined, measure data equates to actual counts multiplied by the number of days in a week.

Before you begin

You must add the required dimension and measures to a dynamic cube before you can define a measure-to-dimension join. For more information, see "Model dimensions" on page 37 and "Model measures" on page 51.

Procedure

- 1. From the Project Explorer tree, right-click the cube and select Open Editor.
- 2. For each dimension, select Edit.
- **3.** Specify the join by relating columns in the dimension to columns in the measure.
- 4. Specify the relationship operator.
- 5. If required, clear the **Join is at the lowest level of detail for the dimension** check box. IBM Cognos Cube Designer cannot automatically detect that a join is at a higher grain than the lowest level of a dimension.

Deploying and publishing dynamic cubes

When you finish modeling a dynamic cube in IBM Cognos Cube Designer, you can deploy it as an OLAP data source to Content Manager. To work with a deployed cube in the IBM Cognos studios, you must also publish a Framework Manager package for it, configure the cube as a data source, and start the cube.

Important: You must validate a dynamic cube before you can deploy it.

You use the **Publish** option to deploy a dynamic cube. You can also perform the additional tasks required to publish a cube in one step.

Select all options

This option publishes a Framework Manager package for the deployed dynamic cube, then configures and starts the cube.

• Publish the package in: My Folders

By default, the cube name is used as the Framework Manager package name. You can specify a different package name in the **Package Name** box.

Tip: You can move the location of published packages by using IBM Cognos Administration.

Add the dynamic cube to the default dispatcher

This option configures the deployed dynamic cube as a data source.

• Start the dynamic cube

This option starts the dynamic cube, if you also configure the cube as a data source.

• Associate my account and signon with the cube datasource

This option allows you to use credentials to access the data source in the IBM Cognos Studios.

Select if anonymous access is disabled. Your account must use associated credentials. Go to the **Personal** tab in the **Set preferences** dialog of the IBM Cognos Portal, and create your credentials.

Important: Because these options use default settings, they are intended for deploying and testing a dynamic cube in a development environment rather than a production environment.

Procedure

- 1. Open the project that contains the dynamic cube that you want to deploy and publish.
- 2. In the **Project Explorer** tree, right-click the required cube, and then select **Publish**.
- 3. Select the additional options required to publish the cube.
- 4. Click OK.

Results

When the deployment and publish process is complete, a confirmation message is shown.

Chapter 7. Advanced dynamic cube modeling

After you create a basic dynamic cube in IBM Cognos Cube Designer, there are numerous ways to enhance the functionality of the cube.

You can perform the following tasks:

- add calculated members and measures
- model relative time dimensions
- · use multi-locales and associated formatting

Calculated members

Calculated members add business logic into dimensions by introducing members whose value is computed from the values present within the underlying data.

The new members are available for use without being added to the underlying relational data source. A calculated member is defined by a dimensional expression.

A calculated measure is a calculated member that belongs to the measure dimension. There are no behavior differences between calculated members and calculated measures.

For more information, see "Calculated members in reports" on page 115.

For information about relative time calculated members, see "Model relative time dimensions" on page 63.

Property	Description
Name	The name that appears in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Parent Member	Specifies the parent of the calculated member in the member tree.
Expression	Defines the value of the calculated member using other members and a valid set of multidimensional operators and functions.

Table 17. Properties of a calculated member

Table 18. Properties of a calculated measure

Property	Description
Name	The name that appears in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Expression	Defines the value of the calculated measure using other members and a valid set of multidimensional operators and functions.
Data Format	Set the default data properties for each type of data.

Table 18. Pro	perties of a	calculated	measure	(continued)
---------------	--------------	------------	---------	-------------

Property	Description
Visible	Controls whether the object is visible in the published package.
	Non-visible measures are typically used to represent intermediate values in the construction of a complex calculated measure. These measures are not intended to be used for direct reporting. However, a non-visible measure is always present in the published package because the measure might be needed by other objects in a dynamic cube.
	Non-visible measures are not displayed in the metadata browser and are removed from the output of reports which contain references to them. For example, a report that references a non-visible measure does not include output from that measure.
	Default: True
Regular Aggregate	The primary method used to aggregate data for the measure.
	Default: Sum

Author calculated member expressions

Cognos Cube Designer validates the syntax of expressions. After a cube is started, the dynamic cube engine validates the semantics of the calculated member and calculated measure expressions.

The expression editor does not limit functions to the valid ones for a specific context.

There are some restrictions that apply to Cognos Dynamic Cubes calculated members.

Do not use the following relational constructs in expressions used to define calculated members:

- Value summary functions (Not Member Summary functions)
- Value Analytic functions (rank, first, last, percentile, percentage, quantile, quartile, distinct clause, prefilter clause) (Summaries/Member Summaries)
- Value Summary functions (standard-deviation-pop, variance-pop, distinct clause, prefilter clause)
- All running- or moving- summary functions (Summaries)
- All FOR clauses in aggregate functions (Summaries/Member Summaries)
- Date/time constants (Constants)
- All business date/time functions (Business Date/Time functions)
- Like, lookup, string concat '| |', trim, coalesce, cast (Common Functions)
- MOD function (Common Functions)

Calculated members examples

IBM Cognos Cube Designer allows for the definition of dimensional calculated members and measures. Such expressions were previously defined only in the reporting environment. When defined in a dynamic cube, the calculated members are accessible in all of the Cognos BI studios. You can use calculated measures so you can use constant or weighted values You can create calculated members that represent an N period rolling window of data relative to a current period member.

Constant and weighted allocation

Measures in base dynamic cubes usually have the same grain since each base cube is constructed from a single fact table. When combining cubes into a virtual cube, it is possible for a measure from one cube to be valid only for a subset of the levels of a virtual hierarchy.

As an example, the virtual cube Sales Inventory is built from two base cubes: Sales and Inventory. The Sales cube has the measure Sales Amount and its Time hierarchy contains Year and Quarter levels. The Time hierarchy in the Inventory cube also contains a Month level. When creating the Sales Inventory cube, the virtual Time hierarchy is created with the Year, Quarter, and Month levels

In this situation, any Sales Amount value that is computed in the virtual cube at the Day level is null since there is no value in the Sales cube at the Month level.

In the following diagram, the Sales Amount measure has no values at the Month level but the Stock measure, from the Inventory cube, does.



Figure 12. Example of differences in the time hierarchy for two cubes

You can use calculated measures to compute constant or weighted values for a measure such as Sales Amount. A constant allocation allocates a measure's value from a higher level (typically the lowest at which the measure is valid, or 'in scope') evenly across all of its descendants at each level below the 'in scope' level.

Using constant allocation, the following diagram shows how Sales Amount values would now appear:



Figure 13. Example of the use of constant allocation

The values from the Quarter level are evenly distributed across the descendants at the Month level.

A weighted allocation performs a similar operation, but allocates values to the descendants relative to the values of another measure which is 'in scope', and which is correlated with the measure being allocated so the allocation is reasonable.

For example, the Sales Amount values are allocated based on the weights of the Stock measure from the Inventory cube.

The Sales Amount values from the Quarter level are now distributed using the same weighting of the Stock measure.



Figure 14. Example of the use of weighted allocation

For the constant allocation example, the expression for a calculated measure called [Sales] in the virtual cube would be as follows:

```
if (roleValue('_levelNumber', currentmember([Sales Inventory].[Time].[Time])) > 1 ) then
    tuple([Sales Amount], ancestor(currentmember([Sales Inventory].[Time].[Time]),
                                   [Sales Inventory].[Time].[Time].[Quarter]))
    /
        count(1 within set
              descendants(
```

```
ancestor(currentmember([Sales Inventory].[Time].[Time]),
                            [Sales Inventory].[Time].[Time].[Quarter]
                  ),
                  roleValue('_levelNumber',
                             currentmember([Sales Inventory].[Time].[Time])
                  ) - 1,
                  self
              )
        )
else
[Sales Amount]
```

In the weighted allocation example, the expression for a calculated measure called [Sales] in the virtual cube would be as follows.

```
if (roleValue(' levelNumber', currentmember([Sales Inventory].[Time].[Time])) > 1 ) then
(
    tuple([Sales Amount], ancestor(currentmember([Sales Inventory].[Time].[Time]),
                                   [Sales Inventory].[Time].[Time].[Quarter])
    )
    tuple([Stock], currentmember([Sales Inventory].[Time].[Time])
    )
    tuple([Stock], ancestor(currentmember([Sales Inventory].[Time].[Time]),
                            [Sales Inventory].[Time].[Time].[Quarter])
    )
else
(
    [Sales Amount]
```

)

FIRST and LAST aggregation

Some measures, such as account balances and inventory levels, consist of opening or closing values over a particular time period. These measures are additive across all dimensions except time.

A data warehouse can record the closing balance of customer bank accounts. The value of this measure is the value for the last leaf member within the time period. For example, the account balance for [2012 May] is the value of the member [2012 May 31], and not the sum of the account balance for all the days of May.

IBM Cognos Dynamic Cubes does not directly support FIRST or LAST aggregation of measure values. However, it is possible to achieve similar behavior using a calculated member to compute the value for such a measure for any time period. You cannot use calculated measures to compute the FIRST or LAST value within an arbitrary set of tuples. Calculated measures cannot exhibit the semi-additive nature of such measures. The following example shows the use of a calculated member in authoring business reports:

Rolling N period window

You can create calculated members that represent an N period rolling window of data relative to a current period member.

For example, to create a member that represents a six month rolling window of data, you can use the following calculated member expression: LastPeriods(6, [Sales].[Time].[Time].[All Time].[Current Year (2012)])

To create a member that represents a previous six month rolling window, you can use the following calculated member expression: LastPeriods(6, Lag([Sales].[Time].[All Time].[Current Year (2012)], 6))

Note: Either expression, especially the second, can return a set of less than six members depending upon how the time hierarchy is structured. Additional conditional logic is required to account for current time period members which are positioned less than the desired number of periods (members) from the first member of the level in which they exist.

Relative time members contain a subset of the overall time hierarchy and are constrained in terms of which members are available for an N period rolling window. For example, you can not perform a three year rolling window of data using the current year member since only the current year and prior year are available.

To access a larger window, you would require an expression such as:

The intent of this expression is to navigate from the sub tree of relative time calculated members which are constrained to a set of current/prior periods, into the portion of the tree that contains the regular members of the hierarchy.

Defining a calculated member

You define calculated members in the expression editor using dimensional constructs and functions. You can define a calculated member based on a calculated member.

Calculated members are added to the member tree as a children of the parent member. You identify the parent member by selecting a member from the member tree under the Members folder of a hierarchy.

If there is no ALL member, the calculated member does not have to have a parent defined. The calculated member then becomes a member of the root level. If there is an ALL member, the calculated member must have a named parent and if one is not specified, the calculated member will fail to load. The failure is recorded in the log file.

It is a good practice to use a naming convention so that you and your report users can easily identify calculated members.

Procedure

- 1. From the Project Explorer, click a dimension and expand it.
- 2. Right-click a hierarchy that belongs to the dimension and select Open Editor.
- 3. Expand the hierarchy to access the Members folder.
- 4. Expand the member tree until you can view the member you wish to define as the parent of your new calculated member.
- 5. Select the Calculated Members tab.
- 6. Click New Calculated Member 📑
- 7. Select the new calculated member.
- 8. To set the **Parent Member** on the **Properties** pane, drag a member from the member tree in the **Project Explorer**. This property specifies the position of the calculated member in the member tree.
- **9**. From the **Properties** pane, define the calculated member in the **Expression** property.
 - To use an object from the project, drag the item from the **Project Explorer** into the expression.
 - To use a calculated member, drag the calculated member from the member tree.
 - To add functions, summaries and operators, select the **Functions** tab, 🔯 to access the required elements.
- 10. Right-click the Members folder of the hierarchy and select Refresh.

Results

The new calculated member is displayed under the **Calculated Members** folder for the hierarchy. The calculated member can also be seen under the parent member in the **Members** folder of the hierarchy.

Model relative time dimensions

IBM Cognos Dynamic Cubes relative time members are specialized calculated members that are added to a time hierarchy at the time a cube is started.

IBM Cognos Cube Designer optionally creates a fixed set of relative time members in a time hierarchy. You can also create your own relative time calculated members.

Defining a relative time dimension

When you use relative time functionality, you can author reports relative to the current period. These reports can be executed at any time and remain valid based on the value of the current period at the execution time. To use relative time, you define a dimension as a time dimension, modify time properties for the level and generate relative time members on a hierarchy by hierarchy basis.

The predefined relative time members are Current Period, Prior Period, Current Period to Date, Prior Period to Date, Current Period to Date Change, Prior Period to Data Change, Current Period to Date Growth and Prior Period to Date Growth. The predefined relative time members are used by simply dragging them into any report.

The modeler can create additional calculated members that are based on these members. The report author can create additional expressions that are based on these members.

Each level has a **Current Period** property. The current period property of a level is used to filter members by their level key value in order to identify the single leaf member which is the current period member in the hierarchy. This is the basis for defining the current member at each level in the hierarchy. If a Current Period expression is defined, it is used to filter members at that level by the value of the level key for that level. The expression can be static, based on a current date/time value, or based on a value in the relational database typically populated by the ETL process.

The levels in a time hierarchy with relative time enabled are not required to have a Current Period. If no Current Period expressions are defined, the Current Period used is the rightmost, most recent leaf level member of the hierarchy.

The combination of level current period expressions is used to identify a specific leaf member. You can determine which member is used as Current Period by examining the levels of the hierarchy in a top-down manner. If there are levels with no current period expression defined, the member chosen at each level is the rightmost, most recent child of the member selected from the previous, higher level. As soon as a level is encountered where a Current Period expression is defined, the default selection of members at the higher levels is ignored and the member at that level which determines the path to the leaf level current period begins with the member defined by the expression. It is possible to define the current period of a hierarchy by providing a current period at the leaf level.

When viewing a time hierarchy, the caption of the relative time members that appear in the hierarchy do not use the current period expressions defined within the project. The product simply uses the right most/recent member at each level as the current period for that level.

Resolution of calculated member expressions is performed when a cube is started or member cache refreshed.

You cannot have a hierarchy with relative time members and non-default security rules.

Procedure

- 1. Select the location from which you want create the dimension:
 - To create a shared dimension at the project level, select **Model** from the **Project Explorer** tree.
 - To create dimension within a dynamic cube, select the cube from the **Project Explorer** tree.

Tip: Create one relative time dimension and use it in all your dimensions to avoid conflicts between multiple time dimensions.

- 2. Click **New Dimension ?**. The dimension contains a set of initial objects you need to complete the dimension.
- 3. On the **Properties** pane of the dimension, set **Dimension Type** to **Time**.
- 4. On the **Properties** pane of a hierarchy that belongs the dimension, set **Add relative time members** to **True**. This enables generation of the predefined relative time members.
- 5. Build your desired level structure. For more information about creating levels, see "Defining a level" on page 42.
- 6. For each time level, select a **Level Type**. The levels must appear in order in the hierarchy. For example, the Year, Month, Day levels cannot appear as Year, Day, Month. Use the **Periods** level type when the level does not conform to one of the predefined level types.
- 7. For each time level, enter an expression in the **Current Period** property. For some examples of current period expressions, see "Examples of level current period expressions."
- 8. With the time dimension selected, right-click, and select **Refresh Members**. Predefined calculated members for relative time are added to the member tree. Cognos Cube Designer does not use of the **Current Period** expression when populating the relative time members. The most recent member at each level is used instead. However, the members may all be used within calculated member/measure expressions since the member identifiers remain constant; only their captions and what they refer to that changes.

Examples of level current period expressions

Some common examples of level current period expressions are defined in the following list.

Year

extract(year, localtimestamp)
⅓ Year

```
if(extract(month, localtimestamp) < 7) then
 (1)
else
 (2)</pre>
```

Quarter

The curent_timestamp function returns Greenwich Mean Time while the localtimestamp function returns local time.

Month

extract(month, localtimestamp)

Week of Year

cast(extract(year, localtimestamp), varchar(4))
|| 'W' || cast(week of year(localtimestamp), varchar(2))

Day of Year

cast(extract(year, localtimestamp), varchar(4))
|| 'W' || cast(_week_of_year(localtimestamp), varchar(2))

Day of Week

_day_of_week(localtimestamp, 7)

Day of Month

days between(localtimestamp, first of month(localtimestamp)) + 1

Hour

extract(hour, localetimestamp)

Week of Month

Calculated member example - creating a 24 month rolling window

The following example shows the process and code required for creating the expression to define a 24 month rolling window relative to the current time period. The final expression can be used within a calculated member created within the [Time] dimension in the Cognos Cube Designer, making it available for use in all of the studios

The expression ties a relative time member back to its corresponding member in the same hierarchy which is not a relative time member.

The reason for doing this is that relative time members are special calculated members which have parent/child and sibling relationships - but only relative to one another.

For example, you can obtain the children of the 'Current Year' member but you cannot apply the PREVMEMBER function to the Current Year member to obtain the prior year.

In the examples below, the expressions are formatted to make it easier to identify the various functions and their operands.

In order to find the member which serves as the basis for the 'Current Year' member, an expression such as following is required:

```
ITEM(
  FILTER(
    MEMBERS (
      MyCube].[Time].[Time].[Year]
    ).
    ROLEVALUE(
      ' businessKey',
      CURRENTMEMBER (
        [MyCube].[Time].[Time]
      )
    )
    =
    ROLEVALUE(
      ' businessKey'
      [Current Year (2012)]
    )
  ),
  0
)
```

The sample expression is based on the following conditions:

- the lowest level in the [Time] hierarchy is the [Month] level
- the time hierarchy contains the levels Year/Quarter/Month
- the Month key is based on the Year and Month level identifiers.

The expression to find corresponding member upon which [Current Month (Sept 2012) is based, is as follows:

```
ITEM(
  FILTER(
    DESCENDANTS (
      FILTER(
        MEMBERS (
          [MyCube].[Time].[Time].[Year]
        ).
        ROLEVALUE(
          ' businessKey',
          CURRENTMEMBER (
             [MyCube].[Time].[Time]
          )
        )
        ROLEVALUE(
           ' businessKey'
          [Current Year (2012)]
        )
      ),
      [MyCube].[Time].[Time].[Month],
      SELF
```

```
),
ROLEVALUE(
    '_businessKey',
    CURRENTMEMBER(
       [MyCube].[Time].[Time]
    )
)
=
ROLEVALUE(
    '_businessKey',
    [Current Month (Sept 2012)]
)
),
0
```

The previous 24 months is then built upon that expression, making use of the LASTPERIODS function. This expression can be used within a calculated member created within the [Time] dimension in the Cognos Cube Designer.

```
LASTPERIODS(
 24,
  ITEM(
    FILTER(
      DESCENDANTS (
        FILTER(
          MEMBERS (
             [MyCube].[Time].[Time].[Year]
          ),
          ROLEVALUE(
             ' businessKey',
             CURRENTMEMBER (
               [MyCube].[Time].[Time]
            )
          )
          ROLEVALUE(
             '_businessKey',
            [Current Year (2012)]
          )
        ),
        [MyCube].[Time].[Time].[Month],
        SELF
      ),
      ROLEVALUE(
         ' businessKey',
        CURRENTMEMBER (
          [MyCube].[Time].[Time]
        )
      )
      =
      ROLEVALUE(
         ' businessKey',
        [Current Month (Sept 2012)]
      )
    ),
    0
 )
)
```

This expression can be used within a calculated member created within the [Time] dimension in the Cognos Cube Designer, making it available for use in all of the studios.

Multiple locales

You can add support for multiple locales to IBM Cognos dynamic cubes. Metadata object names and captions, dynamic cube object names, and member attribute names can be assigned different values in different locales. Then when a user switches between different content languages in IBM Cognos Connection, names and captions are displayed in the appropriate language.

You use IBM Cognos Cube Designer to add support for multiple locales to a project, and you can then add metadata object names and member attribute names and captions in multiple languages. After adding multiple language support, you publish the dynamic cube in the normal fashion.

Selecting the design language and supported locales

When creating a project in IBM Cognos Cube Designer, the design language of the project defaults to the locale setting of the computer. You can change the default design language. Normally the default design language is the locale or language of the data in the database. After the design language is set, you can add other supported locales to the project.

Procedure

- 1. To change the design language, in the **Properties** tab of a project, click the value of the **Design Language** and select the design language from the drop-down list.
- **2.** To add locales, in the **Properties** tab of a project, click **Add Locale(s)** and check the boxes next to the required locales.

Adding multiple locale names to metadata objects and dynamic cube objects

You can add names in multiple languages to metadata objects for supported locales.

Procedure

- 1. In **Project Explorer**, click a metadata object, such as a dimension, or a dynamic cube object, such as a measure.
- 2. In the **Properties** tab, click the value of the **Name** property. The supported locales for the project are displayed.
- 3. For each supported locale, enter a name for the object in that language.
- 4. You can add additional locales to the project by clicking the **Add Language** button. This adds locales to the project, not just to the selected object.
- 5. If the metadata object is a hierarchy, you can add language versions for the **Root Caption** property using the same steps.

Adding support for multiple locales to members and attributes

You add support for multiple locales for members and attributes by dimension. It is not necessary that all dimensions in a dynamic cube support multiple locales. IBM Cognos Dynamic Cubes supports dynamic cube definitions in which only some dimensions have members with multiple locales.

Before you begin

If you are adding multiple locales to attributes, the data source must contain a column for each locale associated with the attribute. For example, the **Great Outdoors Warehouse** data source has a **Product line** attribute in the **Products** dimension. This attribute has columns named PRODUCT_LINE_EN, PRODUCT_LINE_FR, and so on, for each of the supported locales in the database.

Procedure

- 1. In **Project Explorer**, click a dimension for which you want to add support for multiple locales.
- 2. In the **Properties** tab, click the value for **Multilingual Support** and select **By Column**.

You can now provide multilingual names for members and attributes.

- **3**. Perform the following steps for each member in the dimension that you want to give names in multiple languages.
 - a. In **Project Explorer**, click a member in the dimension.
 - b. In the **Properties** tab, click the value of the **Name** property. The supported locales for the project are displayed.
 - c. For each supported locale, enter a name for the member in that language.
 - d. You can add additional locales to the project by clicking the **Add Language** button.
- 4. Perform the following steps for each attribute in the dimension that you want to give names in multiple languages.
 - a. In **Project Explorer**, click an attribute in the dimension.
 - b. In the **Properties** tab, click the value of the **Name** property. The supported locales for the project are displayed.
 - c. For each supported locale, enter a name for the attribute in that language.
 - d. You can add additional locales to the project by clicking the **Add Language** button.
 - e. In the **Properties** tab, change the value of the **Multilingual** property to true.
 - f. In the **Properties** tab, click the value of the **Column Name** property. The supported locales for the project are displayed.
 - g. Expand the data source in **Data Source explorer**, and drag the column associated with each locale into the respective **Column Name** value.

For example, the **Great Outdoors Warehouse** data source has a **Product line** attribute in the **Products** dimension. This attribute has columns named PRODUCT_LINE_EN, PRODUCT_LINE_FR, and so on, for each of the supported locales in the database. If you are enabling multilingual support for a dynamic cube that uses the **Product line** attribute in this database, you would drag the PRODUCT_LINE_EN column into the **Column Name** value for English, the PRODUCT_LINE_FR column into the **Column Name** value for French, and so on.

Use Cognos Framework Manager to publish multilingual packages

You can use IBM Cognos Framework Manager to publish multilingual packages when a multilingual dynamic cube has already been published in Content Manager.

Open the project file in Cognos Framework Manager and choose the same design language that was used for the dynamic cube. You must also choose an IBM Cognos Dynamic Cubes data source for the project. Create the default package and publish it to Content Manager. Cognos Framework Manager creates multiple locale namespaces that match the multiple locale data source names in Content Manager. It also creates multiple locale names packages so that if the user switches languages, names are displayed in that language.

For more information, see the IBM Cognos Framework Manager User Guide.

Chapter 8. Aggregate cube modeling

In IBM Cognos Cube Designer, you can model aggregate cubes within a dynamic cube when the imported data source for a dynamic cube contains fact tables with pre-aggregated data.

For more information about pre-defined aggregate fact tables, see "Aggregate tables" on page 27.

After publishing a dynamic cube that contains aggregate cubes, when you run queries on the cube data source, IBM Cognos Dynamic Cubes analyses these queries and redirects them to the appropriate aggregate table in the data source.

You must be familiar with the fact data in the data source to model an aggregate cube. Understand which fact tables are set up as aggregates, and which detail tables the fact tables relate to.

Tip: It is good practice to prefix aggregate table names in the relational database with "Aggregate" so that you can easily identify them. You can also use the Relational Explorer to check the relationships between fact tables.

Before you can start modeling an aggregate cube, you must set up the dynamic cube and aggregate tables by performing the following tasks:

- 1. For level-based hierarchies only, create the hierarchy levels required for aggregation if they do not exist in the dimension. For example, if an aggregate table in the data source summarizes data by quarter, the Date dimension must include a Quarter level.
- 2. For each aggregation level in the dimension, ensure that the required attributes and level unique keys are defined.
- **3.** Aggregate tables must contain data at the highest level of aggregation used by the aggregate cube so that you can roll up dimensions to the required level.

For example, if a Time dimension contains Year, Quarter and Month levels, and you want to roll up data up to the Year level in an aggregate cube, the aggregate table usually contains data at the Year level.

If Cognos Dynamic Cubes cannot match a rollup level to an aggregate table, it uses an aggregate table defined at a particular level of aggregation to satisfy higher level aggregate requirements. For example, if you want to roll up the Time dimension to the Year level, and the aggregate table only contains data at the Quarter level, it uses this aggregate table and rolls it up to the higher levels.

Model aggregate cubes

The way in which you model an aggregate cube depends on the data it contains:

- Simple aggregate cube
- An aggregate table uses level unique keys or is joined to a separate dimension that contains the required levels for aggregation. You must map the relevant dimension in the aggregate cube to a separate dimension aggregate table.
- Aggregate cube with a parent-child dimension

• A parent-child dimension does not have hierarchy levels. You create the relationships by mapping a single column in the aggregate table to the child key in the parent-child dimension.

You can also filter data in an aggregate cube by using aggregate slicers. This filtering is possible where the data source contains a set of aggregate tables, each providing a subset of the data set available. For example, an aggregate table can contain sales data for specific dates.

The following table lists the properties that you can set when modeling an aggregate cube.

Property	Description
Name	The name of the aggregate cube. If the project supports multiple locales, there can be versions of the name in all supported languages.
Comment	A comment or description of the aggregate cube.
Remove non-existent tuples	This property is applicable to the dynamic cube only and should not be edited.
Ordinal	The order in which the dynamic query mode server redirects queries to an aggregate cube. If there are multiple aggregate cubes that can satisfy a query, the aggregate cube with the lowest ordinal value is used. The aggregate cube with the lowest ordinal value is selected first.

Table 19. Properties of an aggregate cube

Defining a simple aggregate cube automatically

You define a simple aggregate cube when primary keys in the aggregate table match the level keys in dimensions of a dynamic cube. This allows you to create relationships between the dimensions and the aggregate table.

IBM Cognos Cube Designer can create these relationships automatically if the aggregate table contains the following:

- Measures that match the measures in the aggregate cube.
- Dimensions that match the dimensions in the aggregate cube.
- Data at the highest level of aggregation required by the aggregate cube.

Procedure

- 1. Open the Cube editor for the dynamic cube in which you want to define an aggregate cube.
- 2. Click the Aggregates tab.
- **3.** Drag the required aggregate table from the **Data Source Explorer** to the **Aggregates** tab.

An aggregate cube is created in the **Aggregates** tab. The cube also appears under the **Aggregate Cubes** folder in the **Project Explorer** tree. Where matching measures and dimensions are found in the aggregate cube, Cognos Cube Designer maps each of these items to the aggregate table. Where possible, it also attempts to identify the highest level of aggregation required and roll up dimensions. The ability to automatically map is dependent on how the aggregate tables are set up.

Results

The aggregate cube is now complete. You can fine-tune the mapping by following step 4 onwards in the topic "Defining a simple aggregate cube manually." When you finish, you can test the validity of the aggregate cube. For more information, see "Validate a project and individual objects" on page 35.

Defining a simple aggregate cube manually

You define a simple aggregate cube when an aggregate table uses level unique keys or is joined to a separate dimension that contains the required levels for aggregation. For example, to improve query performance, if a dimension table contains many records, you decide to create a dimension table that does not contain the lowest level members and contains only the level unique keys of its members. In this instance, you must map the relevant dimension in the aggregate cube to a separate dimension aggregate table.

Procedure

- 1. Select the dynamic cube in which you want to define an aggregate cube from the **Project Explorer** tree.
- 2. Click New Aggregate Cube 🔩 .
- **3**. Select the measures and dimensions to include in the aggregate cube, then click **OK**.

An aggregate cube is created, which also appears under the **Aggregate Cubes** folder in the **Project Explorer** tree.

By default, each dimension is mapped to the lowest dimension level defined in the detail fact table. If aggregation occurs at a higher level in the aggregate table, you must roll up dimensions in the aggregate cube to the correct level.

4. In the **Project Explorer** tree, double-click the aggregate cube in the **Aggregate Cubes** folder.

The Aggregate cube editor is shown.

5. Click the dimension to roll up, and select the required level from the list of levels shown.

Repeat this step for each dimension you want to roll up.

For dimensions that are mapped to a separate dimension aggregate table, you must now map the level unique keys in the dimensions to columns in the required aggregate table.

- 6. In the Aggregate cube editor, click the Key Mappings tab.
- 7. For each level unique key, drag a column from the required aggregate table in the **Data Source Explorer** to the **Mapping** field.

Tip: If you drag a whole aggregate table, IBM Cognos Cube Designer attempts to automatically map all level unique keys.

Now you must map measures in the aggregate cube to columns in the aggregate table.

In the Aggregate cube editor, click Measures 1.
 The Measures editor is shown.

 Map each measure to a column in the aggregate table by dragging a column from the required aggregate table in the Data Source Explorer onto the Mapping field.

For those dimensions where primary keys in the aggregate table match the level unique keys in dimensions of the dynamic cube, you can now create the relationships between dimensions and measures in the aggregate cube.

10. In the **Project Explorer** tree, double-click the aggregate cube in the **Aggregate Cubes** folder.

The Aggregate cube editor is shown.

- 11. For each dimension, click **Edit**, then select the dimension primary key and measure key to which it is joined.
- 12. If required, define the measure-to-dimension join in the Join is at the lowest level of detail for the dimension check box.

For more information about this check box, see "Defining a measure-to-dimension join" on page 54.

13. Click OK.

Results

The aggregate cube is complete. You can now test the validity of the aggregate cube. For more information, see "Validate a project and individual objects" on page 35.

Defining an aggregate cube with a parent-child dimension

An aggregate cube can contain a parent-child dimension. Because the dimension does not have hierarchy levels, you create the relationships by mapping a single column in the aggregate table to the child key in the parent-child dimension.

The aggregate cube can also contain dimensions with level-based hierarchies. For more information about adding these dimensions, see "Defining a simple aggregate cube manually" on page 73.

Procedure

- 1. Select the dynamic cube in which you want to define an aggregate cube from the **Project Explorer** tree.
- 2. Click New Aggregate Cube 🌆 .
- **3**. Select the measures and parent-child dimension to include in the aggregate cube, then click **OK**.

An aggregate cube is created, which also appears under the **Aggregate Cubes** folder in the **Project Explorer** tree.

Now map a single column in the aggregate table to the child key in the parent-child dimension.

4. In the **Project Explorer** tree, double-click the aggregate cube in the **Aggregate Cubes** folder.

The Aggregate cube editor is shown.

- 5. Select the parent-child dimension, then select the **I want to remap the columns** for this dimension, as they are included in my aggregate check box.
- 6. Click the Key Mappings tab.
- 7. Drag a column from the required aggregate table in the **Data Source Explorer** to the **Mapping** field for the child key.

Next, you must map measures in the aggregate cube to columns in the aggregate table.

8. In the Aggregate cube editor, click Measures 🕍 .

The Measures editor is shown.

 Map each measure to a column in the aggregate table by dragging a column from the required aggregate table in the Data Source Explorer onto the Mapping field.

Results

The aggregate cube is complete. You now test the validity of the aggregate cube. For more information, see "Validate a project and individual objects" on page 35.

Filtering data using an aggregate slicer

You can filter the data in an aggregate cube using aggregate slicers. Filtering is possible where the data source contains a set of aggregate tables, each one providing a subset of the data set available. For example, a data warehouse might contain five years of sales data, and also contain aggregate tables with sales data summarized for each quarter.

Procedure

1. Define the aggregate cube you require.

For more information, see "Defining a simple aggregate cube automatically" on page 72, "Defining a simple aggregate cube manually" on page 73, and "Defining an aggregate cube with a parent-child dimension" on page 74.

- 2. Double-click the aggregate cube in the **Project Explorer** tree, then click the **Slicers** tab.
- **3**. Select the data to include in the filter by dragging and dropping members from the **Members** folder in the **Project Explorer** tree to the **Member Slicers** field.

Note: All selected members must come from a single hierarchy level.

Results

The aggregate cube is complete. You can now test the validity of the aggregate cube. For more information, see "Validate a project and individual objects" on page 35.

Chapter 9. Virtual cube modeling

Using IBM Cognos Cube Designer, you can model virtual cubes in a project.

For information about using virtual cubes, see "Virtual cubes" on page 24.

The following table lists the properties that you can set when modeling a virtual cube.

Table 20. Properties of a virtual cube

Property	Description	
Name	The name of the virtual cube. This is also used as the name of the data source that represents the cube. If the project supports multiple locales, there can be versions of the name in all supported languages. Tip: When creating a Framework Manager package for the virtual cube, select this name from the list of data sources.	
Comment	A comment or description of the virtual cube. Comments are not visible in the IBM Cognos studios.	
Merge Operator	The method used to aggregate data in the source cubes. Default: Sum The cube merge operator is the default merge operator for all virtual measures and virtual members. You can also define a merge operator for specific a virtual measure or virtual member which overrides the cube merge operator.	

Defining a virtual cube

You define a virtual cube at the project level.

Procedure

- 1. Select a namespace in the **Project Explorer** tree.
- 2. Click New Virtual Cube 🜌 .
- **3**. Select a maximum of two source cubes to merge into a virtual cube. You can include dynamic cubes from the current project, and dynamic cubes or virtual cubes deployed as data sources to the content store:
 - To include a dynamic cube from the project, select the cube from the list.
 - To include a dynamic cube or virtual cube from the content store, click Add Content Store Cube, select the required data source, then click OK.
- 4. Click OK.
- 5. Complete the virtual cube definition by using the Properties tab.

You can view the source cubes from which the virtual cube is derived.

- 6. From the **Project Explorer** tree, right-click the virtual cube and select **Open Editor**. You can perform the following tasks from here:
 - To add a source cube, click Add Source Cube 💽.
 - To delete a source cube, select the cube name, and click **Delete** 🔀 .

• To view the virtual measure dimension, click Measures 🕍 .

What to do next

You can now fine-tune virtual objects and define further objects as required. For more information, see "Model virtual dimensions," "Model virtual hierarchies" on page 79, "Viewing virtual levels" on page 81, "Model virtual members" on page 81, and "Model virtual measures" on page 83.

You can also add calculated measures or calculated members to a virtual cube. For more information, see "Calculated members" on page 57.

When you finish, you can test the validity of the virtual cube to check for errors, and then deploy and publish the virtual cube. For more information, see "Validate a project and individual objects" on page 35 and "Deploying and publishing dynamic cubes" on page 54.

Tip: If a virtual cube contains a source cube deployed as a data source to the content store, the data source must be started before you can deploy the virtual cube.

Model virtual dimensions

When you create a virtual cube, IBM Cognos Cube Designer adds dimensions from the source cubes to the virtual cube.

Dimensions with identical names in the source cubes (conformed dimensions) are added to a virtual cube as merged virtual dimensions. Non-conformed dimensions are added to a virtual cube as new virtual dimensions. For examples of the merging process, see "Virtual cubes" on page 24.

If a virtual dimension is not merged correctly, or could not be automatically merged, you can manually merge two source dimensions. You can also delete redundant virtual dimensions.

When merging dimensions in a virtual cube, it is not possible to map a source dimension to more than one virtual dimension.

The following table lists the properties that you can set when modeling a virtual dimension.

Property	Description
Name	The name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Comment	A comment or description of the virtual dimension. Comments are not visible in the IBM Cognos studios.
Default Hierarchy	The hierarchy to use when no hierarchy been specified for a dimension used in an expression. Applies only when multiple hierarchies are defined for a dimension.

Table 21. Properties of a virtual dimension

Table 21. Properties of a virtual dimension (continued)

Property	Description	
Dimension Type	Regular (default) - Identifies a regular dimension. Time - Identifies a time dimension. For more information about relative time dimensions, see "Defining a relative time dimension" on page 63.	

Defining a virtual dimension

Using IBM Cognos Cube Designer, you can define virtual dimensions within a virtual cube.

Procedure

- 1. From the **Project Explorer** tree, right-click the virtual cube and select **Open Editor**. The editor tab shows the following columns:
 - Virtual dimensions the virtual dimensions added to the virtual cube.
 - Dimensions the dimensions in the source cubes to which the virtual dimension is mapped.
- 2. To manually merge source dimensions into a new virtual dimension, follow these steps:
 - a. Click Add Virtual Dimension.
 - b. Click **Editor** for the source dimension column related to the new virtual dimension, then select a source dimension, and click **OK**.

Tip: If you cannot select a source dimension because it is already mapped to a different virtual dimension, you must first delete the source dimension from the other virtual dimension.

- c. Repeat the step b for the second blank source dimension.
- **3**. You can also perform the following tasks from here:
 - To delete a source dimension from a virtual dimension, select the source dimension, and click Delete X.
 - To delete a virtual dimension from a virtual cube, select the virtual dimension, and click **Delete**.
- 4. To complete the definition of a virtual dimension, select the virtual dimension in the **Project Explorer** tree to display the **Properties** tab.

Model virtual hierarchies

When you create a virtual cube, IBM Cognos Cube Designer adds hierarchies from the source cubes to the virtual cube.

Hierarchies with identical names in the source cubes (conformed hierarchies) are added to a virtual cube as merged virtual hierarchies. Non-conformed hierarchies are added to a virtual cube as new virtual hierarchies. For examples of the merging process, see "Virtual cubes" on page 24.

If a virtual hierarchy is not merged correctly, or could not be automatically merged, you can manually merge two source hierarchies. You can also delete redundant virtual hierarchies.

When merging hierarchies in a virtual cube, it is not possible to map a source hierarchy to more than one virtual hierarchy.

The following table lists the properties that you can set when modeling a virtual hierarchy.

Property	Description	
Name	The name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.	
Comment	A comment or description of the virtual dimension. Comments are not visible in the IBM Cognos studios.	
Default Member	The member value to use when evaluating member expressions, where no value is specified for a hierarchy.	
	If the default member is empty, the root member of the hierarchy is used.	
	To set a default member, drag the required member from the Members folder in the Project Explorer tree.	
Parent-Child	True - Indicates that the hierarchy uses a parent-child structure.	
	False - Indicates that the hierarchy does not use a parent-child structure.	
	This property cannot be edited.	
Add Relative Time Members	False (default) - The hierarchy does not belong to a Time dimension.	
	True - The hierarchy belongs to a Time dimension.	
	For more information, see "Defining a relative time dimension" on page 63.	

Table 22. Properties of a virtual hierarchy

Defining a virtual hierarchy

Using IBM Cognos Cube Designer, you can define virtual hierarchies within a virtual cube.

Procedure

- 1. From the **Project Explorer** tree, right-click the virtual dimension for which you want to define virtual hierarchies and select **Open Editor**. The editor tab shows the following columns:
 - Virtual hierarchies the virtual hierarchies added to the virtual dimension.
 - Hierarchies the source hierarchies in the source cubes to which the virtual hierarchy is mapped.

Tip: If the virtual dimension was created from one source dimension only (not merged), only one source hierarchy column is shown.

- **2**. To manually merge source hierarchies into a new virtual hierarchy, follow these steps:
 - a. Click Add Virtual Hierarchy 💽 .
 - b. Click **Editor** for the source hierarchy column related to the new virtual dimension, then select a source hierarchy, and click **OK**.

Tip: If you cannot select a source hierarchy because it is already mapped to a different virtual hierarchy, you must first delete the source hierarchy from the other virtual hierarchy.

- c. Repeat the step b for the second blank source hierarchy.
- 3. You can also perform the following tasks from here:
 - To delete a source hierarchy from a virtual hierarchy, select the source

hierarchy, and click **Delete** 🔀 .

- To delete a virtual hierarchy from a virtual cube, select the virtual hierarchy, and click **Delete**.
- 4. To complete the definition of a virtual hierarchy, select the virtual hierarchy in the **Project Explorer** tree to display the **Properties** tab.

Viewing virtual levels

When you create a virtual cube, IBM Cognos Cube Designer adds levels from the source cubes to the virtual cube.

Source cubes containing identical levels in a hierarchy are merged as virtual levels. If levels in the source cubes are not identical, level names from the first source cube are used as the names of the virtual levels. If one source cube contains more hierarchy levels than the second source cube, the extra levels are added as the lowest levels of the virtual hierarchy. For examples of the merging process, see "Virtual cubes" on page 24.

Procedure

From the **Project Explorer** tree, double-click the virtual hierarchy for which you want to view virtual levels. The editor tab shows the following columns:

- Virtual levels the virtual levels added to the virtual hierarchy.
- Levels the source levels in the source cubes to which the virtual level is mapped.

Tip: If the virtual hierarchy was created from one source hierarchy only (not merged), only one source level column is shown.

Model virtual members

When you create a virtual cube, IBM Cognos Cube Designer adds members from the source cubes to the virtual cube.

For a virtual hierarchy that is merged from two conformed dimensions, all hierarchy members from the source cubes are available as virtual members. If the level key for each source member is identical, members are added to the virtual cube as merged virtual members. Any members that do not have matching level keys are added to the virtual cube as new virtual members. For examples of the merging process, see "Virtual cubes" on page 24.

If a virtual member is not merged correctly, or could not be automatically merged, you can manually merge two source members. You can also delete redundant virtual members.

When manually merging virtual members, if member names do not match, a new virtual member is created using this format: <source member 1?source member 2>.

For example, two source cubes contain a Time hierarchy. Source cube 1 contains one member: All. Source cube 2 contains one member: All_Time. The virtual member created is All?All_Time.

The following table lists the properties that you can set when modeling a virtual member.

Property	Description	
Name	The name that appears in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.	
Comment	A comment or description of the virtual member. Comments are not visible in the IBM Cognos studios.	
Merge Operator	The method used to aggregate virtual members in the source cubes. By default, the merge operator is set to the same method that is defined for the virtual cube.	
Precedence	The merge operator to use if a tuple contains virtual members with different merge operators. The merge operator with the highest precedence is used. If there are two or more merge operators with the same precedence, the merge operator for the first virtual member in the tuple is used. Default: 0	

Table 23. Properties of a virtual member

The following table lists the properties that you can set when working with a source member.

Table 24. Properties of a source member

Header	Header
Name	The name that appears in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.
Include	Controls whether the source member is included in the virtual cube. If the same member exists in two source cubes, and you exclude the member from both source cubes, the member is excluded from the virtual cube.
	If the member is excluded from one source cube only, the member is included in the virtual cube. Default: True

Defining a virtual member

Using IBM Cognos Cube Designer, you can model virtual members within a virtual cube.

Procedure

- 1. From the **Project Explorer** tree, right-click the virtual hierarchy for which you want to define virtual members.
- 2. Select the Members tab. The editor tab shows the following columns:
 - Virtual members the virtual members added to the virtual hierarchy.

• Members - the source members in the source cubes to which the virtual level is mapped.

Tip: If the virtual hierarchy was created from one source hierarchy only (not merged), only one source member column is shown.

- **3**. To manually merge source members into a new virtual member, follow these steps:
 - a. Click Add Virtual Member.
 - b. Click **Editor** for the source member column related to the new virtual member, then select a source member, and click **OK**.

Important: To see the list of source members in a hierarchy, the source cube must be deployed as data source to the content store and started.

Tip: If you cannot select a source member because it is already mapped to a different virtual member, you must first delete the source member from the other virtual member.

- c. Repeat the step b for the second blank source dimension.
- 4. You can also perform the following tasks from here:
 - To delete a source member from a virtual member, select the source member, and click **Delete**
 - To delete a virtual member from a virtual cube, select the virtual member, and click **Delete**.
- 5. To complete the definition of a virtual member, select the virtual member to display the **Properties** tab.

Model virtual measures

When you create a virtual cube, IBM Cognos Cube Designer adds measures from the source cubes to the virtual cube.

Measures with identical names in the source cubes are added to a virtual cube as merged virtual measures. Measures that do not have identical names or that exist in only one of the source cubes are added to a virtual cube as new virtual measures. For examples of the merging process, see "Virtual cubes" on page 24.

If a virtual measure is not merged correctly, or could not be automatically merged, you can manually merge two source measures. You can also delete redundant virtual measures.

When merging measures in a virtual cube, it is not possible to map a source measure to more than one virtual measure.

The following table lists the properties that you can set when modeling a virtual measure.

 Property
 Description

 Name
 The name shown in the IBM Cognos studios. If the project supports multiple locales, there can be versions of the name in all supported languages.

Table 25. Properties of a virtual measure

Property	Description		
Comment	A comment or description of the virtual dimension. Comments are not visible in the IBM Cognos studios.		
Visible	Controls whether the measure is visible in the published package. Non-visible measures are typically used to represent intermediate values. These members are not intended to be used for direct reporting. However a non-visible measure is always present in the published package because the measure might be needed by other objects in a virtual cube. Non-visible measures are not displayed in the metadata browser and are removed from the output of reports which contain references to them. For example, a report that references a non-visible measure does not include output from that object. Default: True		
Merge Operator	The method used to aggregate virtual measures in the source cubes. By default, the merge operator is set to the same method that is defined for the virtual cube, but you can override it.		
Precedence	The merge operator to use if a tuple contains virtual measures with different merge operators.The merge operator with the highest precedence is used. If there are two or more merge operators with the same precedence, the merge operator for the first virtual measure in the tuple is used.Default: 0		
Data Format	Set the default data properties for each type of data.		

Table 25. Properties of a virtual measure (continued)

Defining a virtual measure

Using IBM Cognos Cube Designer, you can define virtual measures within a virtual cube.

Procedure

1. From the **Project Explorer** tree, right-click the virtual measure dimension and select **Open Editor**.

The editor tab shows the following columns:

- Virtual measures the virtual measures added to the virtual dimension.
- Measures the source measures in the source cubes to which the virtual measure is mapped.
- **2**. To manually merge source measures into a new virtual measure, follow these steps:
 - a. Click Add Virtual Measure 💽 .
 - b. Click **Editor** for the source measure column related to the virtual measure, then select a source measure, and click **OK**.

Tip: If you cannot select a source measure because it is already mapped to a different virtual measure, you must first delete the source measure from the other virtual measure.

c. Repeat the step b for the second blank source measure.

- 3. You can also perform the following tasks from here:
 - To a source measure from a virtual measure, select the source measure, and click **Delete X**.
 - To delete a source measure dimension (including all measures) from a virtual cube, select the source measure dimension, and click **Delete**.
 - To delete a virtual measure from a virtual cube, select the virtual measure, and click **Delete**.
- 4. To complete the definition of a virtual measure dimension or a virtual measure, select the required object in the **Project Explorer** tree to display the **Properties** tab.

Chapter 10. Defining security

When you have a valid dynamic cube, you can define security for it. Security is used to control the metadata available to specific users or user groups in the IBM Cognos studios. For example, if a dynamic cube includes a Geography hierarchy with two members, Canada and Europe, you can secure all data for Europe so that it is only accessible to certain users.

To define security you must complete the following tasks:

- Decide which hierarchies to secure and define one or more security filters for them.
- Define one or more security views to apply security to a dynamic cube. A security view can include security filters for hierarchy security. It can also include a definition for measure security.
- Publish a dynamic cube to the content store.

After publishing a dynamic cube to the content store, you must complete the following tasks in IBM Cognos Administration:

- Assign users, groups, and roles to security views.
- Refresh the security settings of the dynamic cubes on the query service.

Tip: If you make further changes to security after publishing a dynamic cube, you must republish and restart the dynamic cube.

Security for virtual cubes

You define security in source cubes as required. Virtual cubes automatically inherit the security settings defined in source cubes to maintain consistent security rules.

Security filters

You secure members in a hierarchy by using a security filter. A security filter specifies whether you are granting or denying access to one or more members.

Every hierarchy in IBM Cognos Cube Designer contains a default security filter named **All Members Granted**. This option explicitly grants access to all hierarchy members.

Tip: If you do not want to apply the default security filter, you can delete it from a security view.

You can define further security filters as required. For each filter, you must specify the scope to indicate if you are explicitly granting or denying access to hierarchy members. You then use a dynamic query mode expression to specify the required hierarchy members to add to the security filter.

Note: In order to be valid, the expression must return a set of hierarchy members.

The following table describes the scope options that you can use when defining a security filter.

Table 26. Security filter scope options

Scope	Description
Grant Members	Explicitly grant access to specified hierarchy members. Report users can see only the specified hierarchy members and associated values.
Grant Members and Descendants	Explicitly grant access to hierarchy members and all their descendants. Report users can see only the specified hierarchy members and associated values.
Grant Members and Ancestors	Explicitly grant access to hierarchy members and all their ancestors. Report users can see only the specified hierarchy members and associated values.
Grant Members, Descendants and Ancestors	Explicitly grant access to hierarchy members together with all their descendants and ancestors. Report users can see only the specified hierarchy members and associated values.
Deny Members and Descendants	Explicitly deny access to hierarchy members and all their descendants. Report users cannot see the specified hierarchy members and associated values.

There are several points to consider when setting up a security filter:

• When you explicitly grant access to a hierarchy member, report users can see only that member and its associated values. Users are denied access to all other hierarchy members.

For example, the Geography hierarchy contains these members: All, Canada, and Europe. If you grant access to the All member only, users cannot see Canada or Europe.

- When you explicitly grant access to a hierarchy member by using the **Grant Members** option or the **Grant Members and Descendants** option, report users can also see the ancestor members, but not their values. The value of any such ancestor member is shown as ERR to differentiate it from a true Null value.
- When you explicitly deny access to a hierarchy member, access to all other members in the hierarchy is implicitly granted.
- When you explicitly deny access to a hierarchy member, access to all descendant members is also denied.

If the result of this option is an unbalanced or ragged hierarchy, padding members are used to balance the hierarchy.

- If a security filter is set up with a grant or deny scope option, but does not contain an expression, then no members are granted or denied.
- If a security filter contains references to a member cannot be resolved, the member reference is ignored.
- If an error occurs as a result of applying a security filter, when a user opens a package or runs a report, an error message is shown because access to the whole hierarchy is automatically denied.

Aggregated data in a secured dynamic cube

When you grant access to hierarchy members, it is possible that report users might inadvertently infer member values to which they are denied.

For example, suppose that you have a Geography hierarchy with these members and values: All (100), Canada (30), Europe (70). Using the **Grant Members and Ancestors** option, access is explicitly granted to Canada and its parent (All). Report

users can see All (100), and Canada (30). If report users are aware that Europe is the only one other hierarchy member, they can infer that its value is 70.

Defining a security filter

You use IBM Cognos Cube Designer to define a security filter for a hierarchy.

Procedure

- 1. Select the hierarchy for which you want to define a security filter from the **Project Explorer** tree.
- 2. Select the **Security** tab.
- 3. Click Add Security Filter 🔀.
- 4. Select the security filter, and then select the required option from the **Scope** drop-down list.
- 5. Click **Edit** to define an expression to add members to the security filter.
- 6. Select the members to include in the filter by dragging and dropping them from the **Members** folder in the **Project Explorer** tree or by manually specifying them.
- 7. Click Validate to check that the expression does not contain errors.
- 8. Click OK.

What to do next

To apply a security filter to a dynamic cube, you must now add the filter to a security view.

Security views

You apply security to a dynamic cube by defining a security view. You can secure measures and hierarchy members in a dynamic cube.

To apply measure security, you grant or deny access to the required measures in a dynamic cube. To apply hierarchy member security, you add one more security filters to a security view.

There are several points to consider when setting up a security view:

- If a security view contains a security filter that explicitly denies access to a hierarchy member, it is not possible for another security filter to grant access to the same member.
- When you add multiple security filters to a security view, each filter is processed independently. If a security view does not include any security filters, users have access to all hierarchy members.

If a security view contains multiple security filters, the resulting list of granted members is derived from merging all granted members minus all denied members.

If there are no explicitly granted members, the merging of all granted members is replaced by the set of all implicitly granted members.

Report users are granted access to an individual member only if that member is granted access in all individual security filters.

• When you merge security views by using IBM Cognos Administration, the resulting list of granted members is derived from merging all granted members minus all denied members.

If there are no explicitly granted members, the merging of all granted members is replaced by the set of all implicitly granted members.

Report users are granted access to an individual member only if that member is granted access in all individual security views.

- When a security view includes security filters containing both grant and deny expressions, the resulting list of granted members is derived from merging all granted members minus all denied members.
- If a report user is not assigned to any security view where security is defined, they are denied access to all hierarchy members.

After defining a security view for a dynamic cube, you can publish the dynamic cube to the content store. To allow users to see metadata in a dynamic cube, you must now add users and groups to security views in IBM Cognos Administration. For more information, see "Assign users and groups to security views" on page 97.

Defining a security view

You use IBM Cognos Cube Designer to define a security view for a dynamic cube. A security view can include security filters and secured measures.

Procedure

- 1. From the **Project Explorer** tree, right-click the required dynamic cube and select **Open Editor**.
- 2. Select the **Security** tab.
- 3. Click Add Security View .
- 4. To add a security filter, click **Add Secured Data**, select the filters to add, then click **OK**.
- 5. To add secured measures to the security view, click **Add Secured Measure**, select the measures to add, then click **OK**.
- 6. Select Grant or Deny as required for each measure listed.

Chapter 11. Cognos Dynamic Cubes administration

You create dynamic cubes in IBM Cognos Framework Manager and publish them as data sources in Content Manager. Dynamic cubes are listed as data sources in IBM Cognos Administration.

Administrators perform specific tasks after dynamic cubes are published in Content Manager. After dynamic cubes are used and analyzed, there are additional tasks that administrators might need to perform.

Information about each dynamic cube data source can be viewed on its properties page.

There are several roles required for modeling, configuring, managing, and optimizing IBM Cognos Dynamic Cubes. Administrators create the roles and then assign capabilities and permissions.

After you model a dynamic cube, you can deploy it as an OLAP data source to Content Manager. To work with a deployed cube in the IBM Cognos studios, you must also publish a Framework Manager package, configure the cube as a data source, and start the cube. There are three different tools that allow you to deploy dynamic cubes:

- Cognos Cube Designer. For more information, see "Deploying and publishing dynamic cubes" on page 54.
- Framework Manager. For more information, see *IBM Cognos Framework Manager User Guide*.
- IBM Cognos Administration. For more information, see *IBM Cognos Business Intelligence Administration and Security Guide*.

Administration workflow

After the dynamic cubes are published as data sources, you perform the following tasks:

- Create and assign a Cognos account to access the relational database that contains the data for the dynamic cubes.
- Assign users and groups to security views.
- Add one or more dynamic cubes to the query service and configure their properties.
- Define routing rules to ensure that reports are directed to the dynamic query server. This task is necessary if you use multiple dispatchers on the query service.
- Edit the query service configuration parameters. For example, you might need to edit the JVM heap size.
- Manage dynamic cubes in the query service, and create and schedule query service tasks.
- Start the dynamic cubes on the query service.

After the dynamic cubes are used in reports and log files are analyzed, you might perform the following tasks:

• Use Aggregate Advisor to view aggregate recommendations.

- Monitor the metrics of the dynamic cubes added to the query service. For information about system performance metrics, see the *IBM Cognos Business Intelligence Administration and Security Guide*.
- Edit dynamic cube configuration in the query service to improve performance.

Roles and required capabilities

There are several roles required for modeling, configuring, managing, and optimizing IBM Cognos Dynamic Cubes. Administrators create the roles and then assign capabilities and permissions.

Administrators must ensure that these roles are available. To administer users, groups, and roles, you must have execute permissions for the **Users**, **Groups**, **and Roles** secured feature, and traverse permissions for the **Administration** secured function. For more information about roles and capabilities, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

The roles required to work with IBM Cognos Dynamic Cubes are described in the following table.

Role	Tasks	
Model cubes	Model cubes in IBM Cognos Cube Designer and publish cube as a data source.	
Secure cubes	Assign users and groups to security views in a dynamic cube data source.	
Configure cubes	Assign a cube to a dispatcher and edit cube properties. Assign a data access account. Assign a server group to a dispatcher and packages, and define routing rules.	
Optimize cubes	Run aggregate advisor and publish in-memory recommendations.	
Manage cubes	Perform operations on cubes such as start, stop, or refresh data cache, and create and schedule a query service task.	
Prime cubes	The data access account requires the ability to run scheduled jobs.	

Table 27. Roles used for IBM Cognos Dynamic Query tasks

Each role requires a specific administration capability. The following table lists the administration capability required for each role, including the access permissions required.

Table 28. Capabilities and permissions according to required roles

Role	Administration Capability	Access Permission Required
Model cubes	Data Source Connections	Read, Write, Traverse, Execute
Secure cubes	Data Source Connections	Read, Write, Traverse
Configure cubes	Configure and manage the system	Read, Write, Traverse
Optimize cubes	Data Source Connections	Read, Write, Traverse
Manage cubes	Administration Tasks	Read, Traverse, Execute
Manage cubes	Query Service Administration	Read, Write, Traverse, Execute
Manage cubes	Configure and manage the system	Read, Write, Traverse
Prime cubes	None	Read, Traverse, Execute

General Properties for dynamic cube data sources

You can view the properties of dynamic cube data sources in IBM Cognos Administration.

You access the General properties from the **General** tab of the **Set properties** page. You can view and edit properties for dynamic cube data sources. The following table lists and describes the properties on the General tab. For information about working with data sources, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

Property	Description		
Туре	The type of property. For example, a Dynamic Cubes database, a Dispatcher or a Namespace are all a type of property.		
Owner	The owner of the entry. By default, the owner is the person who created the entry. When the owner no longer exists in the namespace, or is from a different namespace than the current user, the owner shows as Unknown .		
	If you have Set policy permissions, then you can click Make me the owner to become the owner of the entry.		
Contact	The person responsible for the entry. Click Set the contact and then click Select the contact to set the contact for the entry or click Enter an email address to enter the contact email address.		
Location	The location of the entry in the portal and its ID. Click View the search path , ID and URL to view the fully qualified location and the ID of the entry in the content store.		
	Entries are assigned a unique identification (ID) number.		
Created	The date the entry was created.		
Modified	The most recent date that the entry was modified.		
Icon	The icon for the entry. Click Edit to specify a different icon.		
Disable this entry	When selected, users that do not have write permission for this entry cannot access it. The entry is no longer visible in the portal.		
	If an entry is disabled and you have write permission to it, the disabled icon is displayed next to the entry.		

Table 29. Dynamic cube data source properties

Property	Description
Hide this entry	Select this property to hide reports, packages, pages, folders, jobs, and other entries. Hide an entry to prevent it from unnecessary use, or to organize your view. The hidden entry is still accessible to other entries. For example, a hidden report is accessible as a drill-through target.
	A hidden entry remains visible, but its icon is faded. If you clear
	the Show hidden entries check box in my area options . , My Preferences , the entry disappears from your view.
	You must have access to the Hide Entries capability granted by your administrator to see this property.
Language	A list of languages that are available for the entry name, screen tip, and description according to the configuration that was set up by your administrator.
Name	The name of the entry for the selected language. Note: Renaming a dynamic query cube can cause several problems for any objects that reference this cube. For this reason, you should not change the name of the dynamic cube data source.
Screen tip	An optional description of the entry. The screen tip displays when you pause your pointer over the icon for the entry in the portal. Up to 100 characters can be used for a screen tip.
Description	An optional description of the entry, which displays in the portal when you set your preferences to use the details view.
	Details view is displayed only in Public Folders and My Folders.
Access Account	The access account is used by the dynamic query mode server to access the relational database. The dynamic query mode server uses the data source sign-on credentials to access the relational database that contains the data warehouse of a dynamic cube. Given that, you can select which Cognos account to use based on its credentials. You must create the credentials before you define the access account.
	To define the Cognos account, click Select the access account , and then select the user to own the account. For more information, see "Access account property for dynamic cubes."

Table 29. Dynamic cube data source properties (continued)

Access account property for dynamic cubes

Use the access account property of a dynamic cube data source to assign a Cognos account. The data source credentials from this account are used by the dynamic query mode server to access the relational database. The relational database contains the data warehouse of a source dynamic cube.

The access account property is on the **General** tab of the **Set properties** page for a dynamic cube data source. The Cognos account that you assign must have access to the relational database that contains the source dynamic cube.

Before you set the access account

Before you set the access account, you need to create credentials for the user that will access the relational database. Next, you create a data source signon for that user. This user must have access to the relational database that contains the source dynamic cube. Also, the user ID and password that make up the signon must already be defined in the relational database.

To create credentials, see "Create trusted credentials" on page 96.

To create a signon, see "Create a signon" on page 96.

Access account properties for virtual cubes

Virtual cubes obtain data from other source or virtual cubes. For this reason, a virtual cube does not have an access account property.

However, if a virtual cube has a startup trigger, it needs an access account. In this situation, the virtual cube uses the access account of the first cube in the cube definition.

If a virtual cube is built using two virtual cubes, the access account that is used is the one that belongs to the first source cube of the first virtual cube.

Multiple dynamic cube connections and signons

You can use multiple data source connections or multiple data source signons for any dynamic cube data source.

If you do use multiple connections or signons, then the dynamic query mode server searches for a connection or signon called **DynamicCubes**. When the **DynamicCubes** signon is located, it is used and the dynamic cube can start. For this reason, you must ensure that one of your multiple connections and one of your multiple signons is named **DynamicCubes**.

If the dynamic cube data source uses a single associated data source signon then there are no requirements for a signon name called **DynamicCubes**. For information about creating data source connections and data source signons, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

Tasks and permissions

This table lists tasks that can be performed by the access account and the corresponding permissions that are required.

Table 30. Access account tasks

Task	Permission requirement
Load data and metadata from the relational data source that the cube is based on.	Permission to read a stored signon for the relational data source.
Execute startup triggers.	Capability to run activities and schedules.

Create trusted credentials

You can create trusted credentials when you want to authorize other users to use your credentials when those users do not have sufficient access permissions to perform specific tasks.

Before you begin

For users to use trusted credentials, traverse permissions must be granted for the namespace.

Procedure

- In IBM Cognos Connection, in the upper-right corner, click the my area options button , My Preferences.
- 2. On the **Personal** tab, under **Credentials**, if you have not created credentials before, click **Create the Credentials**.
- **3**. Select the users, groups, or roles you want to authorize to use your credentials. If you are prompted for your credentials, provide your user ID and password.
- 4. If you want to add entries, click Add then choose how to select entries:
- 5. If you want to remove an entry from the list, select the check box next to it and click **Remove**.
- 6. Ensure that the list contains only the users, groups, or roles that you want, and click **OK**.

Create a signon

You can create a signon.

Procedure

- 1. In IBM Cognos Connection, in the upper-right corner, click Launch, IBM Cognos Administration.
- 2. On the Configuration tab, click Data Source Connections.
- **3**. Click the data source, and then click the connection to which you want to add a new signon.
- 4. Click the new signon button
- 5. In the name and description page, type a unique name for the data source signon and, if you want, a description and screen tip, and then click **Next**.
- 6. Type the **User ID** and **Password** to connect to the database, and click **Next**. The **Select the users** page appears.
- 7. To add users and groups that can use the signon, and click Add.
 - To choose from listed entries, click the appropriate namespace, and then select the check boxes next to the users, groups, or roles.
 - To search for entries, click **Search** and in the **Search string** box, type the phrase you want to search for. For search options, click **Edit**. Find and click the entry you want.
 - To type the name of entries you want to add, click **Type** and type the names of groups, roles, or users using the following format, where a semicolon (;) separates each entry:

namespace/group_name;namespace/role_name;namespace/user_name; Here is an example: Cognos/Authors;LDAP/scarter;

8. Click the right-arrow button and when the entries you want appear in the **Selected entries** box, click **OK**.

Tip: To remove entries from the **Selected entries** list, select them and click **Remove**. To select all entries in a list, click the check box in the upper-left corner of the list. To make the user entries visible, click **Show users in the list**.

9. Click Finish.

The new data source signon appears under the connection.

Setting the data access account

The query service uses this account to log on to IBM Cognos Business Intelligence. This account provides access to relational data source credentials and loads data for the dynamic cube.

Before you begin

You need to create credentials for the user that will access the relational database before you set the access account.

If your dynamic cube data source used multiple connections or signons then one of the connections and one of the signons must be defined using the name **DynamicCubes**.

Procedure

- 1. Launch IBM Cognos Administration.
- 2. On the **Configuration** tab, select **Data Source Connections**.
- **3**. Click the set properties button for the dynamic cube data source that you want to modify.
- 4. Click Select the access account.
- 5. Browse the directory and select the user that will own the access account.
- 6. Click **OK**. The user name appears in the **Access Account** section on the **General properties** page.
- 7. Click OK.

Assign users and groups to security views

Administrators assign users and groups to the security views that were defined in IBM Cognos Cube Designer. Security views can be accessed from the model within a dynamic cube data source. The model contains the security views that were defined for the dynamic cube. A model view in IBM Cognos Administration is equivalent to a security view in Cognos Cube Designer.

By default, when a dynamic cube is published to the content store, everyone has access permissions to the model view. Administrators must override the access permissions to remove the Everyone group from the security view, and then add the appropriate users and groups to the model view.

Assigning users, groups, and roles to the model view allows access to the metadata in a dynamic cube.

A model is found in the dynamic cube data source. To view all dynamic cube data sources, go to the **Configuration** tab and select **Data Source Connections**.

For information about assigning permissions to a dynamic cube model view, see "Set access permissions for an entry"

Set access permissions for an entry

Setting access permissions for an entry includes creating new permissions or updating existing permissions. You can specify access permissions for all entries in IBM Cognos software. Some examples of such entries are reports, queries, analyses, packages, agents, metrics, namespaces, groups, users, or dispatchers. You can reference users, group and roles from different namespaces in a security policy for an entry.

If you plan to reference entries from multiple namespaces, log on to each namespace before you start setting access permissions. Otherwise, entries in namespaces to which you are not logged on are shown as **Unavailable**.

Procedure

- 1. In IBM Cognos software, locate the entry for which you want to set access permissions.
- 2. In the Actions column, click the set properties button
- 3. In the **Set properties** page, click the **Permissions** tab.
- 4. Choose whether to use the permissions of the parent entry or specify permissions specifically for the entry:
 - To use the permissions of the parent entry, clear the **Override the access permissions acquired from the parent entry** check box, then click OK if you are prompted to use the parent permissions. Click **OK**.

for the entry.

- To set access permissions for the entry, select the **Override the access permissions acquired from the parent entry** check box, then proceed to step 5.
- 5. If you want to remove an entry from the list, select its check box and click **Remove**.

Tip: If you want to select all entries, select the check box at the top of the list in the upper-left corner. Clear the check box to deselect all entries.

- 6. To specify the entries for which you want to grant or deny access, click Add, then choose how to select entries:
 - To choose from listed entries, click the appropriate namespace, and then select the check boxes next to the users, groups, or roles.
 - To search for entries, click **Search** and in the **Search string** box, type the phrase you want to search for. For search options, click **Edit**. Find and click the entry you want.
 - To type the name of entries you want to add, click **Type** and type the names of groups, roles, or users using the following format, where a semicolon (;) separates each entry:

namespace/group_name;namespace/role_name;namespace/user_name;

Here is an example:

Cognos/Authors;LDAP/scarter;

7. Click the right-arrow button and when the entries you want appear in the **Selected entries** box, click **OK**.

Tip: To remove entries from the **Selected entries** list, select them and click **Remove**. To select all entries in a list, click the check box in the upper-left corner of the list. To make the user entries visible, click **Show users in the list**.

- **8**. For each entry in the list, in the box next to the list, select or clear check boxes to specify what type of access you want to grant or deny.
- 9. Click **OK**.

In the **Permissions** column, an icon appears next to the user, group, or role. This icon represents the type of access granted or denied to the entry.

10. If you want to remove access permissions that were previously set for the child entries so that the child entries can acquire permissions set for this entry, in the **Option** section, select the **Delete the access permissions of all child entries** check box.

This option appears only with entries that are containers. You can use it to restrict access to a hierarchy of entries.

Warning: Select this option only when you are certain that changing access permissions of the child entries is safe.

11. Click OK.

Configure a dynamic cube for the query service

You can configure one or more instances of the query service to support a version of a dynamic cube. The query service supports the dynamic query mode. It manages dynamic query requests and returns the results to the requesting batch or report service.

You can add a dynamic cube to an instance of the query service and then configure its properties. For more information, see "Adding dynamic cubes to the query service" on page 105.

If you plan to use multiple dispatchers for the query service then you will also need to define routing rules to ensure that reports are directed to the dynamic query server for execution. To ensure that your server processes dynamic cube requests, you need to:

- assign a server group to the dispatcher
- assign a routing set to all packages associated with a dynamic cube
- create a routing rule to send queries for the routing set to the server group

You can set routing rules in IBM Cognos Administration or you can use the SDK. For more information about creating routing rules in IBM Cognos Administration, see the *IBM Cognos Business Intelligence Administration and Security Guide*. For information about creating custom routing rules using the SDK, see *IBM Cognos Software Development Kit Developer Guide*.

After you add dynamic cubes to the query service, you can monitor the state and metrics for each cube, and view the settings for each cube. For information about system performance metrics, see the *IBM Cognos Business Intelligence Administration and Security Guide*. You can also manage dynamic cubes on the query service. For more information, see "Managing dynamic cubes in the query service" on page 106.

Administrators can create and schedule query service tasks for dynamic cube data sources. For more information, see "Create and schedule query service administration tasks" on page 108.

Configuring query service properties

The query service uses environment, logging, and tuning configuration settings.

About this task

A summary of the query service settings can be viewed on the **Settings - Query Service** pane. You can configure properties for the query service.

Procedure

- 1. Launch IBM Cognos Administration, and from the Status tab select System.
- 2. On the **Scorecard** section, click the **All servers** list, point to **Services**, and then click **Query**.
- 3. From the **QueryService** list, click **Set properties**.
- 4. Click the **Settings** tab.
- 5. In the **Value** column, type or select the value for the settings that you want to change. The following table describes the properties that you can set for the query service.

Option	Description
Advanced Settings	Click Edit to specify advanced configuration settings. Because an entry acquires advanced settings from a parent, editing these settings overrides the acquired advanced settings. For information about types of advanced settings, see the <i>IBM Cognos Administration</i> <i>and Security Guide</i> .
Dynamic cube configurations	Click Edit to add dynamic cubes to the query service. For information, see "Adding dynamic cubes to the query service" on page 105.
Audit logging level for query service	Select the level of logging that you want to use for the query service.
Enable query execution trace	A query execution trace (run tree trace) shows queries that run against a data source. You use the trace to troubleshoot query-related issues.
	You can find execution trace logs in the following location: <i>c10_location</i> /logs/XQE/ <i>reportName</i> /runtreeLog.xml.
	For more information about the query execution trace, see the <i>IBM Cognos Dynamic Query Guide</i> .
	You can view and analyze these log files using IBM Cognos Dynamic Query Analyzer. For more information, see the <i>IBM</i> <i>Cognos Dynamic Query Analyzer User Guide</i> .
Option	Description
--	--
Enable query planning trace	Query plan tracing (plan tree) captures the transformation process of a query. You can use this information to gain an advanced understanding of the decisions and rules that are executed to produce an execution tree.
	The query planning trace is logged for every query that runs using dynamic query mode. You can find planning trace logs in the following location: <i>c10_location</i> /logs/XQE/ <i>reportName</i> /plantreeLog.xml.
	Since planning logs are large, there is an impact on query performance when this setting is enabled.
	For more information about the query planning trace, see the <i>IBM Cognos Dynamic Query Guide</i> .
Disable query plan caching	Specifies whether the service caches query plans for possible reuse. Caching the query plan takes additional resources and is possibly not suitable for your environment. For example, if your environment does not have much CPU power and the nature of the user requests are often different, caching the query plan does not produce significant performance improvements.
Generate comments in native SQL	Specifies which reports are generating the SQL queries in the database.
Write model to file	Specifies whether the query service will write the model to a file when a query runs. The file is used only for troubleshooting purposes. Modify this property only with the guidance of IBM Software Support. You can find the file in the following location: c10_location\logs\model\ packageName.txt
Additional JVM arguments for the query service	Defines additional parameters to be passed to the Java virtual machine instance for the dynamic query mode. Modify this property only with the guidance of IBM Software Support.
Idle connection timeout	Specifies the number of seconds to maintain an idle data source connection for reuse.
	The default setting is 300. Valid entries are 0 to 65535.
	Lower settings reduce the number of connections at the expense of performance. Higher settings might improve performance but raise the number of connections to the data source.

Option	Description
Do not start dynamic cubes when service starts	Prevents the dynamic cubes from starting when the query service starts.
Dynamic cube administration command timeout	Specify the amount of time to wait for a resource to be available for a dynamic cubes administration action. This action is canceled if the time period is exceeded. Note: Setting this value to zero causes the command to wait indefinitely.
Minimum query execution time before a result set is considered for caching	Specify the minimum amount of time to wait for a query before caching the results. This setting does not apply to dynamic cubes.
Initial JVM heap size for the query service	Specifies the initial size, in MB, of the Java Virtual Machine (JVM) heap.
JVM heap size limit for the query service	Specifies the maximum size, in MB, of the JVM heap.
Initial JVM nursery size	Specifies the size, in MB, that the JVM allocates to new objects. The nursery size is automatically calculated. This setting does not need to be changed.
	The only situation in which this setting needs to change is if you are in consultation with IBM Cognos customer support and they recommend a specialized change.

Types of caches used by dynamic cubes

Several types of caches are available for dynamic cubes to allow improvements to query response times.

Result set cache

Result set cache is an intermediate storage of multidimensional expression language (MDX) query results. This cache is stored on disk in a binary format. The in-memory portion of the result set cache stores the queries and the associated security profile. If an MDX query from the dynamic query mode server to the IBM Cognos Dynamic Cubes engine matches an entry in the result set cache then the result is read from the disk instead of executing the query.

Expression cache

The MDX engine caches the results of various intermediate MDX set expressions that are keyed by the expression, its query context, and the security profile of the user. If the MDX engine encounters a set expression that was previously executed then it will retrieve the result set from the expression cache instead of calculating the set expression.

The expression cache helps to relieve the costs associated with the time and memory it takes to execute set expressions.

Data cache

The MDX engine sends data queries to the Cognos Dynamic Cubes engine. The result of each query that is retrieved from the database (fact table), database aggregate tables, and in-memory aggregate cache is stored in a data cache.

Before sending any query to the database, the Cognos Dynamic Cubes engine scans the data cache for all the entries which are able to provide some or all of the required data without querying the database.

The data cache is also known as the query cache.

Aggregate cache

Aggregate Advisor analyzes dynamic cubes and suggests aggregates that can improve cube performance. The aggregate cache contains pre-calculated values for aggregations that are suggested by Aggregate Advisor. The pre-calculated values are results of queries to the database.

Aggregate tables

Data can be summarized in a table known as an aggregate table. An aggregate table contains detail fact data that is aggregated at a higher level relative to one or more of the dimensions associated with the data. Using an aggregate table allows use of pre-calculated data from a data warehouse and decreases the amount of data that is accessed from the data warehouse.

Dynamic cube configuration settings

You can set dynamic cube parameters when you add a dynamic cube to an instance of the query service. The default settings are most often the best choice, with the exception being the data cache size limit.

The following table lists and describes the parameters that you can set when you add a dynamic cube to the query service. For more information, see "Adding dynamic cubes to the query service" on page 105.

Parameter name	Value description
Disabled	Disables the cube. This means that the cube is configured for a server, but is not running on that server.
Startup trigger name	Type the name of the trigger event to send after this cube starts.
	When a cube is available for query processing, the event is triggered for execution against the server which triggered the event. The purpose of the event is to run reports to populate the cube cache with data.
Disable result set cache	Disables the result cache.
	Disabling the cache is useful during the development or testing phase of a cube because it allows you to test the performance of the data cache.

Parameter name	Value description
Data cache size limit	Type the maximum size of the data cache for the cubes.
	The default value is 1024 MB. The result of each query is written out to disk. If the maximum size is exceeded, the older report sets are removed from the cache.
Maximum amount of disk space to use for	Type the maximum size of disk space.
result set cache	The result of each query is written to disk. If the maximum amount of disk space is exceeded, the older report sets are removed from the cache.
Enable workload logging	Enables workload logging, which is used to capture information about queries that are sent to the dynamic query engine processes. This workload information is used by Aggregate Advisor to determine aggregate recommendations.
	You need to restart the cube after you enable workload logging.
Maximum amount of memory to use for aggregate cache	Type the maximum size of the memory to use for the in-memory aggregates. In-memory aggregates are loaded when cubes are started and restarted, and when data cache is refreshed. The size of aggregate cache is also a component in determining the total JVM heap size of the query service.
	In-memory aggregates are loaded on a first come, first served basis. This means that if the aggregate cache is full, then no more in-memory aggregates can be loaded. Additionally, an in-memory aggregate may fail to load if the in-memory aggregate cache size limit would be exceeded if it did load. The default value is 0, which species not to use in-memory aggregates even if they are
Disable external aggregates	defined.
Disable external aggregates	is useful during the cube and application development phase to measure the impact of external aggregates.
	To measure the impact of external aggregates you need to gather output twice. First, you gather output when external aggregates are enabled and then you gather output again when external aggregates are disabled. You use these two sets of output to determine the impact of external aggregates.

Parameter name	Value description
Percentage of members in a level referenced in a filter predicate	If no limit is required, type 0.
referenceu în a finer preuleate	This value must be between 0 and 100.
	This parameter applies to retrieving data associated with a set of members. If there is a greater percentage retrieved than what you specify in this field, then the SQL query generated retrieves measure values for all of the members at the level (a speculative pre-fetch of data).

Adding dynamic cubes to the query service

You can add and configure dynamic cubes to use the query service.

Before you begin

If you have virtual cubes and source cubes that are part of the same hierarchy then these cubes must be added to the same query service dispatcher. For more information about hierarchies, see "Hierarchies" on page 11.

Procedure

- 1. Launch IBM Cognos Administration, and from the Status tab select System.
- 2. On the **Scorecard** section, click the **All servers** list, point to **Services**, and then click **Query**.
- 3. From the **QueryService** list, click **Set properties**.
- 4. Click the **Settings** tab.
- 5. Click Edit to set the Dynamic cube configurations.
- 6. Click **New configuration**, select the cubes that you want to add to the service, and then click the right-arrow button.
- 7. When the entries you want appear in the Selected entries box, click OK.
- **8**. Optionally, click the edit configuration icon to edit the properties for each dynamic cube. For a description of the property, pause your pointer over the property name .
- 9. Click OK in each window until you are back to the Set properties page.
- **10**. Optionally, edit any of the Environment, Logging, or Tuning properties. For information, see "Configuring query service properties" on page 100.
- 11. Click **OK**. When the page refreshes, the new dynamic cubes appear in the QueryService list. The settings that you defined appear in the **Settings** -

Query Service pane. **Tip:** Click the refresh button to manually refresh the pane.

Deleting a dynamic cube from the query service

You can delete a dynamic cube from an instance of the query service.

Procedure

- 1. Launch IBM Cognos Administration, and from the Status tab select System.
- 2. On the **Scorecard** section, click the **All servers** list, point to **Services**, and then click **Query**.

- 3. From the QueryService list, click Set properties.
- 4. Click the **Settings** tab.
- 5. Click Edit to set the Dynamic cube configurations.
- 6. Select the cubes that you want to delete, and then click Delete.
- 7. Click **OK** in each window until you are back to the **Scorecard** section. When the page refreshes, the available dynamic cubes appear in the QueryService list.

What to do next

If you created and scheduled any tasks for the query service for this dynamic cube, then you must delete the task, otherwise your scheduled tasks will point to nonexistent cubes.

Enable workload log for Aggregate Advisor

Aggregate Advisor can analyze the underlying model in a dynamic cube data source and recommend which aggregates to create. Aggregate Advisor runs on the query service and can reference a workload log file. If you want Aggregate Advisor to consider information from workload logs when making recommendations, then the workload log file must be enabled on the dynamic cube.

When enabled, the workload log file captures the information that represents user workload usage such as running reports and IBM Cognos studio usage. This log file allows Aggregate Advisor to suggest aggregates (in-database or in-memory) that correspond directly to the reports contained in the log file.

For information about enabling the workload log file, see "Dynamic cube configuration settings" on page 103.

For more information about using Aggregate Advisor, see *IBM Cognos Dynamic Query Analyzer User Guide*.

Clear workload log

Clearing the workload log removes all entries for a dynamic cube.

If you want to capture a different set of report usage, then you can clear the workload log to remove existing entries.

You can create and schedule query service tasks for clearing the workload ("Create and schedule query service administration tasks" on page 108) or you can clear the workload manually ("Managing dynamic cubes in the query service").

Managing dynamic cubes in the query service

The query service runs and creates a copy of a dynamic cube. Administrators can start, stop, refresh, and diagnose these copies.

Before you begin

Since virtual cubes are composed of source cubes, there are several things to consider before you start, stop, and refresh cubes:

• Source cubes that are a part of a virtual cube must be started first.

- If source cubes are part of a virtual cube, then the virtual cube must be stopped before the source cubes are stopped.
- When you refresh the data and member cache of a source cube, the data and member cache of any associated virtual cubes is also refreshed.
- Virtual cubes only have a subset of operations that you can manage; **Start**, **Stop after active tasks complete**, and **View recent messages**.

About this task

When you start a cube, the hierarchy members are reloaded and the most recent data is loaded in the cache.

Typically, you stop a cube if it does not need to be online and accessible. For example, when ETL operations are performed on the underlying data warehouse.

When you want to keep cubes accessible to users during updates you can use refresh operations. Member and data caches are refreshed dynamically while queries are still running. An update of the member cache builds a new set of members in the background. This new set becomes available when the refresh is complete. This refresh requires additional memory to store two copies of the member cache in memory while the new cache is built. When the member cache is brought online, a new corresponding data cache is also created.

Even though a new data cache starts as empty, some additional space is required while the new cache is introduced and queries are using the previous version of the data cache.

The query service is available on the **System** page of the **Status** tab in IBM Cognos Administration. Use the following procedure to start, stop, restart, refresh member cache, or refresh data cache for dynamic cubes on the query service. You can view recent log messages to diagnose problems with dynamic cubes. The time zone displayed is the time zone of the Administrator that is viewing the log messages.

Procedure

- 1. Launch IBM Cognos Administration.
- 2. On the **Status** tab, select **System**.
- 3. From the All servers list, point to Services and then select Query.
- 4. Click **QueryService**. The cubes that were added to the QueryService display. It can take several minutes for the cubes to display.
- 5. Click the list next to the dynamic cube that you want to manage, and then select one of the following operations:
 - Start
 - Stop after active tasks complete
 - Stop immediately
 - Restart
 - Refresh member cache
 - Refresh data cache
 - Refresh security settings
 - Clear workload log
 - View recent messages

The **View the results** window displays to report the status of the action. For additional status information, see the **Status** column of the Scorecard section.

Create and schedule query service administration tasks

Administrators can create and schedule query service tasks for data sources. Query service tasks control one or more cubes by clearing, writing, or refreshing its cache. For dynamic cubes, you can also schedule when cubes start, stop, or restart, and refresh security.

You can

- schedule cache clearing and clear the cache to control memory usage by a specific data source or cube
- schedule the generation of a time-stamped report (write cache state)

You can also clear the entire cache manually and write the cache state to a report manually.

You can create query service administration tasks and run them on demand. You can run them at a scheduled time or based on a trigger, such as a database refresh or an email . You can schedule them as part of a job . You can also view the run history of query service administration tasks .

Important: If you schedule **Clear Cache** tasks to run at predetermined times, you must disable automatic cache clearing. For more information, see .

Before you begin

When you create and schedule tasks for dynamic cubes, you need to schedule start and stop tasks for source cubes and virtual cubes separately. There are other factors to consider when scheduling start and stop tasks for dynamic cubes:

- Source cubes that are a part of a virtual cube must be scheduled to start first.
- If source cubes are part of a virtual cube, then the virtual cube must be scheduled to stop before the source cubes.
- You need to provide enough time for source cubes to start before scheduling a virtual cube to start. The same consideration must be made when you schedule virtual and source cubes to stop.

Procedure

- 1. Launch, IBM Cognos Administration.
- 2. On the **Configuration** tab, click **Content Administration**.
- **3**. Click the New Query service administration task button $\overset{[V]}{V}$ in the upper-right corner.
- 4. Specify a name, description, screen tip, and location. Click Next.
- 5. Select an operation, either Clear Cache or Write Cache State.
- 6. For Oracle Essbase and SAP BW data sources, enter the data source, catalog, and cube. Click **Next**.

For relational data sources such as IBM DB2, Microsoft SQL Server, IBM Netezza, Oracle, and Teradata, you only need to specify the data source. Enter an asterisk (*) as a wildcard to specify all.

- 7. For Dimensionally-Modeled Relational (DMR) data sources, enter either the name of a package name or the name of a data source. If you specify a data source name and chose the **Clear Cache** operation, the cache is cleared for all packages that involve that data source.
- 8. For dynamic cube tasks, select the **Server Group**, **Dispatcher**, and **Cubes**, and then click **Next**.
- 9. Choose the action that you want:
 - To run the task now or later, click **Save and run once** and click **Finish**. Specify a time and date for the run, and then click **Run**. Review the run time and click **OK**.
 - To schedule the task at a recurring time, click **Save and schedule** and click **Finish**. Then, select frequency and start and end dates. Click **OK**.

Tip: To temporarily disable the schedule, select the **Disable the schedule** check box.

• To save the task without scheduling or running, click **Save only** and click **Finish**.

What to do next

You must remember to delete a scheduled task if you delete the associated cube from the query service. Otherwise, your scheduled tasks will point to nonexistent cubes.

Appendix A. Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

The major accessibility features for IBM Cognos Cube Designer are described in the following list. You can

- customize the display to enhance accessibility. For example, you can enable a focus ring that emphasizes the selected element.
- use shortcut keys to navigate and trigger actions.
- apply operating system display settings, such as high-contrast display.

For more information about the commitment that IBM has to accessibility, see the IBM Accessibility Center (http://www.ibm.com/able).

Accessibility features in Cognos Cube Designer

You can customize the IBM Cognos Cube Designer display to enhance accessibility.

The View menu includes the following display controls.

Table 31. View menu options

View menu item	Description
View Access Keys	Adds a numeric identifier to each pane. To navigate to a different pane, press Alt+Shift+ <i>pane number</i> . The navigation control works when View Access Keys is disabled.
View Focus Ring	Displays a dotted rectangle around the object that has the current keyboard focus.

Keyboard shortcuts for Cognos Cube Designer

You can use keyboard shortcuts to navigate through and perform some tasks in IBM Cognos Cube Designer.

Applies to	Description	Keyboard shortcut
General	Perform the default action for an active command button.	Enter or Spacebar
General controls	Move forward to the next control at the same level.	Tab
General controls	Move backward to the previous control at the same level.	Shift+Tab

Table 32. Keyboard shortcuts for Cognos Cube Designer

Applies to	Description	Keyboard shortcut
Check boxes	Toggle a check box from selected to cleared or cleared to selected.	Spacebar
	Tip: This shortcut also applies to other settings that can have an on or off state.	
Radio buttons that are not in a group	Move to the next radio button and select it.	Tab
Radio button groups	Move to the next radio button in the group and select it.	Right arrow Down arrow
Radio button groups	Move to the previous radio button in the group and select it.	Up arrow Left arrow
Drop-down lists	Open and display the drop-down list contents.	Alt+Down arrow
Drop-down lists	Close an open drop-down list.	Alt+Up arrow
Tree controls	Move to the first selectable node below, or, if the node below has child nodes and the node is expanded, move to the first child node.	Down arrow
Tree controls	Move to the first selectable node above.	Up arrow
Tree controls	Expand the selected node or move to the first selectable child node.	Right arrow
Tree controls	Collapse the selected node, move to the parent node, or move to the first selectable node above.	Left arrow
Tree controls	Move to the first node in a tree control.	Home
Tree controls	Move to the last node in a tree control.	End
Menus	Move to the next available menu item.	Down arrow

Table 32. Keyboard shortcuts for Cognos Cube Designer (continued)

Applies to	Description	Keyboard shortcut
Menus	Move to the previous available menu item.	Up arrow
Menus	Expand the child menu items.	Right arrow
Menus	Collapse the child menu items.	Left arrow
Context menus	Open the context menu for the selected item.	Shift+F10
Context menus	Close an open context menu.	Esc
Scrolling	Scroll down.	Down arrow Page down
Scrolling	Scroll up.	Up arrow Page up
Columns	Change the width.	Ctrl+Shift+► Ctrl+Shift+◀

Table 32. Keyboard shortcuts for Cognos Cube Designer (continued)

Appendix B. Report considerations

Calculated members in reports

For most reports, IBM Cognos Dynamic Cubes calculated members are used the same way as regular members. However, because of some different constraints and capabilities, the report user may encounter unexpected results. In these cases, you must consider the required type and behavior of the members to obtain the desired output. In reporting environments, calculated members appear to be identical to regular members. It is a good practice to use a naming convention so that report users can easily identify calculated members.

The values of calculated members and measures are not retained within a dynamic cube. The values are computed at every occurrence within reports and analyses when executed.

You create Cognos Dynamic Cubes calculated members manually. Cognos Dynamic Cubes relative time calculated members are specialized calculated members automatically added to a relative time hierarchy and cannot be modified.

Calculated members that you manually create have the following characteristics:

- Each occurrence of a single calculated member in a report or analysis is considered unique. (SET operations, Filtering calculated members)
- They do not have siblings or children.
- They should not be nested.
- Their rank value in IBM Cognos Analysis Studio is always Null.

Relative time calculated members

The relative time feature generates three types of calculated members.

The Period to Date Change and Period to Date Growth relative time calculated members share the following characteristics with Cognos Dynamic Cubes calculated members.

- Each occurrence of a single calculated member in a report or analysis is considered unique. (SET operations, Filtering calculated members)
- They do not have siblings or children.
- They should not be nested
- Their rank value in IBM Cognos Analysis Studio is always Null.

The Current Period, Prior Period, Current Period to Date and Prior Period to Date members may have children. Therefore, the functions CHILDREN, DESCENDANT, FIRSTCHILD and LASTCHILD can return results. These relative time calculated members share the following characteristics with Cognos Dynamic Cubes calculated members:

- Each occurrence of a single calculated member in a report or analysis is considered unique. (SET operations, Filtering calculated members)
- They should not be nested
- Their rank value in IBM Cognos Analysis Studio is always Null.

Reference relative time members refer to other members within the time hierarchy and have the same caption and member key values as the members to which they refer. Within the context of other reference members, these members behave the same as Cognos Dynamic Cubes calculated members. Unlike Cognos Dynamic Cubes calculated members, these members are not considered unique, they can have children and they can be nested. Reference members at the same level are siblings of other reference members. When applied to a reference member, functions such as FIRSTSIBLING or NEXTMEMBER will return a reference member. Their rank value in IBM Cognos Analysis Studio is always Null.

SET operations

Because a calculated member is considered to be unique from all other calculated members, the UNION, EXCEPT, UNIQUE, and INTERSECT functions may give results that appear incorrect.

In the following examples, [USA] and [Canada] are regular members and [CM1] and [CM2] are calculated members.

Example	Result Set
UNION	SET
(SET([USA], [CM1], [CM2]),	([USA], [CM1], [CM2], [Canada], [CM1])
SET([USA], [Canada], [CM1])	The member [CM1] appears twice in the result.
EXCEPT	SET
(SET([USA], [CM1], [CM2]),	([CM1], [CM2], [Canada], [CM1])
SET([USA], [Canada], [CM1])	The member [USA] is removed, but the member [CM1] appears twice in the result.
UNIQUE	SET ([USA], [CM1], [CM1], [Canada])
(SET([USA], [CM1], [USA], [CM1], [Canada])	The member [CM1] appears twice in the result.
INTERSECT	SET([USA])
(SET([USA], [CM1], [CM2]),	Calculated members do not appear in the intersection of two sets
SET([USA], [Canada], [CM1])	increation of two sets.

Table 33. Examples of SET operations with calculated members

Filtering calculated members

Since calculated members are considered to be unique from all other calculated members, a filter will not remove the members.

If a report contains a filter based on IBM Cognos Dynamic Cubes calculated members and the same hierarchy is visible in the report, the data values in the report will be correct. However, the filter will not remove visible members from the report. If the same hierarchy is not visible in the report, the report output will be as expected.

Nesting calculated members

IBM Cognos Dynamic Cubes calculated members should not be nested. Because all calculated members are considered unique, the dynamic query mode query planner resolves the intersection to an empty set. The rows remain in the reports but the values are Null.

Calculated members siblings and children

IBM Cognos Dynamic Cubes calculated members do not have siblings or children. Functions that require a member sibling or child as a result will always be Null.

- NEXTMEMBER([CM1]) = NULL
- PREVMEMBER([CM2]) = NULL
- LEAD([CM1], 0) = NULL
- LAG(([CM2], 0) = NULL

Cognos Analysis Studio rank

In IBM Cognos Analysis Studio, the rank of an IBM Cognos Dynamic Cubes calculated member is always Null. The context in which the rank is computed and the context used to compute the values visible in the cross tab are not the same. Because, the computed rank values could contradict the visible values, the rank is always set to Null.

Relative time calculated members in reports

IBM Cognos Dynamic Cubes relative time members are specialized calculated members that are added to a time hierarchy.

The relative time feature generates three types of calculated members

Period to Date Change, Period to Date Growth

These relative time calculated members share the following characteristics with Cognos Dynamic Cubes calculated members.

- They are considered unique.
- They do not have siblings or children
- They should not be nested
- Their rank value in IBM Cognos Analysis Studio is always null.

Current Period, Prior Period, Current Period to Date, Prior Period to Date

These members behave the same as Cognos Dynamic Cubes calculated members with one exception. These members may have children. Therefore, the functions CHILDREN, DESCENDANT, FIRSTCHILD and LASTCHILD can return results.

These relative time calculated members share the following characteristics with Cognos Dynamic Cubes calculated members:

- They are considered unique.
- They should not be nested
- Their rank value in IBM Cognos Analysis Studio is always null.

Reference relative time members

These members refer to other members within the time hierarchy and have the same caption and member key values as the members to which they refer. Within the context of other reference members, these members behave the same as Cognos Dynamic Cubes calculated members with one exception. These members can have children. Reference members at the same level are siblings of other reference members. When applied to a reference member, functions such as FIRSTSIBLING or NEXTMEMBER will return a reference member.

These relative time calculated members share the following characteristics with Cognos Dynamic Cubes calculated members:

- They are considered unique.
- They can be nested
- Their rank value in IBM Cognos Analysis Studio is always null.

Removal of padding members from reports

The use of padding members can result in skewed calculations related to the members of a hierarchy level. If a level contains padding members, they are included in the count of members. In addition, because padding members can have associated fact data values, this can skew the value of aggregates computed on a level-basis.

For example, in a State/City hierarchy, if the state of California has no city level members, a padding member at the city level is created as a child of California to balance the hierarchy. If the Sales measure value for California is 100, then the child padding member also has a value of 100. The number of city entries across all states is now inflated by 1 and that the sum of all Sales values across all cities is inflated by 100.

To remove skewed data from a report, you can define a filter for a set of members based on a dynamic cube.

Hierarchies with padding members are not displayed as ragged or unbalanced in the IBM Cognos studios. A report user can identify ragged and unbalanced hierarchies by looking for members with a blank caption or the same caption as their parent. These members have a NULL business key because they do not represent real members. Filtering members with a NULL business key removes all the padding members. A report filter such as FILTER(MEMBERS([My Level]), [My Level].[My Level - Key] = NULL) removes padding members from the report.

Appendix C. Troubleshooting

This section provides solutions for problems you may encounter when using IBM Cognos Dynamic Cubes.

BMT-MD-6527 error when generating a dynamic cube based on a relational table

You cannot generate a dynamic cube based on a relational table if the underlying data does not use referential integrity.

You will see the following error:

BMT-MD-6527 The table is inappropriate for use in a measure dimension; it has no foreign keys.

To model a dynamic cube based on the selected relational table, you must define it manually. For more information, see "Defining a dynamic cube manually" on page 51 and Chapter 1, "Cognos Dynamic Cubes overview," on page 1.

Possible overflow in measure attributes

Measure attributes a dynamic cube may be too small to hold aggregate values of the measures.

The Measure properties of **Data type**, **Precision** and **Scale** are inherited from the relational database metadata and cannot be modified. If the aggregate value of a measure exceeds the size of the attribute, you see an error indicating that an overflow has occurred. For example, a Quantity Measure defined as Int(4) overflows when summed in a dynamic cube.

To avoid overflow errors, first evaluate the database columns you want to use as measures. If the resulting data type will not accommodate the aggregation value of the measure, do the following:

- Create a default measure for the database column you want to use as a measure.
- Evaluate the measure to determine an appropriate aggregate size.
- Hide the original measure that you determined to cause an overflow.
- Create a new measure.
- Define the measure using the expression property. The expression must be an explicit cast of the original measure into a larger data type.

The syntax for the CAST function is CAST (<expression>, <datatype>)

For example: CAST([MyDataItem], varchar(10))

If casting to a data type that accepts size, precision, or scale, those parameters appear in parentheses after the data type. For example CAST([MyDataItem], decimal(10,2))

In-memory aggregates fail to load

If in-memory aggregates fail to load when a dynamic cube starts, additional memory may be required for the aggregate cache.

In-memory aggregates are defined by running Aggregate Advisor in IBM Cognos Dynamic Query Analyzer and saving the in-memory aggregate definitions. When a dynamic cube is restarted, the in-memory aggregates are loaded. If they fail to load, check the dynamic cube error log for the following message:

"Loading of in-memory aggregates was skipped because the value for the ''Maximum amount of memory to use for aggregate cache'' property is zero. To enable loading in-memory aggregates, update the property to a value greater than zero to be the amount of memory to allocate for the aggregate cache."

In IBM Cognos Administration, open the properties for the cube and set the **Maximum amount of memory to use for aggregate cache** to a value greater than or equal to the one used when creating the recommendations in Aggregate Advisor.

Notices

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 1623-14, Shimotsuruma, Yamato-shi Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group Attention: Licensing 3755 Riverside Dr Ottawa, ON K1V 1B7 Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, 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 "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

- Netezza is a registered trademark or trademark of Netezza Corporation, an IBM Company.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.



Index

A

access account 95 access permissions setting 98 accessibility features 111 keyboard shortcuts 111 administration 91 Aggregate Advisor 106 in-database 29 in-memory 29 aggregate awareness 27, 71 aggregate cubes defining 73, 74 defining automatically 72 filtering data for 75 in-database aggregates 27 in-memory aggregates 27 modeling 71 overview 27 parent-child dimension 74 properties 71 simple 72, 73 aggregate slicer 75 aggregation rules calculated 22 custom 22 defining 53 attributes level key 16, 18 level unique key 18 member caption 18 member description 18 overview 18 relational mapping 19 support multiple locales 68

В

balanced hierarchies 12 browsing members in a hierarchy 47 business keys *See* level keys

С

calculated measures 57, 115 calculated members 57, 65, 115, 117 defining 62 relative time 115, 117 unique 115 Cognos Cube Designer overview 31 user interface 31 Cognos Dynamic Cubes overview 1 workflow 5 configuring dynamic cubes 54 Content Manager data source importing metadata 33 current data 27 current period expression examples 64 current period examples expressions 64 current period expressions examples 64

D

data members hiding 15 showing 15 data source access account 95 defining dimensions 37 level unique keys 43 security filters 89 security views 90 sort order of members 44 virtual dimensions 79 denying security 87, 89 deploying dynamic cubes 54 design language 68 dimensional metadata attributes 18 dimensions 11 hierarchies 11 joins 18 levels 16 modeling 37 overview 11 parent-child hierarchies 15 dimensions defining 37 degenerate 11 modeling 37 parent-child 11 regular 11 relational mapping 19 relative time 63, 64 disabling content maintenance job 109 dynamic cubes 91 advanced modeling 57 based on a relational table 50 configuration 106 configuring 54 defining manually 49 deploying 54 editing 34 errors 35 generating from a relational table 49 importing metadata for 33 issues 35 measure dimension 19 measures 21

dynamic cubes *(continued)* modeling 49 opening 34 overview 19 publishing 54 saving 34 security 87 starting 54 validating 35 warnings 35

Ε

editing dynamic cubes 34 projects 34 errors 35 examples 24 month rolling window 65 calculated members 58 level current period expressions 64 relative time 65 execution trace query service 100 expresssion editor defining calculated members 62 extraneous padding members 15

F

filtering data aggregate cubes 75 Framework Manager publishing multilingual packages 69

G

granting security 87, 89

Η

hiding data members 15 measures 52, 57 padding members 15 hierarchies balanced 12 browsing members 47 defining 40 modeling 39 multiple 12 overview 11 padding members 14 ragged 13 relational mapping 19 security 87, 89 sort order of members 44 unbalanced 12

idle connection timeout query service 100 importing Content Manager data source metadata 33 metadata 33
in-database 29
in-memory 29
in-memory aggregates fail to load error 120
issues 35

J

```
joins
measure-to-dimension 49, 54
overview 18
relational mapping 20
```

K

keyboard shortcuts 111

L

languages adding to metadata objects 68 design language 68 publishing multilingual packages 69 support multiple locales 68 supporting different locales 68 level keys 16, 43 attribute 18 level unique keys 16 attribute 18 defining 43 levels alternative modeling 16 best practice modeling 16 modeling 41 overview 16 relational mapping 19 locales adding languages to metadata objects 68 adding to members and attributes 68 design language 68 supporting 68

Μ

measure dimension 19 measure-to-dimension joins defining 54 measures aggregation rules 22 calculated 57, 115 calculated measures 21 dynamic calculated measures 21 hidden 52, 57 modeling 51, 52, 53 overview 21 regular aggregation 22 security 89 visible 52, 57 member caption attribute 18 member description attribute 18

members browsing in hierarchies 47 calculated 57, 115, 117 defining sort order 44 predefined relative time 64 support multiple locales 68 metadata adding languages 68 importing 33 importing from a Content Manager data source 33 relational 50 modeling advanced 57 aggregate cubes 71 dimensional metadata 37 dimensions 37 dynamic cubes 49 hierarchies 39 levels 41 measures 51 parent-child hierarchies 44 relative time dimensions 63 virtual cubes 77 virtual dimensions 78 virtual hierarchies 79 virtual measures 83 virtual members 81 multiple hierarchies 12

0

objects errors 35 issues 35 validating 35 warnings 35 opening dynamic cubes 34 projects 34 overview aggregate cubes 27 Cognos Cube Designer 31 Cognos Dynamic Cubes 1 dimensional metadata 11 dimensions 11 dynamic cubes 19 hierarchies 11 parent-child hierarchies 15 user interface 31 virtual cubes 24

Ρ

packages publishing multilingual 69 padding members 14 hiding 15 removing from reports 118 showing 15 parent member 57 parent-child hierarchies data members 15 defining 46 modeling 44 overview 15 pre-cached historical data 27 projects adding support for locales 68 defining virtual cubes 77 design language 68 editing 34 errors 35 issues 35 opening 34 saving 34 validating 35 warnings 35 properties query service 100 publishing dynamic cubes 54

Q

query planning trace query service 100 query service create and schedule administration tasks 108 execution trace 100 idle connection timeout 100 properties 100 query planning trace 100 run tree trace 100 write model to file 100

R

ragged hierarchies 13 rational tables using for dynamic cubes 50 regular aggregation 22 relative time calculated members 115, 117 relative time dimensions defining 64 modeling 63 run tree trace query service 100

S

saving dynamic cubes 34 projects 34 scenarios virtual cube 27 schedules query service administration tasks 108 schemas snowflake 19 star 19 security denying access 87, 89 dynamic cubes 87 granting access 87, 89 hierarchies 87, 89 measures 89 virtual cubes 87 security filters 87 defining 89 security views 87, 89 defining 90

setting access permissions 98 shared dimensions 27 showing data members 15 padding members 15 snowflake schemas 19 sort order defining for members 44 star schemas 19 starting dynamic cubes 54

T

triggers 95 troubleshooting in-memory aggregates 120

U

unbalanced hierarchies 12 user interface 31

V

viewing members in a hierarchy 47 virtual cubes current data 27 virtual cubes (continued) defining in a project 77 defining virtual dimensions 79 modeling 77 objects 24 overview 24 pre-cached historical data 27 scenarios 27 security 87 shared dimensions 27 virtual dimensions 24 defining 79 modeling 78 virtual hierarchies modeling 79 virtual measures 24 modeling 83 virtual members 24 modeling 81 visible 57

W

warnings 35 workflow Cognos Dynamic Cubes 5 workload logs 106 write model to file dynamic query service parameters 100 query service 100