
PeopleTools 8.58: Cube Builder

May 2020

PeopleTools 8.58: Cube Builder
Copyright © 1988, 2020, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

- Preface: Preface..... v**
 - Understanding the PeopleSoft Online Help and PeopleBooks..... v
 - Hosted PeopleSoft Online Help..... v
 - Locally Installed Help..... v
 - Downloadable PeopleBook PDF Files..... v
 - Common Help Documentation..... v
 - Field and Control Definitions..... vi
 - Typographical Conventions..... vi
 - ISO Country and Currency Codes..... vi
 - Region and Industry Identifiers..... vii
 - Translations and Embedded Help..... vii
 - Using and Managing the PeopleSoft Online Help..... viii
 - Understanding PeopleSoft Cube Builder..... viii
 - PeopleTools Related Links..... viii
 - Contact Us..... viii
 - Follow Us..... ix
- Chapter 1: Getting Started with PeopleSoft Cube Builder..... 11**
 - PeopleSoft Cube Builder Overview..... 11
 - Essbase Cube Builder Business Processes..... 11
 - Other Sources of Information..... 11
 - Essbase Cube Builder Implementation..... 12
 - Designing Cube Metadata..... 12
 - Defining Connection Definitions..... 12
 - Creating Dimensions..... 12
 - Creating an Outline..... 13
 - Building Cubes..... 13
- Chapter 2: Designing Cube Metadata..... 15**
 - Understanding Cubes..... 15
 - Cube Overview..... 15
 - Hierarchy Overview..... 16
 - Understanding PeopleSoft Metadata..... 18
 - Understanding Metadata Types..... 19
 - Understanding Oracle Essbase Properties..... 19
 - Understanding PeopleSoft Process Scheduler Integration..... 20
 - Using Trees..... 20
 - Creating Queries..... 21
 - Using Dimension Queries..... 21
 - Using Data Source Queries..... 22
- Chapter 3: Defining Essbase Cube Builder Connection Definitions..... 31**
 - Defining Essbase Cube Builder Connections..... 31
- Chapter 4: Creating Dimensions for Essbase Cubes..... 33**
 - Understanding Dimensions of Essbase Cubes..... 33
 - Setting Up Cube Dimension Builder..... 33
 - Creating Essbase Dimensions..... 35
 - Defining an Essbase Dimension..... 35
 - Inserting a Tree Structure into the Dimension..... 54

| | |
|--|------------|
| Inserting a Query into the Dimension..... | 59 |
| Defining Time Dimensions..... | 67 |
| Adding Generation Names..... | 71 |
| Chapter 5: Defining Essbase Cube Outlines..... | 73 |
| Defining Cube Outlines..... | 73 |
| Adding Cube Dimensions to a Cube Outline..... | 73 |
| Adding Cube Data Queries to a Cube Outline..... | 80 |
| Chapter 6: Creating Dynamic Dimension Templates..... | 87 |
| Understanding Dynamic Dimension Templates..... | 87 |
| Creating Dynamic Dimension Templates..... | 87 |
| Chapter 7: Building Essbase Cubes..... | 91 |
| Building Essbase Cubes..... | 91 |
| Integrating with Oracle Smart View..... | 94 |
| Using API and Command-line Specifications..... | 95 |
| Chapter 8: Securing Cubes..... | 99 |
| Securing Cube Builder Cubes..... | 99 |
| Controlling User Operations..... | 99 |
| Setting User Actions..... | 101 |
| Viewing Content References..... | 102 |
| Chapter 9: Maintaining Essbase Cubes..... | 105 |
| Maintaining Essbase Cubes..... | 105 |
| Deleting Dimensions..... | 105 |
| Deleting Outlines..... | 106 |
| Deleting Run Control IDs..... | 106 |
| Viewing Dynamic Objects..... | 107 |
| Viewing Outline Structures..... | 108 |
| Expanding Dimensions..... | 109 |
| Appendix A: Viewing Query Results That Have Large Result Sets..... | 111 |
| Query Chunking and the Chunking Class..... | 111 |
| Dynamic Query Expansion..... | 113 |
| Appendix B: Using Essbase Cube Builder Demo Data..... | 115 |
| Understanding Essbase Cube Builder Demo Data..... | 115 |
| Applying Project CUBE_DEMO_PROJ..... | 115 |
| Granting Permissions to All Menu CUBE_TEST Components..... | 115 |
| Importing PS_CUBEBUILDER.DAT File Using Data Mover Script PS_CUBEBUILDER.DMS..... | 116 |
| Granting Query Security..... | 116 |
| Adding Data to Tables..... | 117 |
| Using Sample Dimensions and Outlines..... | 117 |

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

Hosted PeopleSoft Online Help

You can access the hosted PeopleSoft Online Help on the [Oracle Help Center](#). The hosted PeopleSoft Online Help is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support. The hosted PeopleSoft Online Help is available in English only.

To configure the context-sensitive help for your PeopleSoft applications to use the Oracle Help Center, see [Configuring Context-Sensitive Help Using the Hosted Online Help Website](#).

Locally Installed Help

If you're setting up an on-premise PeopleSoft environment, and your organization has firewall restrictions that prevent you from using the hosted PeopleSoft Online Help, you can install the online help locally. See [Configuring Context-Sensitive Help Using a Locally Installed Online Help Website](#).

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format on the [Oracle Help Center](#). The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The two main types of common help are:

- Application Fundamentals
- Using PeopleSoft Applications

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft applications.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

| Typographical Convention | Description |
|---------------------------------|--|
| Key+Key | Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For Alt+W, hold down the Alt key while you press the W key. |
| ... (ellipses) | Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax. |
| { } (curly braces) | Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (). |
| [] (square brackets) | Indicate optional items in PeopleCode syntax. |
| & (ampersand) | When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables. |
| => | This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character. |

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation

does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region and Industry Identifiers

Information that applies only to a specific region or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in the PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in the PeopleSoft Online Help:

- USF (U.S. Federal)
- E&G (Education and Government)

Translations and Embedded Help

PeopleSoft 9.2 software applications include translated embedded help. With the 9.2 release, PeopleSoft aligns with the other Oracle applications by focusing our translation efforts on embedded help. We are not planning to translate our traditional online help and PeopleBooks documentation. Instead we offer very direct translated help at crucial spots within our application through our embedded help widgets. Additionally, we have a one-to-one mapping of application and help translations, meaning that the software and embedded help translation footprint is identical—something we were never able to accomplish in the past.

Using and Managing the PeopleSoft Online Help

Select About This Help in the left navigation panel on any page in the PeopleSoft Online Help to see information on the following topics:

- Using the PeopleSoft Online Help
- Managing Hosted online help
- Managing locally installed PeopleSoft Online Help

Understanding PeopleSoft Cube Builder

PeopleSoft Cube Builder is a set of PeopleTools pages and processes that you use to create and maintain Essbase applications and databases. It also provides mechanisms to interconnect PeopleTools with Oracle Performance Manager tools such as Smart View.

Note: The concepts and terminology used in PeopleSoft Cube Builder are the same concepts used in Oracle Performance Manager, which is the suite that contains Essbase and Smart View. For more information, see the documentation for Oracle Hyperion products.

PeopleTools Related Links

[PeopleTools 8.58 Home Page](#)

[PeopleTools Elasticsearch Home Page](#)

"PeopleTools Product/Feature PeopleBook Index" (PeopleTools 8.58: Getting Started with PeopleTools)

[PeopleSoft Hosted Online Help](#)

[PeopleSoft Information Portal](#)

[PeopleSoft Spotlight Series](#)

[PeopleSoft Training and Certification | Oracle University](#)

[My Oracle Support](#)

[Oracle Help Center](#)

Contact Us

Send your suggestions to psft-infodev_us@oracle.com. Please include the applications update image or PeopleTools release that you're using.

Follow Us



[Facebook.](#)



[YouTube](#)



[Twitter@PeopleSoft_Info.](#)



[PeopleSoft Blogs](#)



[LinkedIn](#)

Chapter 1

Getting Started with PeopleSoft Cube Builder

PeopleSoft Cube Builder Overview

PeopleSoft Cube Builder is an extract-transform and load (ETL) application that enables you to use PeopleSoft Query and PeopleSoft Tree Manager to build Essbase applications and online analytical processing (OLAP) databases. It comprises an easy to use graphic user interface to create metadata of cubes based in PeopleSoft structures and to give a look and feel similar to the Essbase Outline Editor. It also uses an enhanced cube builder program that provides fast performance to load data and metadata, as well as interconnections with the Oracle Performance Manager - Smart View.

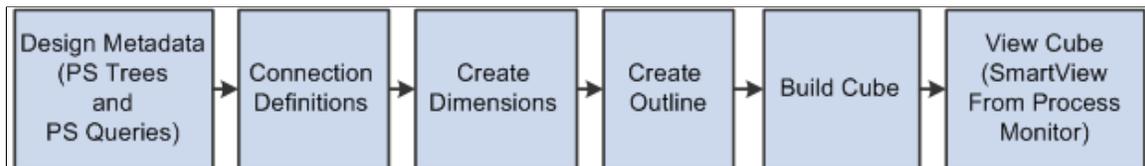
Note: The concepts and terminology used in PeopleSoft Cube Builder are the same concepts used in Oracle Performance Manager, which is the suite that contains Essbase and Smart View. For more information, see the documentation for Oracle Hyperion products.

To view outputs from PeopleSoft Pure Internet Architecture, you must have Smart View for Office installed on the client machine. Please refer to the Oracle Smart View for Office Installation Guide for details.

Essbase Cube Builder Business Processes

Image: Essbase Cube Builder business process flow

The following process flow illustrates the Cube Builder business processes.



Note: Oracle Essbase, SmartView, and other Oracle Hyperion tools are not delivered as part of our application; therefore, this PeopleSoft Cube Builder product documentation does not discuss how to use Oracle Hyperion applications. It does, however, include a reference to the appropriate documentation.

Other Sources of Information

This topic provides information to consider before you begin to use PeopleSoft Cube Builder.

Take advantage of all PeopleSoft sources of information, including the installation guides, release notes, product documentations, red papers, the Updates and Fixes area of My Oracle Support, and curriculum courses of Oracle PeopleSoft.

See [PeopleSoft Cube Builder Overview](#).

Essbase Cube Builder Implementation

Essbase Cube Builder implementation can be divided into different phases. This topic provides an overview of the phases involved in Essbase cube builder implementation.

Designing Cube Metadata

The initial phase for creating cubes is to define their metadata, that is, the data that describes the cube structure. You can define metadata with either PeopleSoft trees or queries. Perform the steps in the following table to design cube metadata.

| Step | Reference |
|---|--|
| Create new trees or use existing trees. | See "PeopleSoft Tree Manager Overview" (PeopleTools 8.58: Tree Manager), "Understanding Steps to Create Trees" (PeopleTools 8.58: Tree Manager). |
| Create new queries or use existing queries. | See "Creating New Queries" (PeopleTools 8.58: Query). |

Defining Connection Definitions

The second phase of creating Essbase cubes is to create connectivity information. Perform the step listed in the following table to set up the Essbase server connection information.

| Step | Reference |
|---|---|
| Define Essbase Cube Builder connection definitions. | See Defining Essbase Cube Builder Connections . |

Creating Dimensions

Dimensions are equivalent to fields in a relational database. In data analysis terms, dimensions can be thought of as criteria that are used to pinpoint a particular piece of data, for example, time, account, and salesperson. Dimensions are subdivided into categories called members; for example, the time dimension contains the members 2007, 2008, 2009. You can add members to dimensions manually; members are populated by the results of a PeopleSoft Query or populated from a PeopleSoft Tree structure.

Perform the steps listed in the following table to create dimensions.

| Step | Reference |
|--------------------------------|---|
| Set up cube dimension builder. | See Understanding Dimensions of Essbase Cubes . |
| Create Essbase dimensions. | See Creating Essbase Dimensions . |

Creating an Outline

To create an outline, you group dimensions together to define the cube structure, and then you define the data queries that will be used to populate the cube.

Perform the steps listed in the following table to create an outline.

| Step | Reference |
|--|--|
| Define cube outlines. | See Defining Cube Outlines . |
| Add cube dimensions to a cube outline. | See Adding Cube Dimensions to a Cube Outline . |
| Add cube data queries to a cube outline. | See Adding Cube Data Queries to a Cube Outline . |

Building Cubes

The last phase of Essbase Cube Builder implementation is sending all of the defined information—cube structure and data—to the Essbase server using the Create Cube component, which runs the Process Scheduler.

Perform the steps listed in the following table to build Essbase cubes.

| Step | Reference |
|----------------------|--|
| Build Essbase cubes. | See Building Essbase Cubes . |

Chapter 2

Designing Cube Metadata

Understanding Cubes

Cube Overview

The key concept of online analytical processing (OLAP) is that of a cube. In this document, we use the term *cube* to refer to any analytic data store. An OLAP cube is a collection of related data—a database—that has multiple dimensions. The term *cube dimensions* roughly describes the equivalent of fields in a relational database. In terms of data analysis, dimensions can be thought of as criteria—such as time, account, and salesperson—that can pinpoint a particular piece of data. These pieces of data are usually transactions from an online transaction processing (OLTP) system.

Although they are called cubes, OLAP databases can have more than three dimensions. In fact, most cubes have three to eight dimensions. To understand the concept of OLAP cubes, start with a simple data analysis model and then expand it.

Suppose you want to analyze unit sales of your company. You can examine the total units that were sold in a particular year, but that number might not help you understand much about your business. Instead, you might want to see unit sales broken down by time and by product. The matrix that you use to analyze this data might look like the following table, which represents a cube with two dimensions (time and product).

| Product | 2001 | 2002 | 2003 |
|----------------|-------------|-------------|-------------|
| Widgets | 3000 | 6500 | 8200 |
| Gadgets | 1200 | 1450 | 3000 |
| Doohickeys | 2500 | 3400 | 2000 |
| Whatzits | 500 | 670 | 1300 |

Dimensions and Members

In OLAP terminology, the preceding table is an OLAP cube that represents units sold dimensioned by time and product. Time and product are the dimensions of the cube, and units sold is the fact data.

In the preceding table, each dimension is subdivided into categories, called *cube members*, which represent individual years and products.

- In the time dimension, the members are 2001, 2002, and 2003.
- In the product dimension, the members are widgets, gadgets, doohickeys, and whatzits.

Measures and Cells

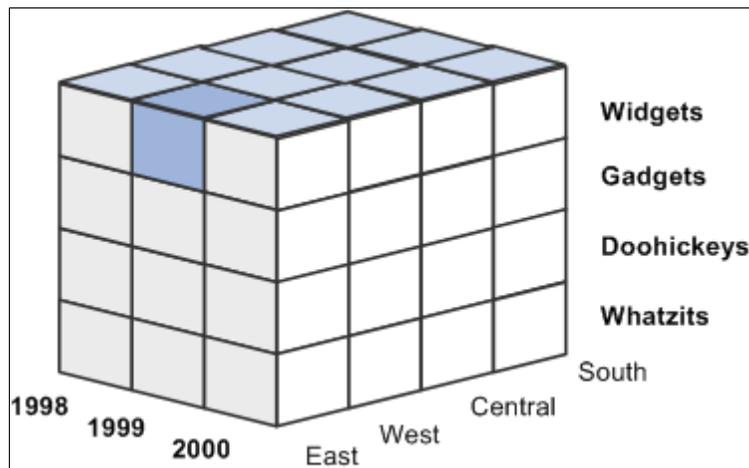
In the preceding table, the values of the most interest are not years or products. The purpose of the table is to find the number of units that were sold. Sold units make up the data element that is being evaluated or measured. In OLAP terminology, the number of units sold is called the *measure*, or *fact*, of this cube. The areas of the table where members intersect with other members represent individual measure and fact values. These intersections are called *cells*. The italicized cell in the preceding table represents the number of widgets sold in 2002: 6500 units.

Multiple Dimensions

The two-dimensional cube represented by the table in the Understanding Cubes section is basic for reporting purposes. For example, it does not provide data about where any of the units were sold. You can provide this information by adding another dimension, *location*, to the model, as shown in the following diagram.

Image: Diagram of a cube with three dimensions

This diagram illustrates a cube with three dimensions.



The preceding three-dimensional OLAP cube represents units sold dimensioned by time, product, and location. (The location members are East, West, Central, and South.) The shaded cell represents the number of widgets sold in the East region in 1999. You could find the number of units sold for any other product in any other region at any other time by finding the cell at the intersection point of three members, one from each dimension.

Suppose you also want to factor customer accounts into the analysis. Although showing four dimensions graphically is a challenge, the result of this added dimension is clear: in our example, each cell of the OLAP cube represents the intersection of an account, a year, a region, and a product.

Hierarchy Overview

Hierarchy is the organization of cube data elements with their reporting structures. It represents both the hierarchy and the method of consolidation in a dimension level.

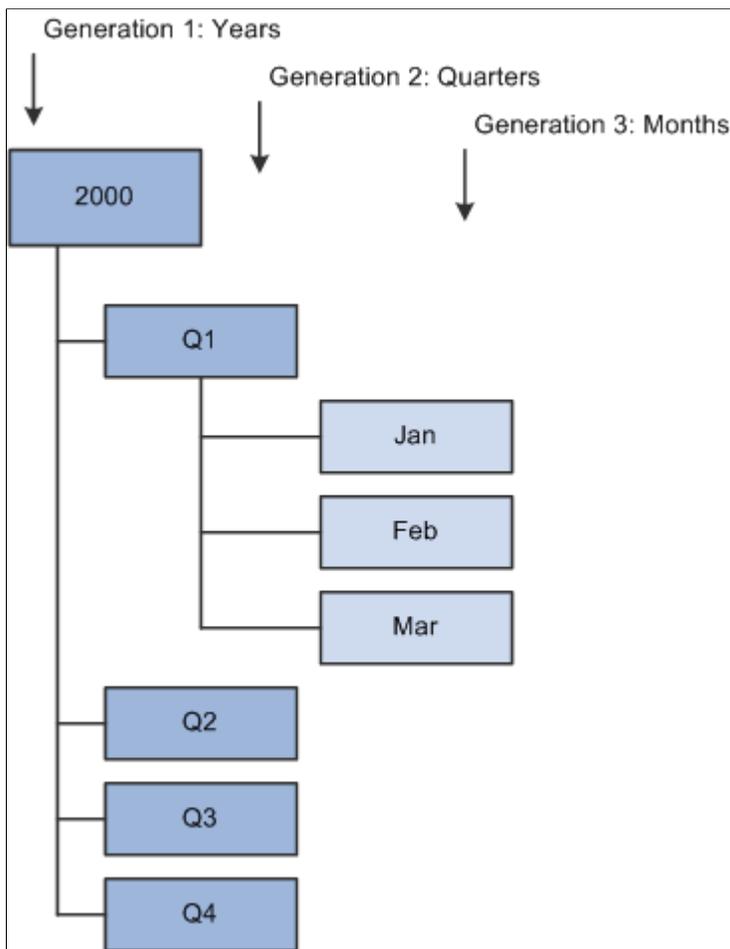
The example cube has only one level in each dimension. The time dimension consists of one level containing three members (years), and the location dimension consists of one level containing four members (regions). However, the data used to build such OLAP cubes probably supports more than just one level in each dimension.

For example, when a company records a sale, that sale occurs in a particular month, which occurs in a particular quarter and in a particular year. You can examine the time dimension at one of three levels: month, quarter, or year. Likewise, you can record that each sale occurs in a particular office, in a particular city, or in a particular region. The location dimension might also have three levels, such as office, city, and region.

As mentioned, the categories found at each level of a dimension are called members. You can envision multilevel dimensions as tree diagrams, the members of which relate to each other in various parent-child relationships. Some members are parents of other members, some are children, and some are both.

Image: Example of a time hierarchy diagram

This diagram shows an example of a portion of a typical time dimension with its various levels and members.



Each box in the diagram represents a unique member. This diagram is familiar to PeopleSoft Tree Manager users. In fact, PeopleSoft trees can play an important role in defining the hierarchy of an OLAP cube.

See "PeopleSoft Tree Manager Overview" (PeopleTools 8.58: Tree Manager).

Consolidation

Viewing the dimension of a hierarchy tells you about the organization of its members, but you should consider another facet of the dimension. You need to know how to consolidate the values that are found

under child members into the value of their parent members. For example, the children might be added together to equal the parent. This scenario is certainly the case in a time dimension, in which the value for each member is added to its siblings to equal the value of its parent. (Three months can be consolidated into their parent quarter, four quarters can be consolidated into their parent year, and so on.)

The following table shows the cube example, adding a second level, quarters, to the time dimension of the original example.

Image: Consolidation example in tabular format

This table shows the cube example, adding a second level, quarters, to the time dimension of the original example.

| Product | 2001 Q1 | 2001 Q2 | 2001 Q3 | 2001 Q4 | 2002 Q1 | 2002 Q2 | 2002 Q3 | 2002 Q4 | 2003 Q1 | 2003 Q2 | 2003 Q3 | 2003 Q4 |
|------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Widgets | 400 | 700 | 950 | 400 | 1200 | 980 | 995 | 975 | 2200 | 990 | 750 | 500 |
| Gadgets | 300 | 300 | 300 | 300 | 300 | 300 | 400 | 450 | 600 | 650 | 850 | 900 |
| Doohickeys | 600 | 600 | 700 | 600 | 750 | 850 | 950 | 850 | 750 | 400 | 300 | 550 |
| Whatzis | 125 | 125 | 125 | 125 | 125 | 200 | 225 | 120 | 170 | 230 | 400 | 500 |

To consolidate the data at the quarterly level into the yearly level, the quarterly data is added together. The 2001 rollup is Q1 2001 + Q2 2001 + Q3 2001 + Q4 2001.

However, you also might find dimensions in which certain members are to be subtracted from their siblings, such as in a profit dimension. In such a dimension, suppose two members are at the first level, margin and total expenses, both of which are reported as positive values. To find the total profits, you would not add margin and total expenses, instead you would subtract total expenses from margin.

Related Links

"PeopleSoft Tree Manager Overview" (PeopleTools 8.58: Tree Manager)

"Creating New Queries" (PeopleTools 8.58: Query)

Understanding PeopleSoft Metadata

Metadata is data that defines data. Metadata conveys information about how data is formatted, structured, and stored. In an OLAP cube, metadata defines dimensions, levels, members, member attributes, and interrelationships of the cube. PeopleSoft Cube Builder uses two types of PeopleSoft structures—trees and queries—to define cubes.

PeopleSoft Trees

A PeopleSoft tree defines the summarization rules for a database field. It specifies, for purposes of reporting or security access, how the field values are grouped in the system.

For example, the values of the DEPTID field might identify individual departments in your organization. You could build a tree for the DEPTID field, which defines the organizational hierarchy that specifies how each department relates to the others: departments 10700 and 10800 report to the same manager, department 20200 is part of a different division, and so on.

You can easily see how you can use PeopleSoft trees to define a cube structure. Like cube dimensions, trees consist of levels and members. (In PeopleSoft Tree Manager, members are called *nodes* and *leaves*.)

See "PeopleSoft Tree Manager Overview" (PeopleTools 8.58: Tree Manager).

PeopleSoft Queries

PeopleSoft queries are SQL statements that are created by PeopleSoft Query. You can use these SELECT statements to return field values based on certain criteria. The standard PeopleSoft security mechanism can secure the data returned by PeopleSoft Query. Also, PeopleSoft Query can return data in any of the database-supported globalized formats.

You can use queries in a number of ways to define an OLAP cube. Finally, you can use queries to populate OLAP cubes with data; the query results are the rows of data that fill the cells of the cube.

See "PeopleSoft Query Overview" (PeopleTools 8.58: Query).

Understanding Metadata Types

Three types of metadata are available that define an OLAP cube:

- Dimensional metadata (hierarchy and members)
- Member property metadata (consolidation method, sign flip, and label)
- Cube metadata (dimensions and measures)

You use PeopleSoft Tree Manager and PeopleSoft Query to describe all of this metadata to PeopleSoft Cube Builder.

Understanding Oracle Essbase Properties

Oracle Essbase has the following valid property types:

- Data storage

Data storage enables Essbase to recognize what type of storage to allocate for the member. Valid values are *0* or *blank* (store data), *1* (never share), *2* (label only), *3* (shared member), *4* (dynamic calculation and store), and *5* (dynamic calculation, no store).

PeopleSoft Cube Builder sets the default value as store data for all members in the first rollup and the non-detail nodes of all other rollups. Detail nodes in secondary rollups are set to shared members.

- Expense item

Expense item applies to account dimensions only. Essbase has certain built-in formulas that can take advantage of the knowledge that an item is an expense. To pass this knowledge on to Essbase, you should use this property. Valid values are *Blank* (set) and *non-Blank* (do not set).

- Time balance

Time balance affects how the parent time value is calculated. Valid values are *0*, *1*, *2*, and *3*, which correspond to none, first, last, and average, respectively.

- Consolidation

This property enables you to define the mathematical operator used for rolling up members. Most often, you expect that data is added (using the + operator) when rolled up. However, you might occasionally need to specify other operators, such as those listed in the following table.

| Valid Value | Action |
|--------------------|--|
| + (plus sign) | Add (default). |
| – (minus sign) | Subtract. |
| * (asterisk) | Multiply. |
| / (forward slash) | Divide. |
| [Blank] | Do not consolidate. |
| ~ (tilde) | Do not consolidate. |
| % (percent sign) | Divide the total of previous member calculations by this member and multiply by 100. |

Note: See the Oracle Essbase documentation for more information.

Understanding PeopleSoft Process Scheduler Integration

PeopleSoft Process Scheduler includes a process type definition specifically for use with PeopleSoft Cube Builder. This process type is the Essbase Cube Builder process type, and you invoke it whenever you launch the process to create a cube from the standard run control page. During this process, the data and metadata are translated into a format that is understood by Oracle Essbase.

See [Building Essbase Cubes](#).

Using Trees

Metadata that exists in PeopleSoft trees can be particularly useful when you design cube dimensions. The main reason to use an existing tree is to leverage the rules that are associated with the outline that the tree represents. Because trees are used to validate information that is stored in the OLTP database, all of that tree information is already related to the transactional data. Using effective-dated trees in an outline generates the automatic evolution of your data that is used for data analysis.

PeopleSoft Cube Builder leverages the information that is already stored in your PeopleSoft trees as outlines upon which to build each dimension. Using the Dimension page in Essbase Cube Builder, you map a tree to a dimension so that the rollup of the resulting cube dimension is the same as that of the specified tree.

By default, data is summarized exactly as the tree is defined. Each node and detail value becomes a member of the cube hierarchy for that dimension. The descriptions of the nodes and details become the labels, or aliases, of the members.

You might want to use existing trees for your dimensions, or you might need to create new trees. If you have an existing tree that is close to what you want the dimension to look like, make a copy of the tree and modify that copy.

If the hierarchy that you want to use is a subset of an existing tree, you do not have to create a new tree. PeopleSoft Cube Builder enables you to use a subset of an existing tree by specifying a starting node, Top Node, and the number of levels below the top node of the tree to include in the hierarchy. You can use more than one tree belonging to the same business unit to define a single hierarchy.

In addition, if a tree does not provide the structure that you need for a dimension, you can add members, attributes, and generations by using one or more queries to provide the additional metadata.

PeopleSoft Cube Builder treats uppercase and lowercase characters as distinct, so the names *ABC*, *Abc*, and *abc* are all considered unique member names. However, Oracle Essbase offers an option to change all member names to uppercase. If you enable this option, you create problems in PeopleSoft Essbase Cube Builder with members that are identical except for their letter casing.

Note: PeopleSoft Cube Builder permits duplicate node names if you cannot avoid the duplication.

In Essbase, a dimension can have multiple rollups. The resulting total for the first rollup is calculated differently from the resulting totals for subsequent rollups. For example, a dimension exists with two rollups. Two different trees are used for these two rollups: the first tree is set up as A, B, C and the second tree is set up as A, D, C.

Assume that A, B, C, and D have the following fact data values: 2, 5, 10, and 20 respectively. The total for the first rollup, A, B, C, is 17 because the total equals 2 + 5 + 10. The total for the second rollup, A, D, C, is 10 because each parent gets its total from its children. Because C has the fact value 10, the total for C is 10. Because D is the parent of C, D gets its total from its children. Therefore, D gets 10 from C. A gets its total from its children, which is D. Therefore, A has the total value of 10.

Creating Queries

You can create several types of queries to use with PeopleSoft Cube Builder, all of which you must define as user (ad hoc) queries rather than role queries or database agent queries.

Using Dimension Queries

Dimension queries enable you to define the dimension structure using query results instead of, or in addition to, a tree. However, remember that you are using queries to create a tree-like structure.

You can convey hierarchical information by parent/child relationship or by a narrow query.

In PeopleSoft Cube Builder, dimension queries can be dynamic or static. A dynamic query indicates that any incremental change in the tables that the query uses are reflected in the next run of PS2Essbase. A static query indicates that further changes to the tables used to create the first hierarchy will not be reflected unless the hierarchy is reloaded manually.

PeopleSoft Cube Builder uses dynamic queries to populate members at the leaf levels of a hierarchy and under the same parent.

Using Data Source Queries

Data source queries define the data that you bring into the cube. Writing a data source query is straightforward; the query must return one column for each dimension and one column for the measure. Assume that you want to build a data source query for a cube containing amounts that are dimensioned by account, department, and period.

The output of your query has four columns, as shown in the following table.

| Account <i>(Dimension)</i> | Department <i>(Dimension)</i> | Period <i>(Dimension)</i> | Amount <i>(Measure)</i> |
|--------------------------------------|---|-------------------------------------|-----------------------------------|
| XXX | XXX | XXX | XXX |
| XXX | XXX | XXX | XXX |

You can use several queries as the data source for a single cube, to load data into the measure. Every data source query that is used must include an output column for every dimension that is used and for the measure. In the version of the cube builder, only one measure is allowed per cube.

The following tables show examples of how you can use two separate queries as a data source for a cube. Note that both queries return columns for every dimension, as required, and that they differ only in which measure they include.

Results of Query 1.

| Account <i>(Dimension)</i> | Department <i>(Dimension)</i> | Period <i>(Dimension)</i> | Budget Amount <i>(Measure 1)</i> |
|--------------------------------------|---|-------------------------------------|--|
| 1000 | DEV | Q4 2003 | 4000 |
| 1100 | SALES | Q4 2003 | 6000 |

Results of Query 2.

| Account <i>(Dimension)</i> | Department <i>(Dimension)</i> | Period <i>(Dimension)</i> | Actual Amount <i>(Measure 2)</i> |
|--------------------------------------|---|-------------------------------------|--|
| 1000 | DEV | Q4 2003 | 3000 |
| 1100 | SALES | Q4 2003 | 5000 |

Resulting cube using Query 1 and Query 2 as data sources.

| Account <i>(Dimension)</i> | Department <i>(Dimension)</i> | Period <i>(Dimension)</i> | Budget Amount <i>(Measure 1)</i> | Actual Amount <i>(Measure 2)</i> |
|--------------------------------------|---|-------------------------------------|--|--|
| 1000 | DEV | Q4 2003 | 4000 | 3000 |
| 1100 | SALES | Q4 2003 | 6000 | 5000 |

Incremental Updates

Two types of incremental updates are available:

- Metadata

Incremental updates for metadata refers to the addition, removal, or modification of dimension members and/or their properties.

- Data

Incremental updates for data refers to the aggregation of delta values to the cube data cells product of dimensional intersections of members.

Data Cell Types

By default, two categories of data cells are available for incremental data updates:

- Calculated data cells (or non-leaf nodes)

Calculated data cells store values of dimensional intersections of non-level zero members (branch nodes), which are calculated based on the consolidation properties of their children.

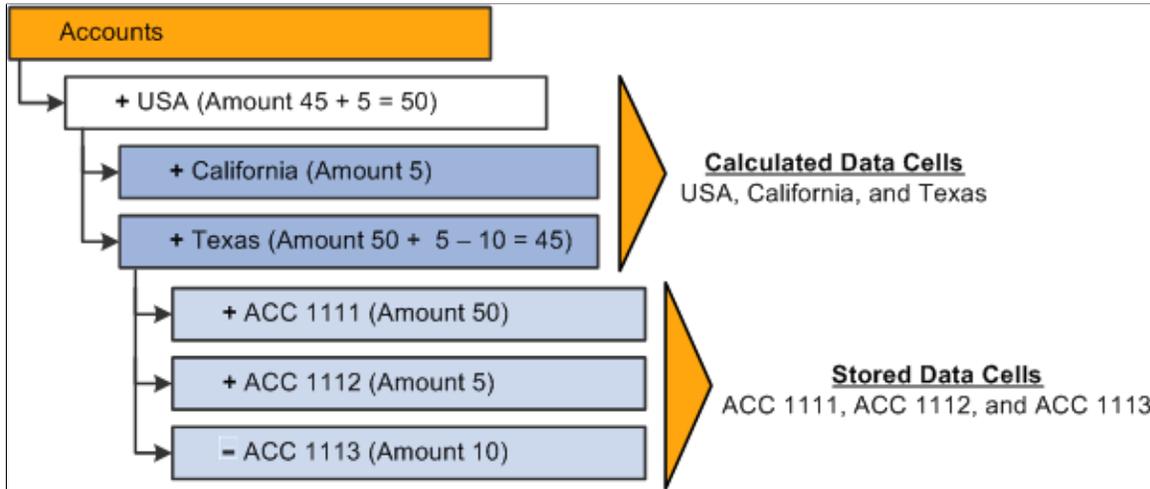
- Stored data cells (or leaf nodes)

The stored data cells store values of dimensional intersections of level zero members (leaves) assigned manually.

Note: You can modify the stored properties, calculated properties, or both on any member at any level.

Image: Example of how to calculate upper data cells based on stored data cells

This example shows how to create data queries for incremental updates, taking as a reference the next Accounts dimension.



Aggregations on Load Tool

Essbase uses the term *Aggregations on Load* (replace, addition, and subtraction) as *incremental updates for stored data cells*. Essbase supports three data Aggregations on Load options:

- Replace current value

Use the *Replace* option to overwrite or replace the currently stored value in the data cell with the value being passed. Basically, if the data file has multiple rows with the same dimensional intersection, then the last value in the data cell is stored and the previous ones are ignored.

- Add to existing values

Use the *Addition* option to add all the values for the stored data cell to the currently stored value. Use this option for instances in which loads are always additive, for example, an implementation that loads transactions to an already stored summary amount.

- Subtract from existing values

Use the *Subtraction* option as the opposite of the *Addition* option. It subtracts the values passed from the currently stored value. This option is rarely used, but you can use it to perform an additive load that then needs to be backed out.

Image: Create Cubes page with the Aggregations on Load options of Replace, Addition, and Subtraction

This example shows the aggregations on load options in the run control ID of the Create Cubes page (CUB_RUNCNTL).

The screenshot shows the 'Create Cubes' page with the following configuration:

- Run Control ID: RUNCNTL
- Report Manager
- Process Monitor
- Run button
- *Outline: QE_CUBT_TRANS_DATE
- *Essbase Connection ID: QE_CUBT_API_TEST
- *Application Name: GEN
- *Database Name: GEN1
- *Meta-Data Update Action: Create or Re-create
- *Restructuring options: Discard all data
- *Aggregation On Load: Replace
- Log level: 31
- Dimension: QE_CUBT_TRANS_DATE
- Query: QE_CUBT_TRANSACTIONS
- Last Update Dttm: 10/27/2015 11:06:21AM

Note that:

- The aggregations on load options are ignored when only metadata (dimension) is built.
- When you build multiple factories (multiple data queries) in the same build, the same Aggregations on Load option is used for all of the factories.
- The default value of the Aggregations on Load drop-down list is *Replace*.
- After you run the PS2Essbase process, the Aggregations on Load option serially builds each dimension and then serially builds each factory. The factory creates a data file with all the data cells to be loaded to the Essbase server.

Example: Designing Data Queries for Incremental Updates

Image: Example of how to create data queries for each Aggregations on Load option, next Accounts dimension

This example shows how to create data queries for each Aggregations on Load option, taking as a reference the next Accounts dimension.

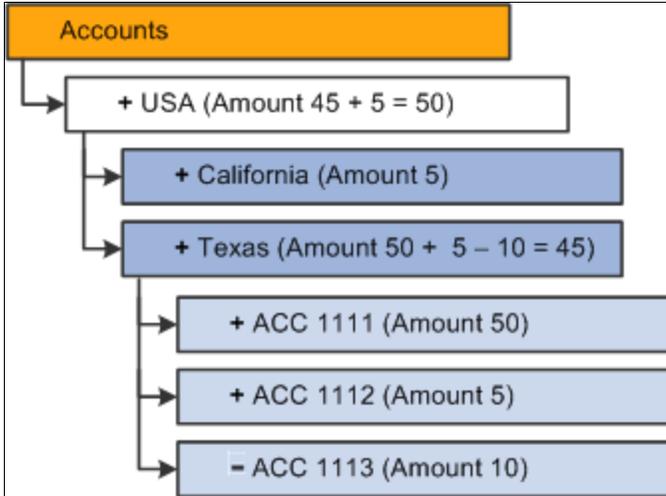


Image: Example of how to create data queries for each Aggregations on Load option, next staging table

This example shows how to create data queries for each Aggregations on Load option, taking as a reference the next staging table that contains the data cells to be loaded.

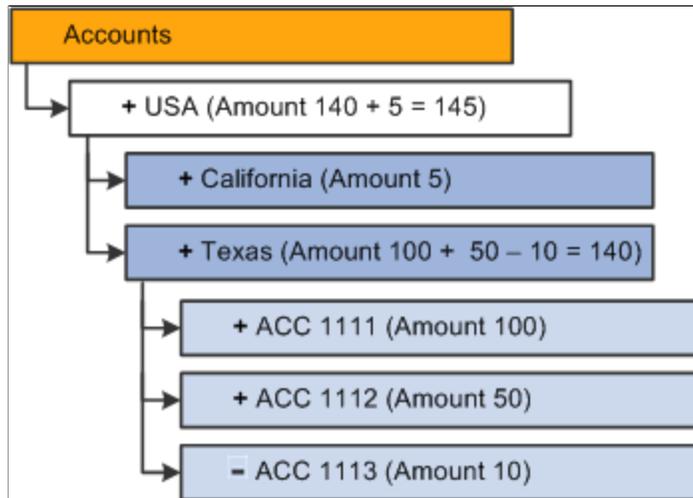
| Account | Amount |
|----------|--------|
| ACC 1111 | \$200 |
| ACC 1111 | \$100 |
| ACC 1112 | \$10 |
| ACC 1112 | \$50 |

Essbase Cube Builder can be used to produce three different results:

1. Results of replacing the existing data cells.

Image: Results of replacing the existing data cells

This diagram illustrates the resulting dimension if the *Replace* (replace data load) option is selected.



Note: The last value passes for each cell is the valid value. In this example, cells ACC 1111 (\$200) and ACC 1112 (\$10) are ignored.

When you want to perform an additive incremental update using the *Replace Aggregations on Load* option, you need to create a data query that summarizes repeated fields into one field. For example, to achieve an additive incremental update on stored data cells, you need to create a data query that summarizes repeated fields into one field.

Image: Example of summarizing repeated fields into one field

This example illustrates how to create a data query that summarizes repeated fields into one field to achieve an additive incremental update on stored data cells.

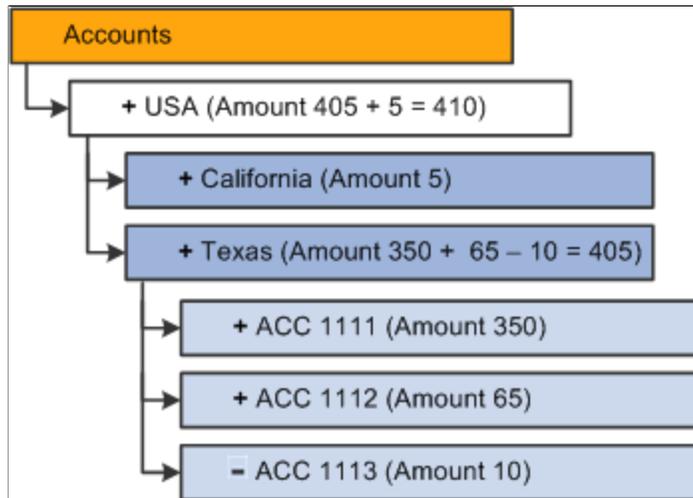
| Account | Amount |
|----------|--------------------------------|
| ACC 1111 | $\$50 + \$200 + \$100 = \350 |
| ACC 1112 | $\$5 + \$10 + \$50 = \65 |

This example illustrates the SQL query for summarizing repeated fields into one field:

```
SELECT DISTINCT A.ACCOUNT,
  (
    SELECT SUM(B.AMOUNT)
    FROM STAGING_TABLE B
    WHERE A.ACCOUNT = B.ACCOUNT
  )
FROM STAGING_TABLE A;
```

Image: Results of replacing incremental updates

This example illustrates the results of replacing incremental updates.



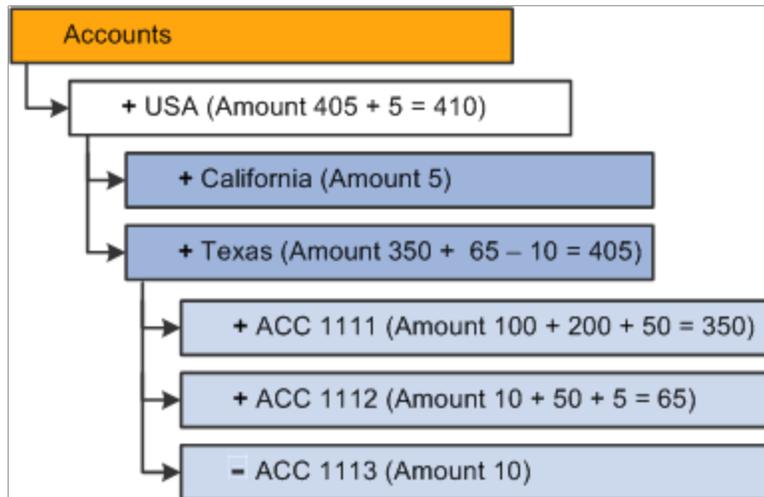
Note: The current stored value is overwritten and lost, therefore, in the staging table used for the data query loads, you must not eliminate the previous increments that are already loaded into Essbase. These previous increments are required to recalculate the total summarized amount in each load. In this section, the concept of staging tables is used frequently; notice that the staging tables are not tables that PeopleTools creates or maintains, the staging tables are optional customer or application tables. A staging table is any table that is used to store temporary increments to be loaded into Essbase. Depending on the option selected in the Aggregation on Load drop-down list, the staging tables may be cleaned up or maintained.

2. Results of additive incremental updates.

Results of additive incremental updates are the previously stored values plus each one of the new cells that corresponds to the cube intersection are added.

Image: Results of additive incremental updates

This example illustrates the resulting dimension when the *Addition* (addition data load) option is selected.



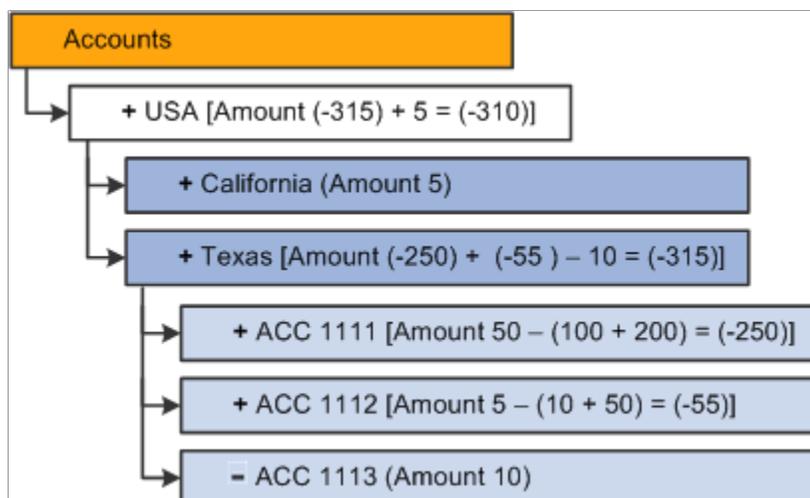
The resulting amounts in the dimension after the build is the same as when you want to perform an additive incremental update using the *Replace Aggregations on Load* option.

3. Results of subtractive incremental updates.

Results of subtractive incremental updates are the previously stored values minus each one of the new cells that corresponds to the cube intersection conforms to the result for the stored data cells.

Image: Results of subtractive incremental updates

This example illustrates the resulting dimension when the *Subtraction* (subtraction data load) option is selected.



In almost all instances, *Replace* is the preferred option over *Addition* and *Subtraction* because *Replace* makes forming queries against relational sources easier (it summarizes data to the dimensional intersection versus having duplicate rows in the load file). The *Replace* option is also a better alternative because it reduces the number of records to be passed to Essbase, thus improving performance. Moreover, the *Addition* and *Subtraction* options make recoverability more difficult after the database fails while

loading the data. Although when a recover is required, Essbase lists the number of the last rows committed in the application event log file, which is used to restart the build process.

If you run the cube using the *Addition* or *Subtraction* options, you need to clear the staging table after each build so that the values that are already loaded is not required again. However, PeopleSoft Cube Builder does not modify or remove the loaded cells from the staging table; either you or the application that uses the Cube Builder is responsible for cleaning up the staging table after the incremental data is loaded. If the staging table is clear, the notification of the Process Scheduler shows that the cube building process runs successfully. To successfully clear the staging table, you should create a PSJob that first runs the PS2Essbase process and then runs another AE program to clear the specific staging table.

Note: PSJob and the AE program are not provided by PeopleTools.

Defining Essbase Cube Builder Connection Definitions

Defining Essbase Cube Builder Connections

The first step in creating Essbase cubes is to create connectivity information using the Connections page (CUB_CONNECTID). The Connections page enables you to define all the connectivity information, including the server name, user access, user ID, and user password.

Navigation

PeopleTools > Cube Builder > Connections

Image: Connections page

This example illustrates the fields and controls on the Connections page. Definitions for the fields and controls appear following the example.

Connections

Essbase Connection ID QE_CUBT_SMOKE

*Description Smoke Connection

Comment

Owner ID PeopleTools

*Server localhost

*Database State Design

*User Name admin

*Password

Unicode Enabled:

Updated By QEDMO Updated on 03/10/09 3:39:47PM

Owner ID

Select the owner of the Essbase connection.

The owner ID helps to determine the product, feature, or application team that last made changes to this Essbase connection.

The available options are *Mobile Company Directory*, *PeopleTools*, and *PeopleTools Demo*.

Server

Enter the Essbase network server name or IP address.

Database State

Set the default user access level to the database after it is created or modified.

If explicit security is not granted in Essbase, then it applies to every user except the system administrator, to whom the User Name and Password options apply.

The available options are *Calculate*, *Design*, *None*, *Read*, and *Write*.

Oracle recommends that you set this option to *Read* at the least.

User Name

Enter the Essbase user name.

Password

Enter the Essbase password, which will be encrypted in the database table.

Note: Only users with the following permission lists are able to create the Essbase connectivity information: AEPNLS, ALLPAGES, CPPT1080, PTPT1200, and SMPPNLS.

Unicode Enabled

Specify whether the database to be created will contain Unicode characters.

Note: Because selecting this option creates an Essbase Unicode database, the Essbase server must be in Unicode mode. The Unicode Enabled option on this page is not synchronized with the setting in the PSSTATUS table, which enables customers to override the Unicode setting in the PeopleSoft environment if desired.

Last Upd User (last update user)

Displays the ID of the user who last modified the Essbase connection.

Last Upd DtTm (last update date and time)

Displays the date and time the Essbase connection was last modified.

Chapter 4

Creating Dimensions for Essbase Cubes

Understanding Dimensions of Essbase Cubes

After creating an Essbase cube connection, you can create Essbase dimensions. Dimensions represent different aspects that are related to a multidimensional database. Typical dimensions are:

- A time dimension, which is used to map transactional information over a time period.
- An account dimension, which is used to map various affected accounts.
- A country dimension, which is used to map geographic transactional areas.

Groupings of dimensions make up a cube outline.

Setting Up Cube Dimension Builder

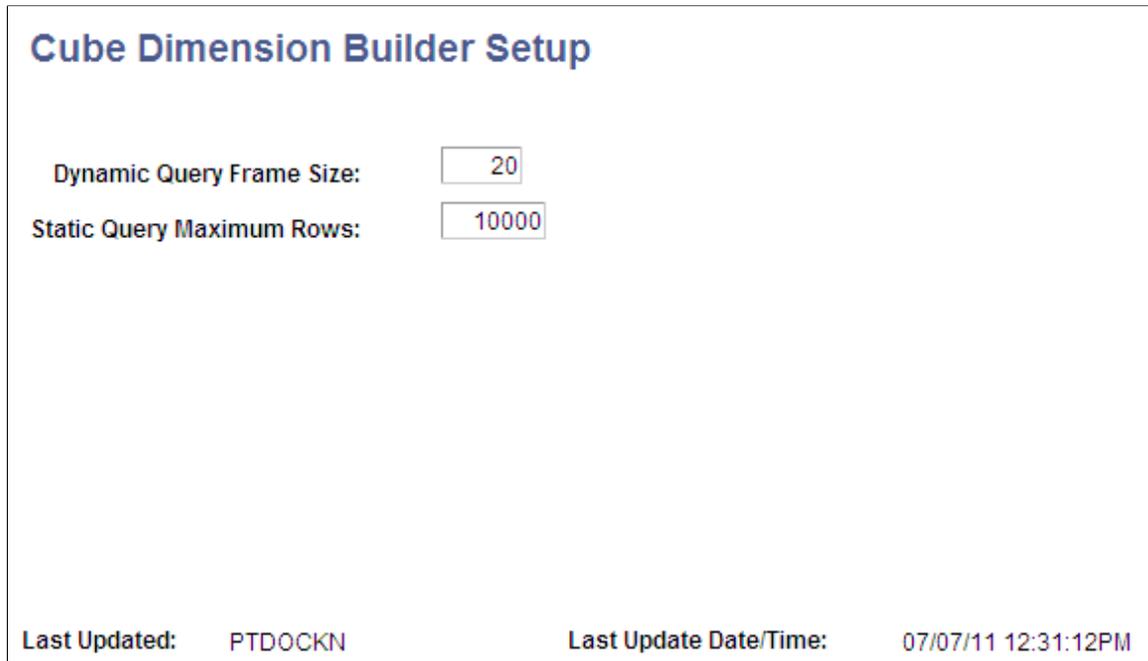
Use the Cube Dimension Builder Setup page (CUB_GUISETUP) to set values for the number of members to display at a time for dynamic queries, and to set the maximum number of rows to retrieve for static queries.

Navigation

PeopleTools > Cube Builder > Cube Dimension Builder Setup

Image: Cube Dimension Builder Setup page

This example illustrates the fields and controls on the Cube Dimension Builder Setup page. Definitions for the fields and controls appear following the example.



Dynamic Query Frame Size

Enter the number of items to show at a time for a dynamic query.

The dynamic query frame size controls the frame size for the dimension builder to use. This number can be set to any number from 10 to 1000. If the number of rows returned by a dynamic query exceeds this frame size number, then the dimension builder enters into chunking mode and displays a link to enable users to view the next and previous frame of the dynamic query.

Static Query Maximum Rows

Enter the maximum number of rows to be retrieved by the dimension builder when you run large static queries.

In general, you should avoid large static queries. Typically, when you build the structure of the dimension, entering the higher levels of the dimension manually—without queries or trees and with only the leaf values (level 0 members)—requires large queries. You should use dynamic for large queries.

For dynamic queries and trees, you cannot manually add more members so you would not typically use these queries or trees to create the higher order members of the dimension.

Last Updated

Displays the ID of the user who last modified the Essbase connection.

Last Update Date/Time

Displays the date and time that the Essbase connection was last modified.

Creating Essbase Dimensions

This topic discusses how to create Essbase dimensions in the dimension page.

Defining an Essbase Dimension

Use the Dimension page (CUB_DIMENSION) to define an Essbase dimension.

When you first access the Dimension page to create a dimension, this page displays dimensional properties and a blank Dimension Structure section with a root node that has the same name as the dimension; most of the action icons at the bottom are disabled. Later on, depending on the Dimension Type value, the tree object members may or may not have all of the available properties that are shown in the following example. Action icons appear available as they become relevant when you build the dimensional structure.

Navigation

PeopleTools > Cube Builder > Dimensions

Image: Dimension page when you first access it to create a dimension

This example illustrates the fields and controls on the Dimension page when you first access it to create a dimension.

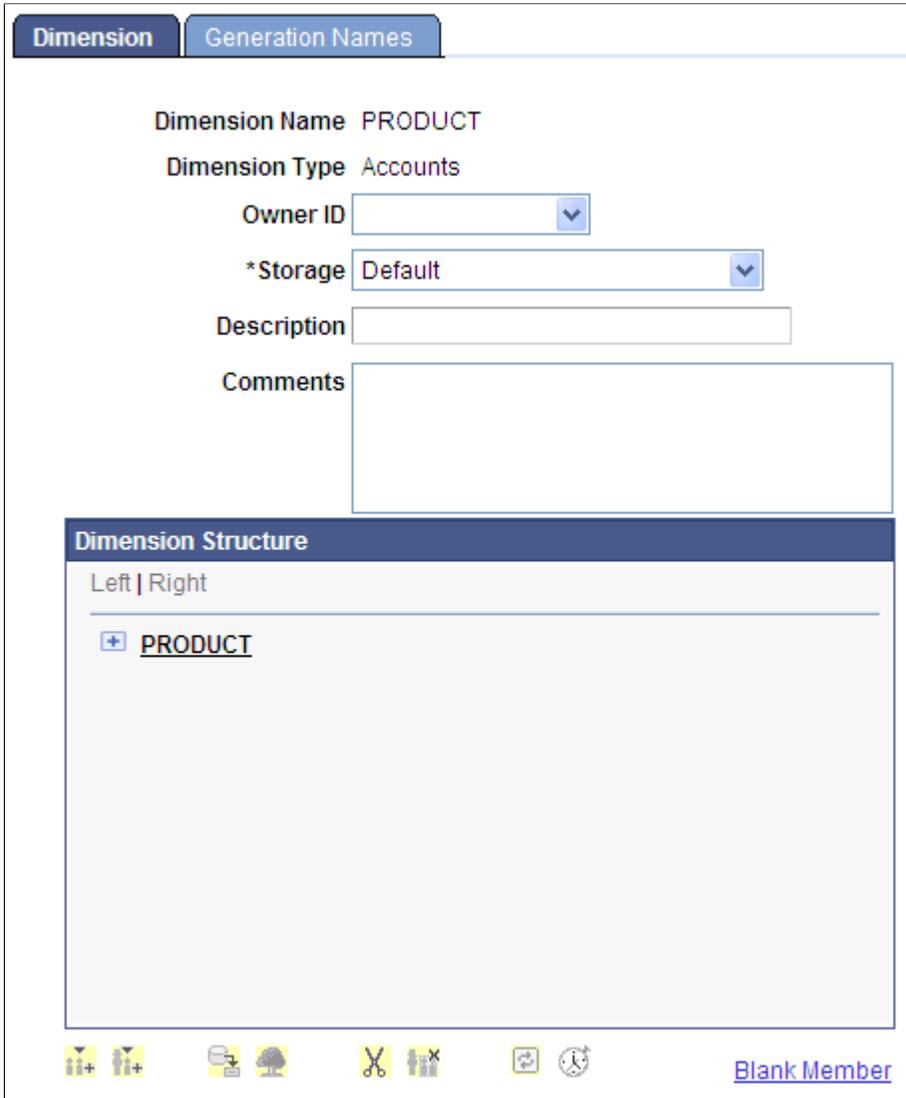


Image: Dimension page with some properties and active icons

This example illustrates the fields and controls on the Dimension page with some properties and active icons.

The screenshot displays the 'Dimension' page for 'QE_CUBT_EURO_PRODUCTS'. The 'Dimension Type' is set to 'None'. The 'Owner ID' is 'PT', and the '*Storage' is 'Default'. The 'Description' is 'Standard Dimension'. The 'Comments' field is empty. The 'Dimension Structure' pane shows a tree view with 'TOP_NODE' and its children: '[VAT_FREE - (+)(VAT Free Products)]', '[EP602 - (+)]', '[EP603 - (+)]', '[EP605 - (+)]', and '[HIGH_VALUE_PRODUCTS - (+)(High Value Products)]'. The 'Member Properties' pane shows '*Member Name' as 'TOP_NODE', with options for 'Label Only' and 'Never Share'. The 'Dynamic Source' section shows 'Tree' as 'QE_CUBT_PRODUCTS', 'SetID' as 'QEEU', and 'Effective Date' as '02/27/2009'. A 'Save' button is visible at the bottom left.

Dimension Type

The system displays the type of the dimension. Available options are:

- *Accounts*
- *Country*
- *None (or Standard)*
- *Time*

Owner ID

Select the owner of the dimension.

The owner ID helps to determine the product, feature, or application team that last made changes to this dimension.

Storage

Select from the following storage options:

- *Default*
- *Dynamic Calc*
- *Dynamic Calc and Store*
- *Label Only*

- *Never Share*

Dimension Structure



Displays dimension data graphically to show a tree hierarchy.

The Add Member Peer icon is available when you highlight a tree member.

Click this icon to add a new member that is a peer of the currently highlighted member.



If the current member can have children, the Add Member Child icon is available when you highlight a tree member.

Click this icon to add a new member that is a child of the currently highlighted member.



Click the Assign From Query icon to run a public query. The results of running the query will be placed directly under the currently highlighted member.

You can select which query to run, and you need to supply any prompt values that the query requires. You need to supply the column of the query to use to populate the child members, which will be created when the query is run. You also can select the query to be run in dynamic mode. The query must return at least one row of data or it is considered invalid and is not used.



Click the Assign From Tree icon to copy a tree structure into the dimension.

Select the starting node of the tree; the system copies the selected node and all of its children. The currently highlighted node becomes the selected tree top node, and the member name of the current node is changed to reflect this selection. As with queries, you can select to have the tree copy be dynamic.



Click the Delete Member icon to delete dimension (children) members.



Click the Delete All Children icon to delete all query or tree children.



Click the Reset Values icon to clear or refresh member override values, if any, that you have entered for dynamic trees and queries.

Note: This icon is used for dynamic queries and trees when the concept of member overrides and templates is defined. This clearance is important if you have changed a template and want to reset all previously saved values.



Click the Time Dimension Wizard icon to access the Time Dimension Wizard page, where you can automatically populate the members of a time dimension.

Note: The Time Dimension Wizard icon is available only for the time dimensions and not for other dimension types.

Blank Member

Click to access the Dimension Options page where you can specify a unique name for the blank member in the dimension.

The name that you specify for a dimension member is used as a placeholder for any blank member that Cube Builder finds in the resulting query data rows.

Member Name

Enter the name or identifier of the member.

Label Only

This option is available only for child members, which are not at level 0.

Select to enable the system to group members together (with a dummy parent member) and enable easier access to the report.

Note: If you define a parent member as label only, the system does not calculate parent member from its child and reserves no storage space for that parent member. Use the default calculations for parent members to save time and storage space. When you view the report in Microsoft Excel, the label-only member shows its first child value and does not occupy any storage space.

Expense Report

Expense item applies to account dimensions only. Essbase has certain built-in formulas that can take advantage of the knowledge that an item is an expense.

Select this check box to pass this knowledge on to Essbase. Valid values are *Blank* (set) and *non-Blank* (do not set).

Shared Member

Select to share values between members.

Never Share

Select to not allow members to be shared implicitly.

Note: Members that are set as Never Share can only be explicitly shared. To explicitly share a member, create the shared member with the same name and set it as *shared*.

Consolidation

This property enables you to define the mathematical operator used for rolling up members. Most often, you expect that data is added (using the + operator) when rolled up. However, you might occasionally need to specify other operators. The available operators are:

- % (percent sign): Divide the total of previous member calculations by this member and multiply by 100.
- * (asterisk): Multiply.
- + (plus sign): Add.

- – (minus sign): Subtract.
- / (forward slash): Divide
- ~ (tilde): Do not consolidate.
- [Blank]: Do not consolidate.

Time Balance

Time balance affects how the parent time value is calculated. Select from these values:

- 0: none
- 1: first
- 2: last
- 3: average

Alias

You have the options to either:

- Manually enter each alias every time the cube is built or rebuilt, or
- Populate member aliases together with the member names from a query.

The member aliases are populated from the existent narrow queries and the wide queries, and member aliases are populated when you create a dimension using a query.

After saving the dimension, member aliases are displayed next to member names.

Note: When you use a tree to create a dimension, the descriptions of the nodes will be the member aliases.

Storage

Data storage enables Essbase to recognize what type of storage to allocate for the member. Select from these values:

- 0 (or blank): store data.
- 1: never share.
- 2: label only.
- 3: shared member.
- 4: dynamic calculation and store.
- 5: dynamic calculation, no store.

Formula

Click to access the Member Formula page, where you can enter formula information for a dimension member.

Dynamic Source

If you select a dynamic query or dynamic tree member of the dimensional structure, the attributes on the right also display the dynamic object key values that were used in creating the entries:

- For a query, this section displays the query name that was used and if a particular template was used.
- For a tree, this section displays the tree name and the key values of the tree.

Using the Blank Member Option

PeopleSoft Cube Builder uses queries and trees to define the structure of a dimension. You can define a dimension query that returns data rows used to generate the members of a dimension as well as its hierarchical structure, or insert a tree structure into a dimension.

To populate the data in a cube, you need to define a data source query that must return one column for each dimension and one column for the measures (facts). If you are building a data source query for a cube that contains the amounts dimensioned by account, region, product, and date, then the output of the query should have five columns.

Image: Data source query of a cube

This example shows the data source query of a cube that contains the amounts dimensioned by account, region, product, and date.

| Account | Region | Product | Date | Amount |
|-------------|------------|--------------|-------------------|------------|
| E100 | 402 | EP602 | 2008-10-20 | 20.1 |
| E401 | 401 | EP701 | 2008-10-27 | 17.52 |
| E202 | 402 | AP802 | 2008-10-27 | 17.52 |
| E102 | 401 | EP602 | 2009-02-01 | 30.24 |
| E202 | 403 | AP803 | 2009-01-01 | 50.16 |
| E102 | 401 | EP601 | 2008-03-01 | 23.46 |
| E102 | 401 | EP603 | 2009-01-01 | 100 |
| E402 | 404 | XP801 | 2009-02-17 | 24.56 |
| E401 | 405 | XP900 | 2009-01-19 | 34.47 |
| A201 | 402 | AP801 | 2009-02-04 | 123 |
| E301 | 403 | AP801 | 2008-12-23 | 24.67 |

The dimension columns (Account, Region, Product, and Date) contain the dimension member names, and the measure column (Amount) contains the value of the measure. Thus, each result data row defines the data cell to be populated and the value of the measure column. For example, the amount (measure) of 100 is the results of the data cell where account equals E102, region equals 401, product equals EP603, and date equals 2009-01-01.

If the data is not defined for every dimension in the result data cells, then some data queries return rows with blank or empty dimension members.

Image: Data source query for a cube with blank members

This example shows the data queries that return rows with blank or empty dimension members.

| Account | Region | Product | Date | Amount |
|---------|--------|---------|------------|--------|
| E100 | 402 | EP602 | 2008-10-20 | 20.1 |
| E401 | 401 | EP701 | 2008-10-27 | 17.52 |
| | 402 | AP802 | 2008-10-27 | 17.52 |
| E102 | 401 | | 2009-02-01 | 30.24 |
| E202 | 403 | AP803 | 2009-01-01 | 50.16 |
| E102 | 401 | EP603 | 2009-01-01 | 100 |
| E402 | | XP801 | 2009-02-17 | 24.56 |
| E401 | 405 | XP900 | 2009-01-19 | 34.47 |

If blank members are returned and the blank member name is not specified, then the data load processing of Cube Builder ignores the rows that contain blank members. Thus, some measures are not added to the rollup total, and the rollup total is not accurate

To accurately add all measures to the rollup total when a member has no name, you use the Blank Member link in the Dimension page to access the Dimension Options page, where you can specify a unique name for the blank member in the dimension. The name that you specify for a dimension member is used as a placeholder for any blank member that Cube Builder finds in the resulting query data rows.

Using the Dimension Options Page

Use the Dimension Options page (CUB_DIM_OPTIONS) to specify a name for the blank member.

- If the Blank Member field is specified, when you create dimension or load data, Cube Builder creates a dimension member for the blank member using the value in the Blank Member field to hold the blank member.
- If the Blank Member field is blank, Cube Builder ignores the blank member when you create dimension or load data.

Navigation

PeopleTools >Cube Builder >Dimensions >Blank Member

Image: Dimension Options page

This example illustrates the fields and controls on the Dimension Options page.

Dimension Options

Blank Member

Image: Dimension for the blank member from a static tree

This example shows the dimension for the blank member (No Product) from a static tree when the Blank Member field is specified as *No Product*.



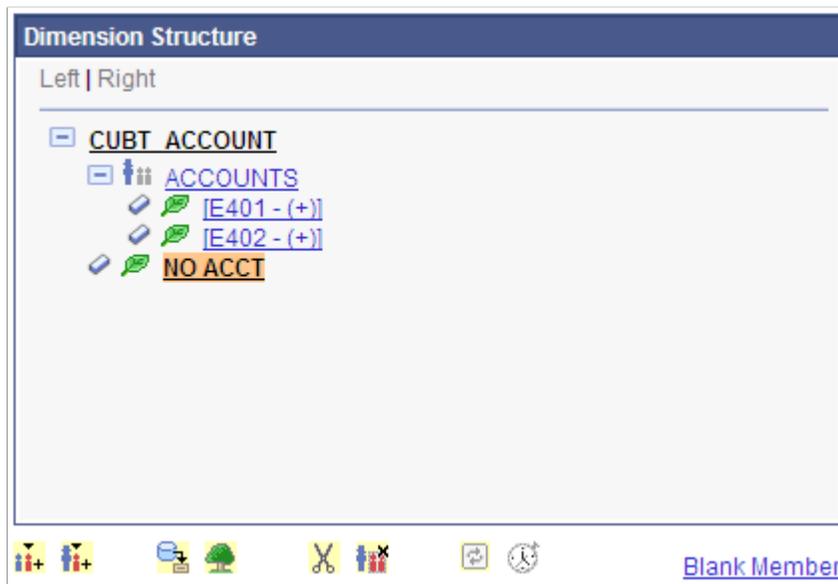
If the dimension is not populated from a static or dynamic tree structure, you can manually add the blank member name to the dimension and then specify the Blank Member field value using the same name to indicate that the dimension member using the specified name is the blank member.

To manually add a blank member name to a dimension:

1. In the Dimension Structure section of the Dimension page, add a dimension member.

Image: Dimension member (NO_ACCT) in the Dimension Structure section

This example shows the newly added dimension member (NO_ACCT) in the Dimension Structure section.



2. Select the newly added dimension member (NO_ACCT) and click the Blank Member link.

The Dimension Options page appears.

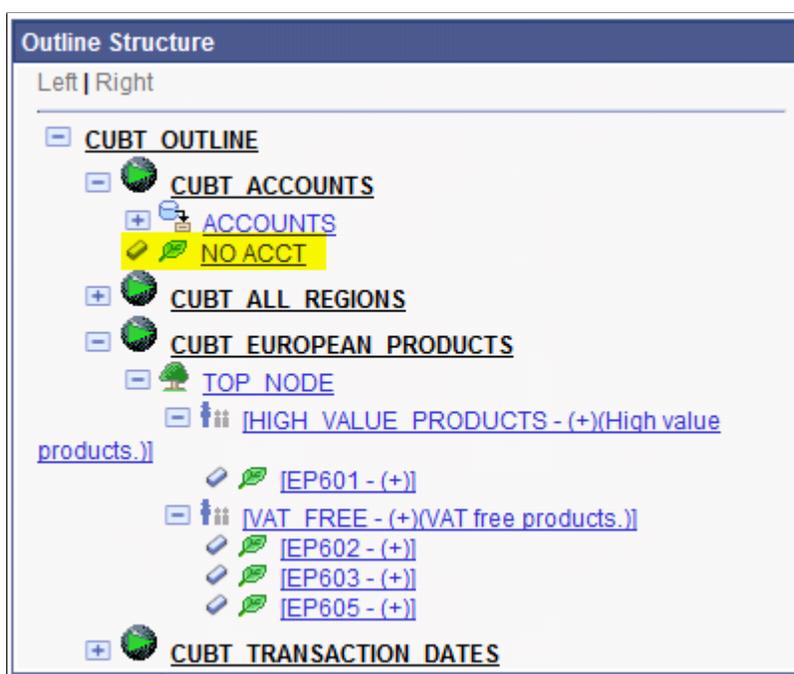
3. Enter a name (NO_ACCT) in the Blank Member field.

4. When Cube Builder creates the dimension structure:

- If the dimension member (NO_ACCT) has the same name that is specified in the Blank Member field on the dimension, then the dimension member is used as a place holder for the blank member.

Image: Outline displays the blank member

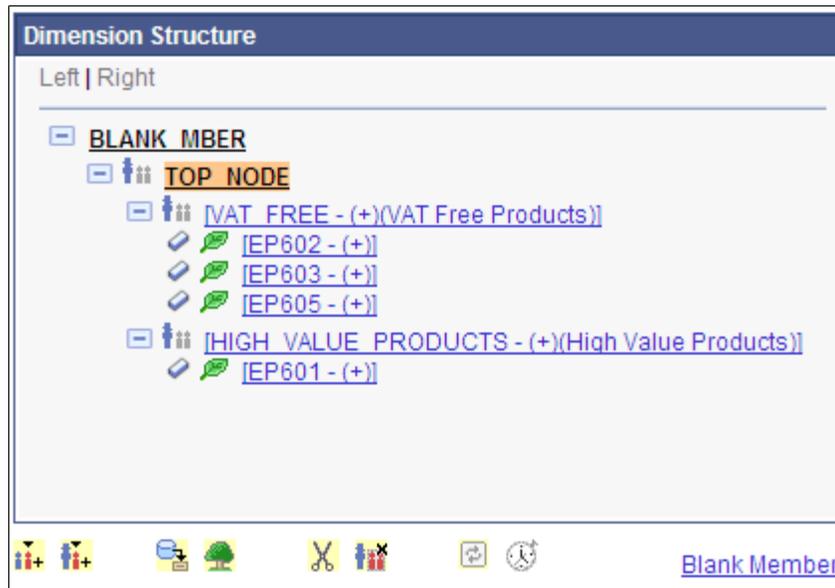
This example shows the outline that displays the blank member (NO_ACCT) under the CUBT_OUTLINE dimension.



- If blank member is not defined for the dimension, then when you insert a tree structure with a blank member (leaf) into the dimension, the blank member is ignored and no blank dimension is created.

Image: Outline displays the dimension structure without the blank dimension created

This example shows the dimension structure without the blank dimension created.



5. When you run the data to create or populate the cube, Cube Builder checks whether a blank member is defined for the dimension.
 - If the dimension has blank members and the blank member name is defined, then the measure value is populated to the cell for the blank member of the dimension.
 - If no blank member is found in the dimension or no blank member name is defined, then the measure value is ignored.

Image: Results from the data source query with a blank member

This example shows the results from the data source query with a blank member in the Account dimension.

| Account | Product | Amount |
|---------|---------|--------|
| E402 | EP603 | 21.00 |
| E402 | EP605 | 17.91 |
| E401 | EP602 | 16.68 |
| E401 | EP604 | 11.73 |
| | EP603 | 12.34 |
| E401 | EP605 | 16.68 |
| E401 | EP601 | 19.77 |
| E401 | EP603 | 26.57 |
| E402 | EP604 | 14.21 |
| E402 | EP602 | 14.21 |

In the previous example, when member name *NO_ACCT* is selected as the blank member, the data row that shows Account equals *<blank>* and Product equals *EP603* will be populated to a cell with Account equals *NO_ACCT* and Product equals *EP603*.

Image: Sample cube

This example shows the sample cube.

| | | E401 | E402 | ACCOUNTS | NO ACCT | CUBT_ACCOUNTS |
|------------------------|------------------|--------|--------|----------|---------|---------------|
| High value products | CUBT_ALL_REGIONS | 118.63 | 158.77 | 277.4 | 0 | 277.4 |
| EP602 | CUBT_ALL_REGIONS | 109.97 | 115.53 | 225.5 | 11.58 | 237.08 |
| EP603 | CUBT_ALL_REGIONS | 63.64 | 121.69 | 185.33 | 12.34 | 197.67 |
| EP605 | CUBT_ALL_REGIONS | 87.73 | 68.57 | 156.3 | 0 | 156.3 |
| VAT free products | CUBT_ALL_REGIONS | 261.34 | 305.79 | 567.13 | 23.92 | 591.05 |
| TOP_NODE | CUBT_ALL_REGIONS | 379.97 | 464.56 | 844.53 | 23.92 | 868.45 |
| CUBT_EUROPEAN_PRODUCTS | CUBT_ALL_REGIONS | 379.97 | 464.56 | 844.53 | 23.92 | 868.45 |

Note that:

- If a dimension is already created from a static tree or query, entering a name for the blank member after inserting a tree or query to the dimension will not automatically add the blank member. In this case, you must manually add a member with the same name to indicate the member is a placeholder for blank member.
- Using the blank member in dimension is optional. If the blank member name is not defined and if data rows contain blank members in the dimension during data load, then those blank members are excluded from the dimension on the cube.
- Each dimension can have only one blank member name, so only one member is attached to the blank member name.
- Essbase Cube Builder applies the duplicate member rules when it uses the blank member name.
- The blank member name must be unique to the dimension.

Creating a Dimension from a Query Without the Blank Member Name Defined

To create a dimension from a query without the blank member name defined:

1. Select PeopleTools >Cube Builder >Dimensions to add a new dimension.
2. In the Dimension page, select the root node in the Dimension Structure section and click the Add Member Child icon.
3. Enter the name for the child member.
4. Select the member child and click the Assign From Query icon to populate the dimension from a dimension query.
5. In the Find Query page, select the Dynamic option to create a dimension from a dynamic query.
Alternatively, clear the Dynamic option to create a dimension from a static query.
6. Search for and select a query, and click the OK button.

7. In the Select Column page, select the column that contains the member name and its alias.
8. Click the OK button to populate the dimension with members from the query.

Note: The blank members from the dimension query are not listed in the dimension structure because the blank member name is not specified in the dimension.

Creating a Dimension from a Query With the Blank Member Name Defined

To create a dimension from a query with the blank member name defined:

1. Select PeopleTools >Cube Builder >Dimensions to add a new dimension.
2. In the Dimension page, select the root node in the Dimension Structure section, and click the Add Member Child icon.
3. Click the Blank Member link to access the Dimension Options page.
4. Enter a name for the blank member and click the OK button.
5. In the Dimension page, select the member child and click the Assign From Query icon.
6. In the Find Query page, select the Dynamic option to create a dimension from a dynamic query.
Alternatively, clear the Dynamic option to create a dimension from a static query.
7. Search for and select the dimension query.
8. In the Select Column page, select the column that contains the member name and its alias.
9. Click the OK button to populate the dimension with members from the query.

Note: The blank member from the dimension query is added to the dimension structure using the blank member name that you defined in the dimension.

Modifying the Blank Member Usage in an Existing Dimension with the Members Populated from a Static Query

If you add a blank member name to an existing dimension or remove a blank member name from an existing dimension that was created from a static query, the dimension query structure is not changed. To modify the blank member usage in a dimension that was created from a static query, you can use one of the following methods:

- Reload the dimension members from dimension query by deleting the member structure and then adding it again.
- Manually add the blank member to the dimension structure.

The name should be specified in the Blank Member option of the dimension.

- Manually remove the blank member from the dimension.

Removing a Blank Member from an Existing Dimension with the Members Populated from a Dynamic Query

To remove a blank member from an existing dimension with the members populated from a dynamic query:

1. Select PeopleTools >Cube Builder >Dimensions.
2. Open an existing dimension with the members from a dynamic query.
3. Click the Blank Member link in the Dimension page.
4. Clear the Blank Member field in the Dimension Options page, click the OK button, and save the dimension.
5. Reopen the dimension to see the changes.

The blank member is removed from the dimension structure.

Note: In a dimension with the members populated from a dynamic query, you must save and then reopen the dimension to view the changes of the blank member in the dimension structure.

Adding a Blank Member to an Existing Dimension with the Members Populated from a Dynamic Query

To add a blank member to an existing dimension with the members populated from a dynamic query:

1. Select PeopleTools >Cube Builder >Dimensions.
2. Open an existing dimension with the members from a dynamic query.
3. Click the Blank Member link in the Dimension page.
4. Enter a blank member name in the Blank Member field.
5. Click the OK button and save the dimension.
6. Reopen the dimension to see the changes.

The blank member is listed in the dimension structure.

Creating a Dimension Using a Tree and Without the Blank Member Name Defined

To create a dimension using a tree and without the blank member name defined:

1. Select PeopleTools >Cube Builder >Dimensions to add a new dimension.
2. In the Dimension page, select the root node in the Dimension Structure section and click the Add Member Child icon.
3. Enter a name for the member child and click the OK button.

The Dimension page reappears.

4. Click the Assign From Tree icon to populate the dimension from a tree.
5. In the Find Tree page, select the Dynamic option to create a dimension from a dynamic tree.

Alternatively, clear the Dynamic option to create a dimension from a static tree.

6. Select a tree and the top nodes that will be used to populate the dimension members.

- Click the OK button and save the dimension.

Note: The Dimension Structure section in the Dimension page does not list the blank members or leaves from the tree because the blank member name is not defined (or no blank member name is specified) in the dimension.

Creating a Dimension Using a Tree With the Blank Member Name Defined

To create a dimension using a tree with the blank member name defined:

- Select PeopleTools >Cube Builder >Dimensions to add a new dimension.
- In the Dimension page, click the Blank Member link.
- Enter a blank member name in the Dimension Options page and click the OK button.
- In the Dimension page, select the root node and click the Add Member Child icon.
- Enter a name for the member child and click the OK button.
- In the Dimension page, click the Assign From Tree icon to populate the dimension from a tree.
- In the Find Tree page, select the Dynamic option to create a dimension from a dynamic tree.

Alternatively, clear the Dynamic option to create a dimension from a static tree.

- Select a tree and top nodes that are used to populate the dimension members.
- Click the OK button to populate the dimension from the tree structure.

Note: The blank member from the tree is added to the dimension using the blank member name that you entered in the Dimension Options page.

Adding a Child or a Peer Member

Use the Child Member Add page (CUB_DIMENSION_SEC) on initial member creation to enter the initial dimension name and consolidation operator, and to add a new member that is a child of the currently highlighted tree member.

Navigation

Access the Child Member Add page by highlighting a tree member on the Dimension page and click the Add Member Child icon.

Image: Child Member Add page

This example illustrates the field and controls on the Child Member Add page.

The screenshot shows a dialog box titled "Child Member Add". It contains a label "Member:" followed by a text input field. Below the input field are two buttons: "OK" and "Cancel".

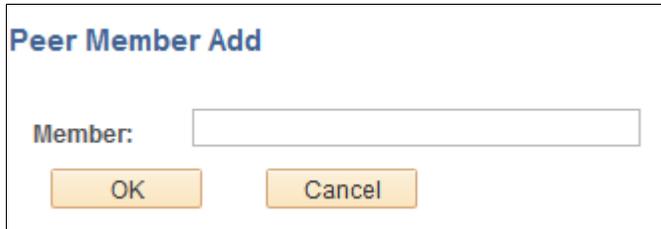
Use the Peer Member Add page to add a new member that is a peer of the currently highlighted tree member.

Navigation

Access the Peer Member Add page by highlighting a tree member on the Dimension page and click the Insert Member Peer icon.

Image: Peer Member Add page

This example illustrates the field and controls on the Peer Member Add page. Definitions for the field and controls appear following the example.



Member

Enter the name of the new member that will be a child or a peer of the currently highlighted member on the Dimension page.

OK

Click to add the child or a peer member and return to the Dimension page.

Inserting a Member Formula

Use the Member Formula page (CUB_FORMULA_SEC) to enter formula information for a dimension member.

Navigation

Access the Member Formula page by clicking the Formula link on the Dimension page.

Image: Member Formula page

This example illustrates the field and controls on the Member Formula page. Definition for the field appears following the example. Use the Formula text box to enter a formula that will be used to calculate aggregates for the specific member.



The image shows a dialog box titled "Member Formula". On the left side, there is a label "Formula:" followed by a large, empty rectangular text input area. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Example: Defining Essbase Dimensions

Suppose that you need to create a product dimension based on a table that contains setID-controlled products.

To create a product dimension, access the Dimensions page:

1. Select PeopleTools > Cube Builder > Dimensions.
2. Add a dimension with a name of PRODUCT.

Image: Example of the Dimension page when you first create the PRODUCT dimension

This is an example of the Dimension page when you first create the PRODUCT dimension.

At this point, the Dimension page:

- Displays dimensional properties such as dimension type, storage option, and so on.

Later on, depending on the dimension type, some of the tree object members may or may not have all of the available properties discussed in this book.

- Has an empty Dimension Structure section with a root node that has the same name as the dimension, PRODUCT.
- Has all icons at the bottom of the page appearing not active.

Icons become active as they become relevant when you build the dimensional structure. For example, when you highlight a tree member, the relevant action icons become available.

All actions in the icon area are applicable to whatever tree member is highlighted. For example, for dynamic trees and queries, you cannot delete any children.

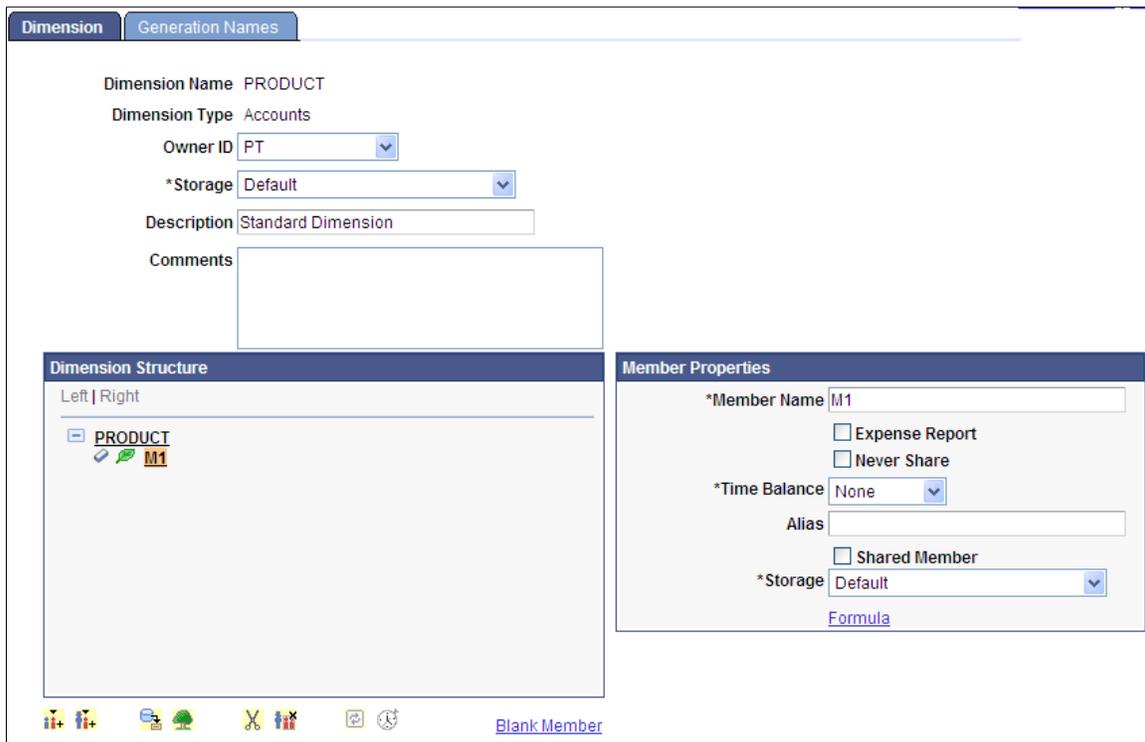
Note: To create dimension, you often add member children and peers manually and then supply the various attributes for each member. Parent members (those with children) may have attributes that leaf members (those without children) do not have, or vice versa.

To add a child member, highlight the PRODUCT node and click the Add Member Child icon. The Child Member Add page appears.

Enter the name for the child member and click the OK button. The Dimension page reappears.

Image: Example of the Dimension page with M1 member and its properties

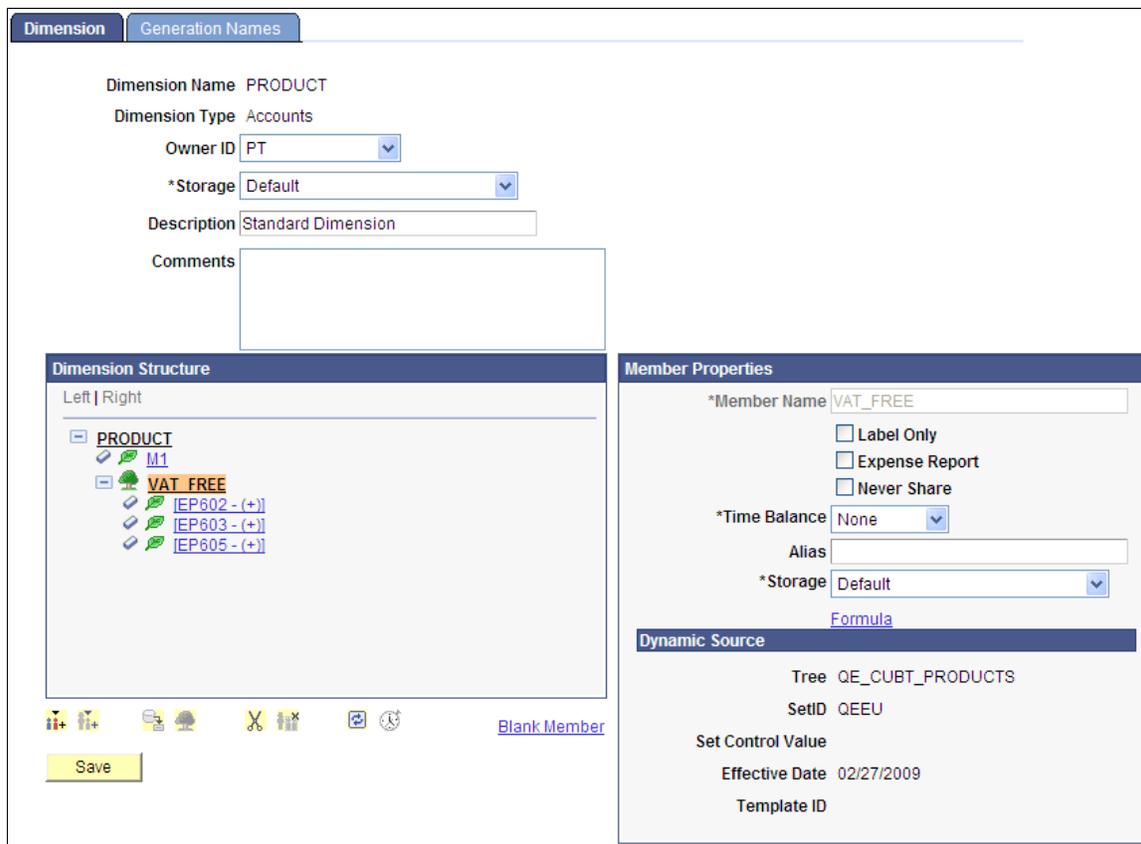
This example shows a manually entered child member called *M1* and the properties that are available to be set for that particular *M1* member.



Note: *M1* is highlighted and its corresponding attributes appear to the right of the tree. You can change any of these attribute values. After adding more members, you can view attributes of each member by selecting each new node. Selecting a node highlights it and displays the attributes for that node to the right of the dimension tree.

Image: Example of the attributes of the member VAT_FREE

This example illustrates the attributes of the member VAT_FREE, which shows that VAT_FREE member was built using a dynamic tree called QE_CUBT_PRODUCTS. All of the tree key field values appear in the Dynamic Source section of the attributes. The top portion shows the general relevant attributes for the current member node.



Inserting a Tree Structure into the Dimension

When inserting a tree structure into a dimension using the Find Tree page, you need to define whether that tree reference is dynamic or static. Note the following differences between dynamic and static (nondynamic) trees:

- Dynamic trees do not store dynamic referenced members in the dimension. They store only a reference to the tree definition.

When the dimension is sent to Essbase, the tree is reopened, read, and sent to Essbase when the cube needs to be populated.

If you select the Dynamic option on the Find Tree page, the Template ID field appears, enabling you to select a template.

You use templates to determine what each member of the dynamic tree should have as its defined properties. You create templates in their own component (PeopleTools > Cube Builder > Dynamic Dimension Templates). You are not required to use templates because the system uses a default template automatically if you do not provide a value in this field. Only members that do not match template values are actually stored as part of the dimension.

For example, if a tree has 50 members, the dimensional definition will not save 50 rows in the dimensional definition. Instead, it will store only the single row that references the tree name. From that point on, if you decide to add an alias for the 45th member, then the system stores only the 45th member and the tree reference member. Basically, the system stores only members for which you have changed the default values in the dimensional structure. For all other members, the assumption is that the member values match what is in the template for that dynamic reference or the default template that is used automatically. This technique saves storage space in the database.

- For static (nondynamic) trees, static references read the contents of the tree and store them within the dimension. All subsequent changes made to the tree are not reflected in the dimension. After the contents of the tree are read, the system does not save even the tree reference; it is as if you had manually entered all tree node and leaf values in the dimension.

Selecting a Tree

Use the Find Tree page (CUB_TREESEARCH_SEC) to search for and select a tree that is defined in the system.

Navigation

Access the Find Tree page by selecting a tree name on the Dimension page and clicking the Assign From Tree icon.

Image: Find Tree page

This example illustrates the fields and controls on the Find Tree page. Definitions for the fields and controls appear following the example.

The screenshot shows the 'Find Tree' interface. At the top, there is a 'Tree:' input field containing 'QE_C' and a checked 'Dynamic' checkbox. Below this is a 'Template ID:' input field with a search icon. A yellow 'Search' button is positioned below the Template ID field. The main part of the page is a table titled 'Available Trees' with a toolbar above it containing 'Personalize', 'Find', 'View All', 'First', '1-2 of 2', and 'Last'. The table has five columns: 'Select', 'Tree Name', 'SetID', 'Set Control Value', and 'Effective Date'. Two rows are visible in the table.

| Select | Tree Name | SetID | Set Control Value | Effective Date |
|--------------------------|------------------|-------|-------------------|----------------|
| <input type="checkbox"/> | QE_CUBT_PRODUCTS | QEEU | | 02/27/2009 |
| <input type="checkbox"/> | QE_CUBT_REGION | | | 03/01/2009 |

Dynamic

Select to define this tree reference as dynamic.

Deselect this option to define this tree reference as static.

Template ID

If the Dynamic option is selected (for dynamic trees), the Template ID field appears, enabling you to select a template.

If the Dynamic option is deselected (for static queries), after the contents of the query are read not even the query reference is saved.

See [Creating Dynamic Dimension Templates](#).

Available Trees

Displays a list of trees that are defined in the system after you click the Search button.

Selecting the Top Node

Use the Select Top Node page (CUB_TREENODE_SEC) to select an existing node of a highlighted tree.

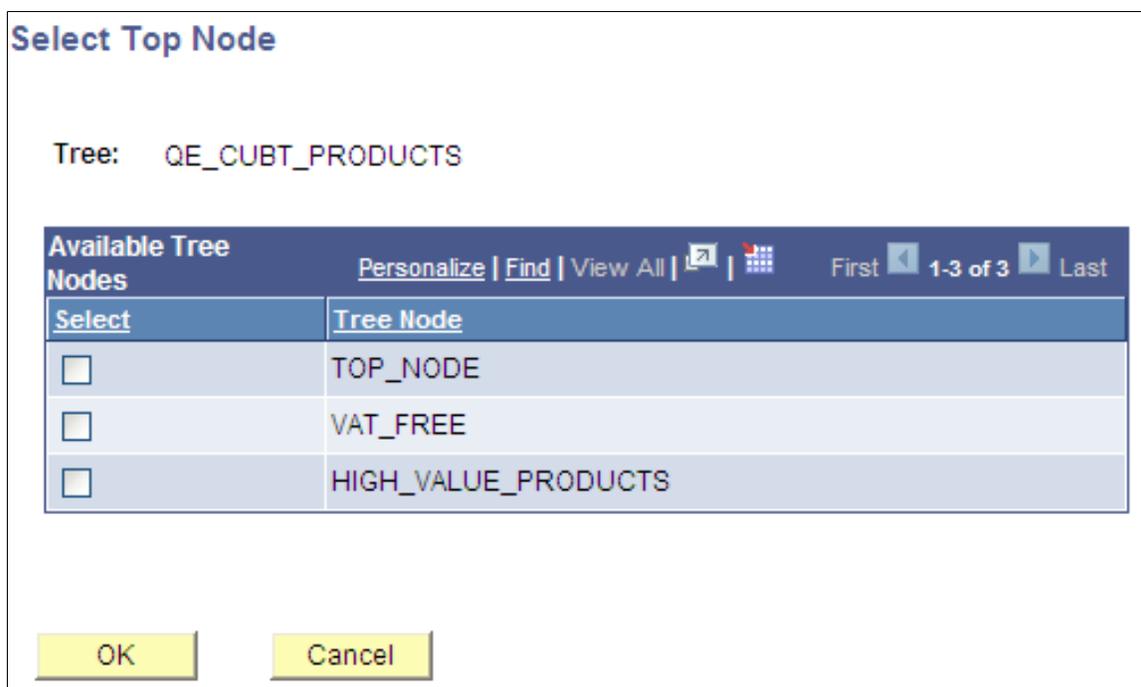
Navigation

Access the Select Top Node page by either:

- Selecting a tree name on the Dimension page and clicking the Assign From Tree icon, or
- Selecting trees on the Find Tree page and clicking the OK button.

Image: Select Top Node page

This example illustrates the fields and controls on the Select Top Node page. Definitions for the fields and controls appear following the example.



Available Tree Nodes

This section displays a list of all the nodes from the selected tree.

Select

Select one tree node as a starting point in the tree definition to use when copying the structure into the current dimension.

Note: If the actual top node of the tree is selected, then the entire tree will be copied.

OK

After selecting the starting node, click the OK button to return to the Dimension page.

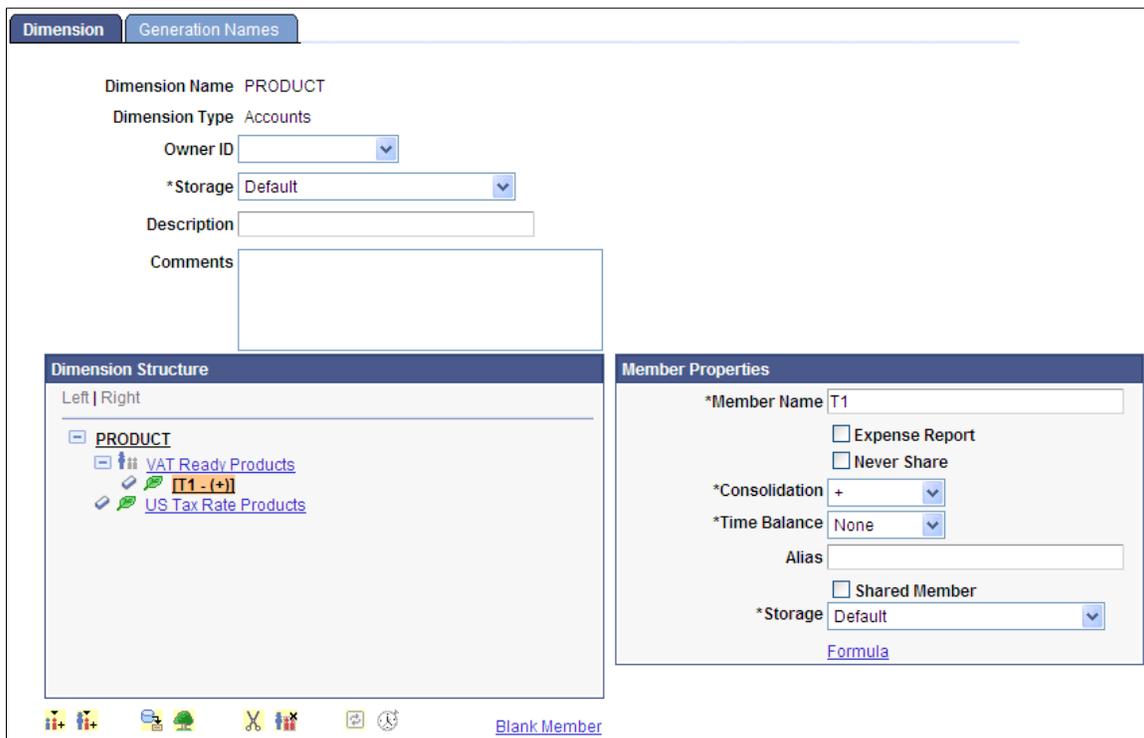
The tree structure is read, and the contents of the tree are placed in the current node of the dimension.

Example: Inserting a Tree Structure into the Dimension

The Dimension page has two nodes, *VAT Ready Products* and *US Tax Rate Products*, that were added manually under the **PRODUCT** node.

Image: Example of the Dimension page with two nodes added under the **PRODUCT** root node

This example illustrates the Dimension page that has two nodes, *VAT Ready Products* and *US Tax Rate Products*.

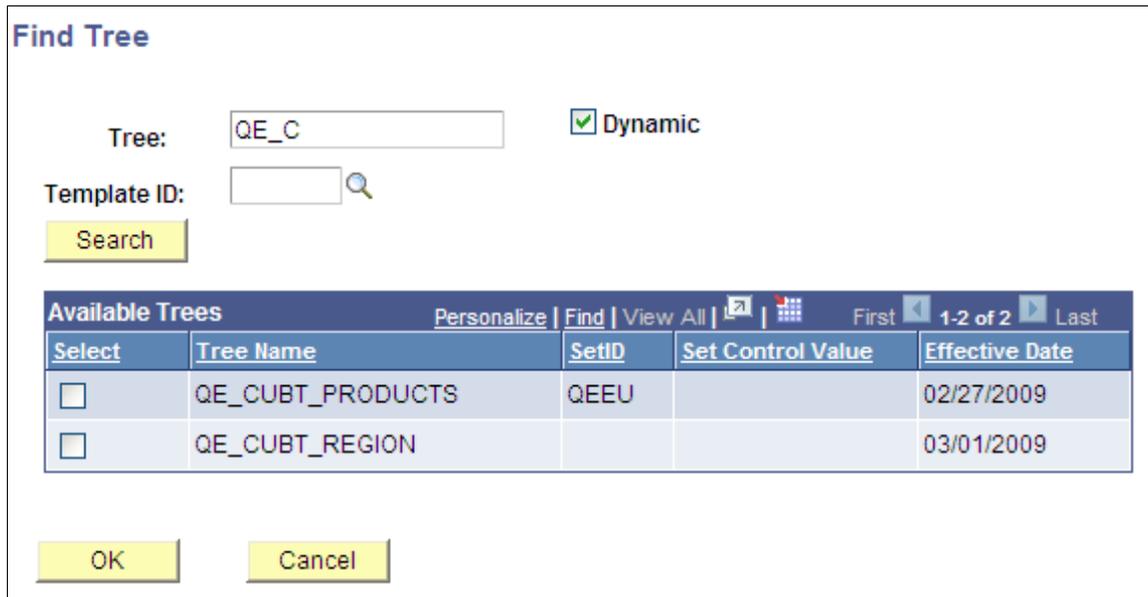


When you highlight the **PRODUCT** root node, the Add Member Child button below the tree becomes active. Subsequently, below the *VAT Ready Products* member, a child member was added with the name *T1*. When the *T1* child member is highlighted, the Assign From Tree icon became active; click it to connect with a tree structure at that point in the dimension tree.

Click the Assign From Tree icon to access the Find Tree page, where you can select a tree.

Image: Example of the Find Tree page

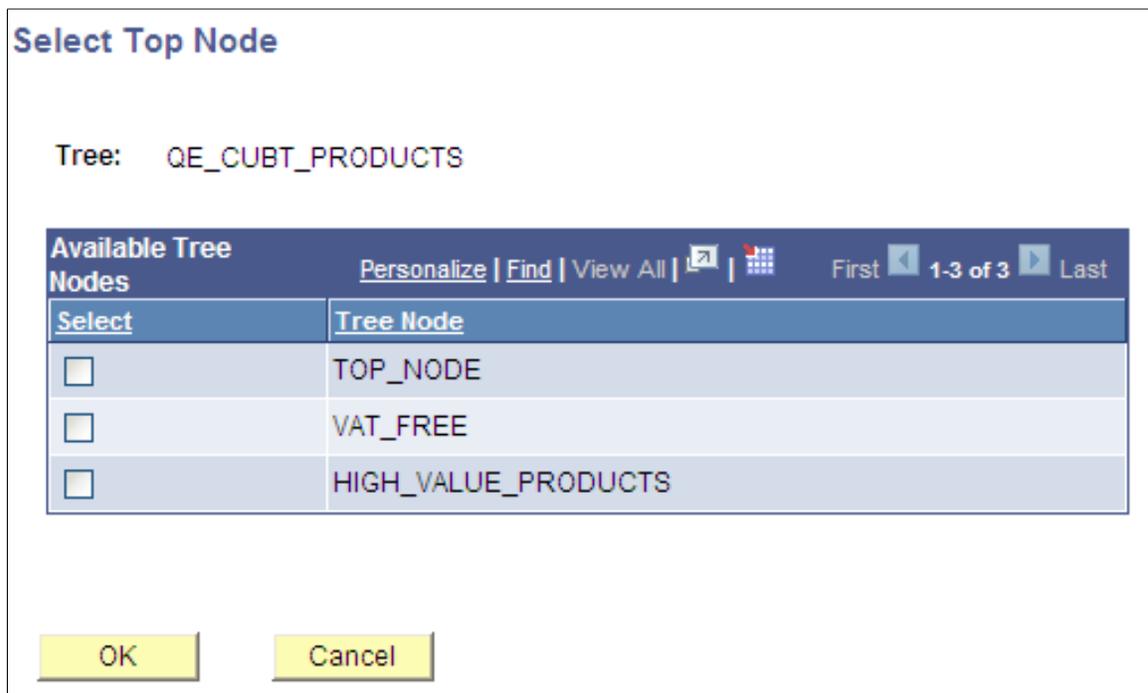
This example illustrates the fields and controls on the Find Tree page.



You need to define whether this tree reference is dynamic or static, select a tree name, and click the OK button. The Select Top Node page appears with the Available Tree Nodes section listing all the nodes from the selected tree.

Image: Example of the Select Top Node page

This example illustrates the fields and controls on the Select Top Node page.



You need to select a starting point in the tree definition and click the OK button to return to the Dimension page. Note that the tree structure is read and the contents of the tree are placed in the current node of the dimension.

Because VAT_FREE was selected as the starting node, that member is placed at the starting point and all of the children of that node are read, after which all of them become members of the dimension.

Image: Example of the Dimension page with contents of the tree added in the VAT_FREE node

This example shows the PRODUCT dimension after the contents of the tree definition are copied.

The screenshot displays the 'Dimension' configuration page for the 'PRODUCT' dimension. The 'Dimension Name' is 'PRODUCT', the 'Dimension Type' is 'Accounts', and the 'Owner ID' is 'PT'. The '*Storage' is set to 'Default' and the 'Description' is 'Standard Dimension'. The 'Comments' field is empty.

The 'Dimension Structure' pane shows a tree view with the following nodes:

- PRODUCT
 - VAT Ready Products
 - VAT FREE - (+)** (highlighted with a tree icon)
 - EP602 - (+)
 - EP603 - (+)
 - EP605 - (+)
 - US Tax Rate Products

The 'Member Properties' pane for the selected member 'VAT_FREE' shows:

- *Member Name: VAT_FREE
- Label Only
- Expense Report
- Never Share
- *Consolidation: +
- *Time Balance: None
- Alias: (empty)
- *Storage: Default
- Dynamic Source:
 - Tree: QE_CUBT_PRODUCTS
 - SetID: QEEU
 - Set Control Value: (empty)
 - Effective Date: 02/27/2009
 - Template ID: (empty)

At the bottom of the 'Dimension Structure' pane, there are icons for adding, deleting, and refreshing members, along with a 'Blank Member' link and a 'Save' button.

In this example, because you selected a dynamic tree, a tree icon is present in the dimension member that references the tree definition. The three items that were read are not actually stored as part of the dimension, but they are read every time you view the dimension. That is, after you save this dimension to the database, if the tree is modified to have more nodes or leaves, then the next time you view the dimension it will display the contents of the tree with the modifications. Note that this process does not happen with static tree members.

For a static tree member, no reference is made to the tree that you used during the initial creation of the members and no tree icon appears to indicate that those members were originally created from a tree definition.

Inserting a Query into the Dimension

As with trees, queries can also be used to populate a dimension. Queries also use templates. While assigning members from query definitions, you need to supply prompt values, if any, that are required by the query.

For dynamic queries, as with dynamic trees, you do not actually store query results in the dimension definition, except for those that do not match any assigned template for the query (or the default template

if no actual template was assigned). At the time the dimension is sent to the Essbase database, the query will be rerun to reflect the current result rows from the query.

For static queries, you take the query results and store all of those rows within the dimension.

Note: When you create dimensions, you may have issues with queries that return a large number of rows. When a query returns thousands of rows, it may cause PIA to crash or it may cause the dimension builder to have long waiting times to display the results in the dimensional tree. If you have issues with large results query, use dynamic queries. By using dynamic queries, you ensure that only a certain number of rows appear at one time within the dimensional structure.

Selecting a Query

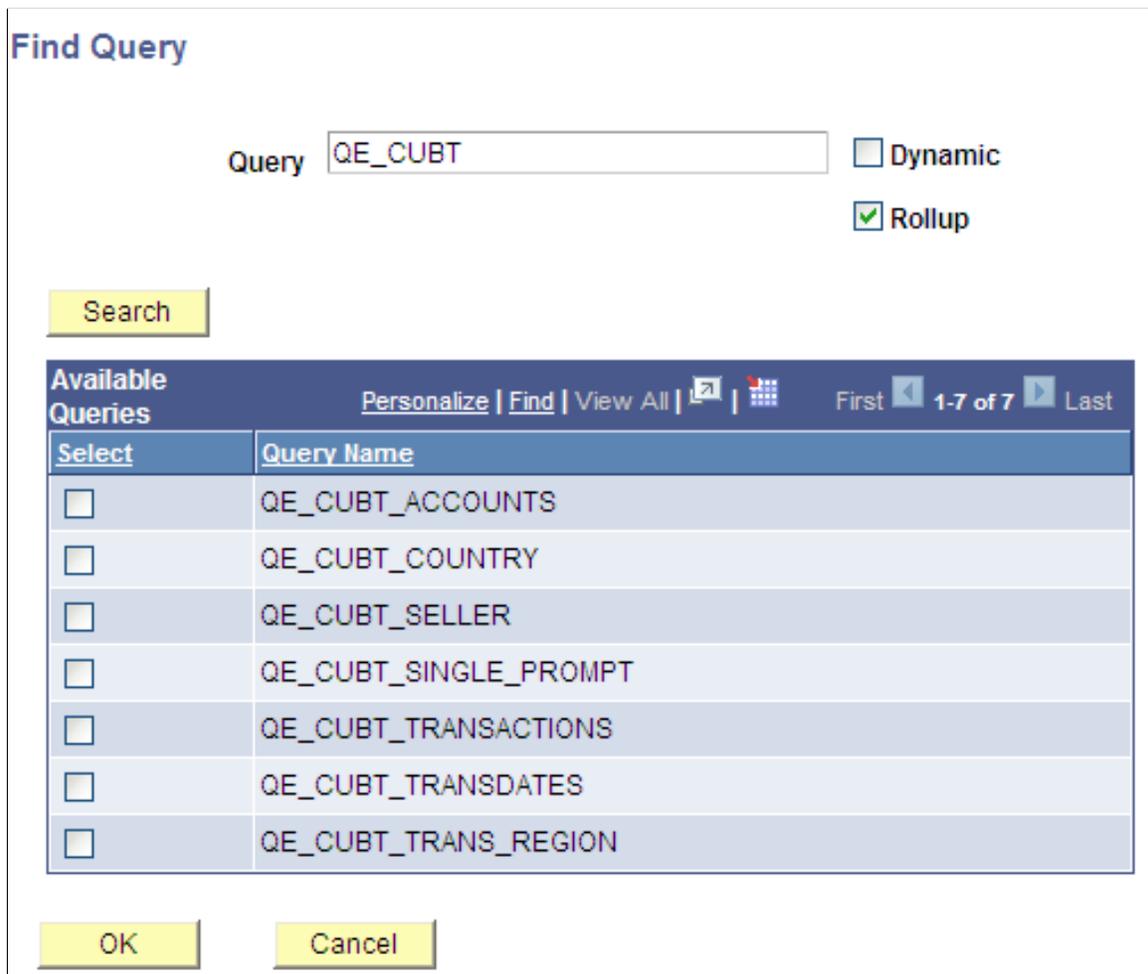
Use the Find Query page (CUB_QRYSEARCH_SEC) to select a query that is defined in the system.

Navigation

Access the Find Query page by highlighting a tree member on the Dimension page and clicking the Assign From Query icon.

Image: Find Query page

This example illustrates the fields and controls on the Find Query page.



Dynamic

Select to define this query reference as dynamic.

Deselect this option to define this query reference as static.

Rollup

Select to create a multi-level dimension structure using a single dimension query.

Note: You do not need to create multiple queries to populate members on each level.

Query results must be sorted properly to be used as wide query when you are creating members for dimension.

Duplicate member names are still not allowed; for example, no duplicate member names are allowed at the lowest level of a dimension, so confirm that your query results do not return duplicate values for the column selected at the lowest level.

Template ID

If the Dynamic option is selected (for dynamic queries), then the Template ID field appears enabling you to select a template. Optionally, assign a defined template to the query.

If the Dynamic option is deselected (for static queries) after the contents of the query are read, not even the query reference is saved.

See [Creating Dynamic Dimension Templates](#).

Available Queries

After you click the Search button, select a query from a list of queries that are defined in the system.

Note: Only public queries to which the current user has authorization appear in this section.

Selecting a Column

Use the Select column page (CUB_QRYCOLUMN_SEC) to select an existing column of a highlighted query. The columns displayed are from the select statement of the query.

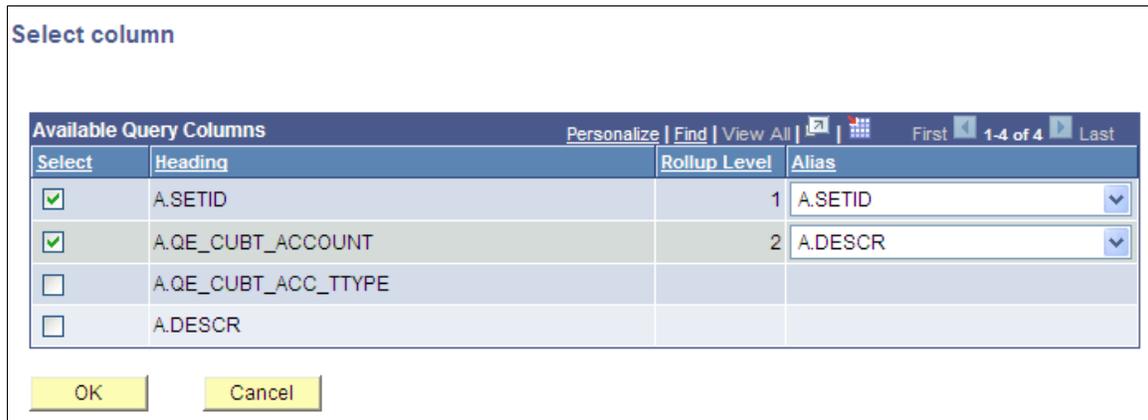
Navigation

Access the Select column page by either:

- Highlighting a tree member on the Dimension page and clicking the Assign From Query icon, or
- Selecting queries on the Find Query page and clicking the OK button.

Image: Select column page

This example illustrates the fields and controls on the Select column page.



Note: If you are creating a multi-level dimension, you are able to define the dimension structure as well as a member alias for each level.

Available Query Columns

Select columns from a list of columns that are defined in the system.

Note: You are able to select more than one columns from the query. Each column represent a different level on the dimension.

Rollup Level

If you select the Rollup option in the Find Query page, the Rollup Level column appears displaying the hierarchical structure of the dimension and the query column that will be used to populate the dimension level.

Alias

Optionally, select one or more columns from the query to be used to populate the alias for the selected member.

Note: You do not need to manually enter the alias for each member; but if you want to enter the alias manually, you should not select any value for the Alias drop-down list.

OK

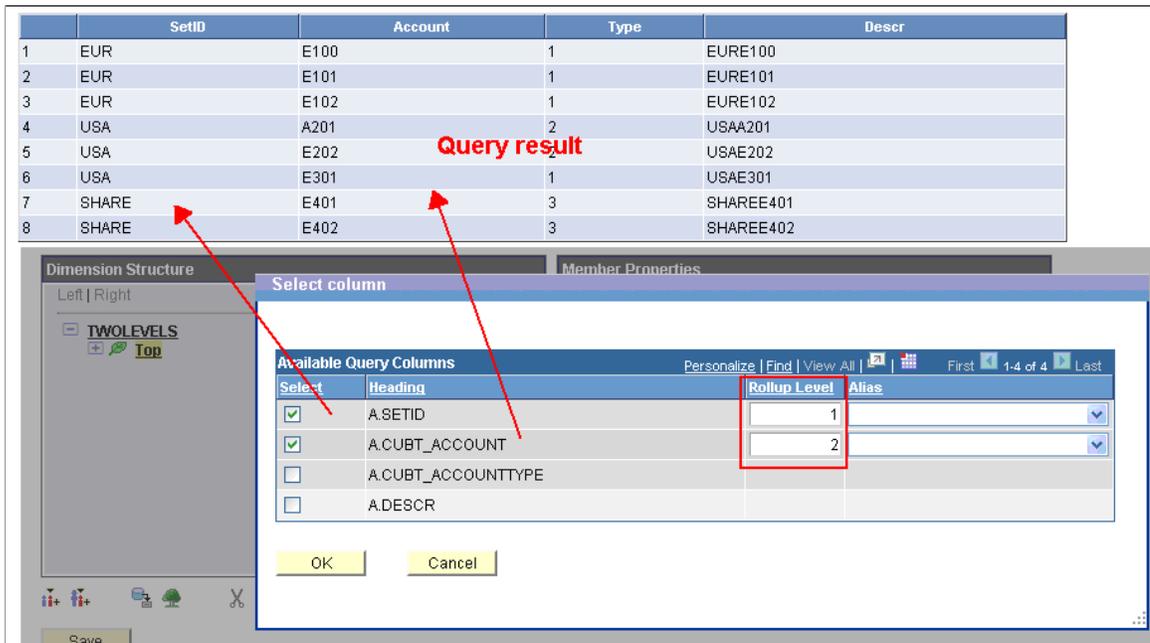
Click to return to the Dimension page.

The following examples demonstrate the use of a wide query and rollup levels that define the hierarchical structure of a dimension:

- Use the Select Column page to set the rollup levels:

Image: Setting the rollup levels

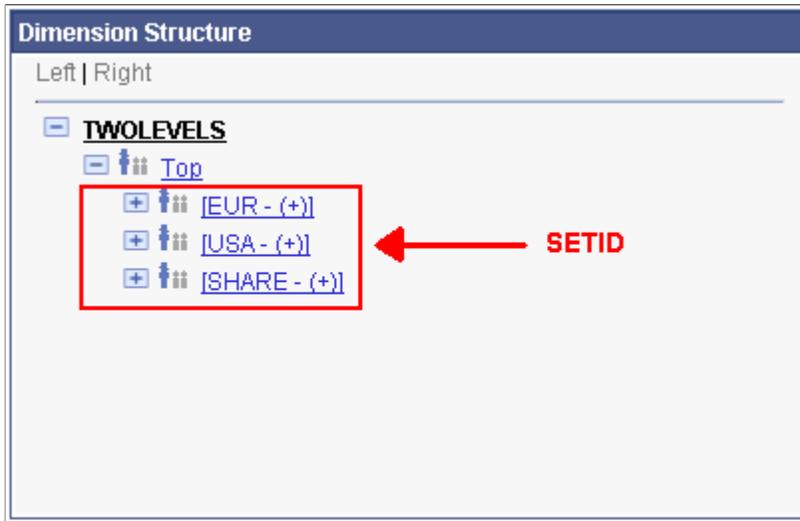
This example illustrates the setting of the rollup levels.



- After you select columns and click the OK button in the Select Column page, the first level under the selected member is populated using the column that was designated as Rollup Level 1.

Image: Example of rollup level 1

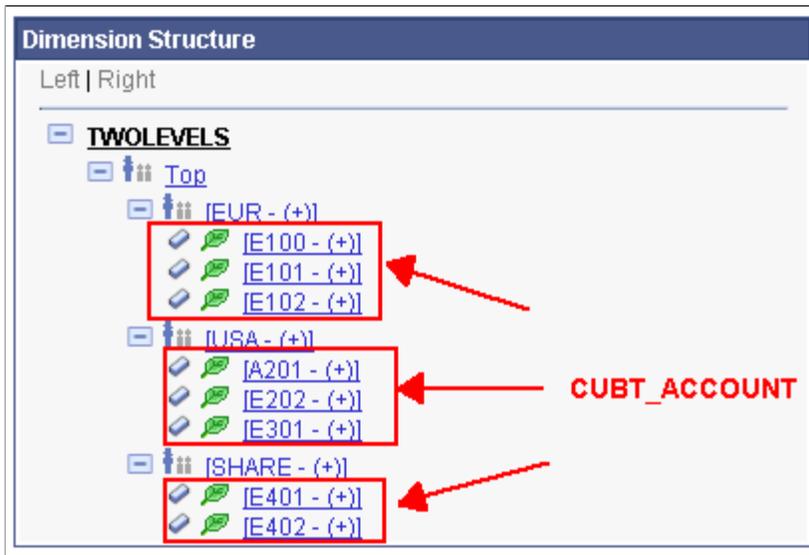
This is an example of rollup level 1.



- The second level is populated using the column designated as Rollup Level 2.

Image: Example of rollup level 2

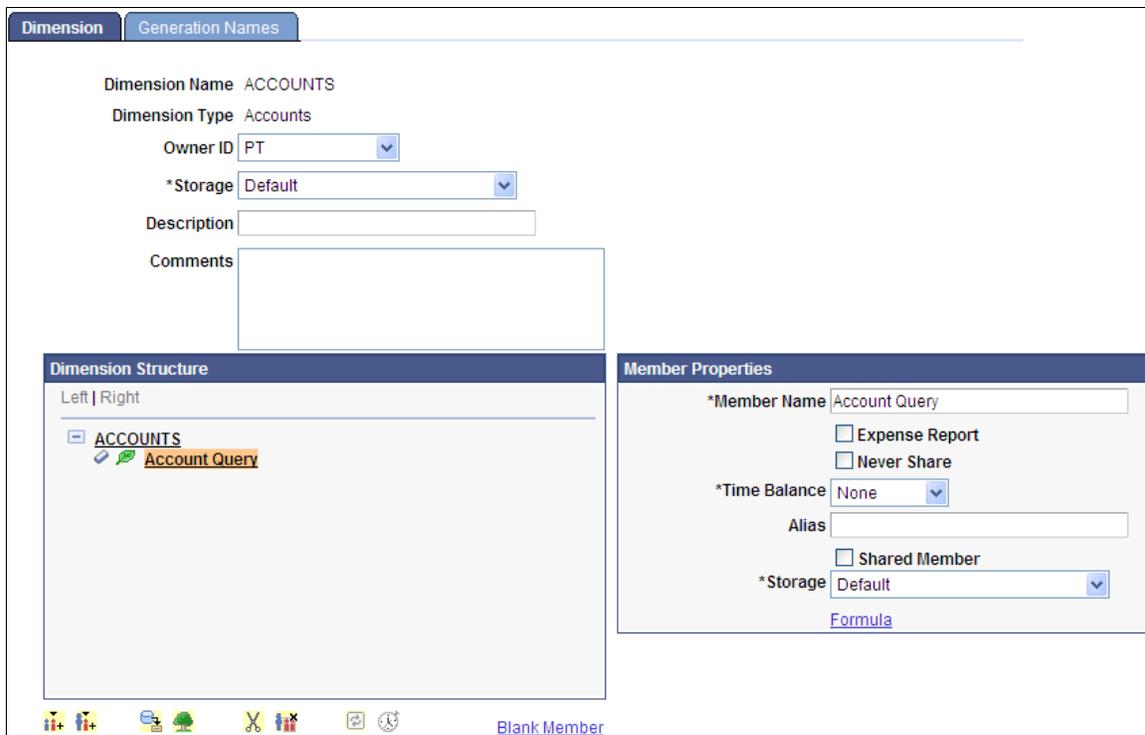
This is an example of rollup level 2.



Example: Inserting a Query into the Dimension

Image: Example of the Dimension page with Account Query member

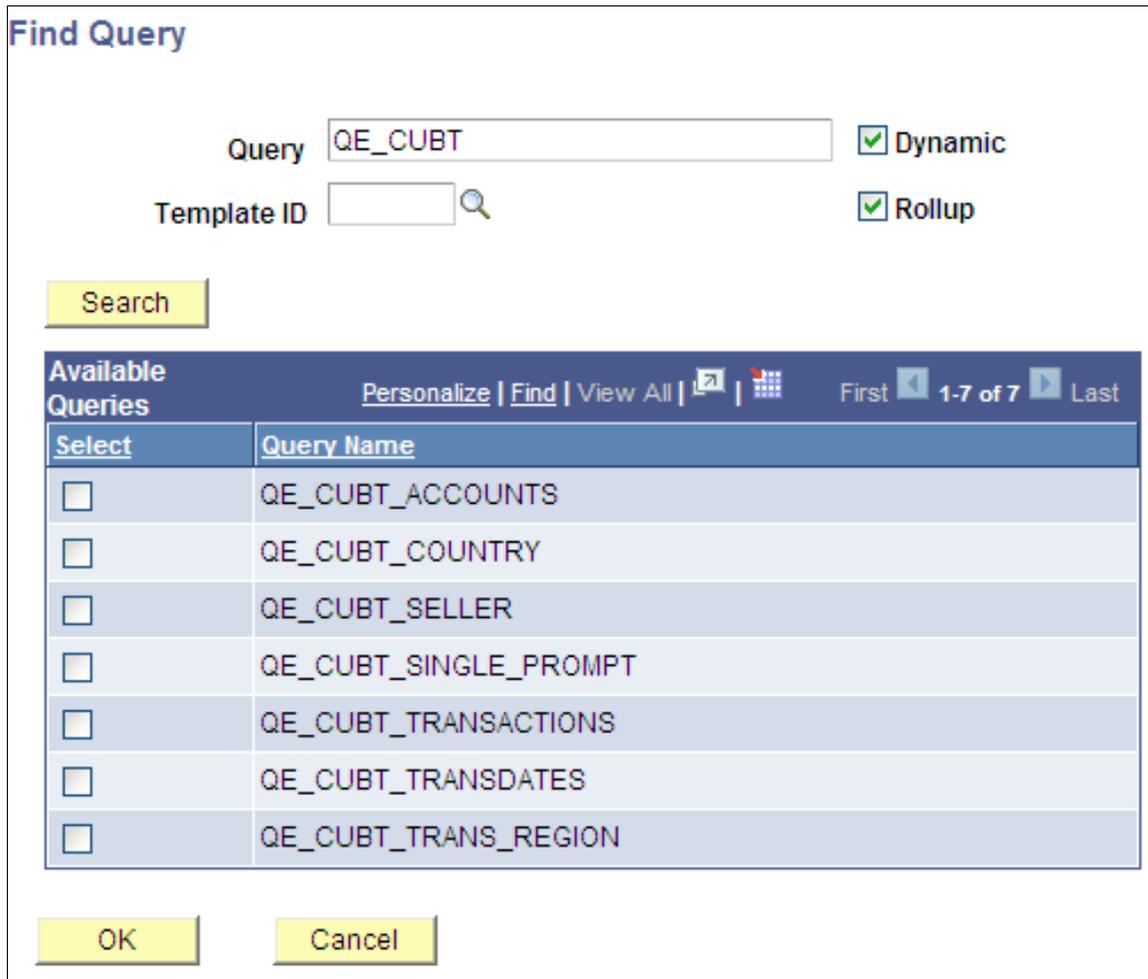
This example illustrates a child member of the root node called *Account Query* that was added manually.



Highlight the *Account Query* member and click the Assign From Query icon, the Find Query page appears.

Image: Find Query page

This example illustrates the fields and controls on the Find Query page.



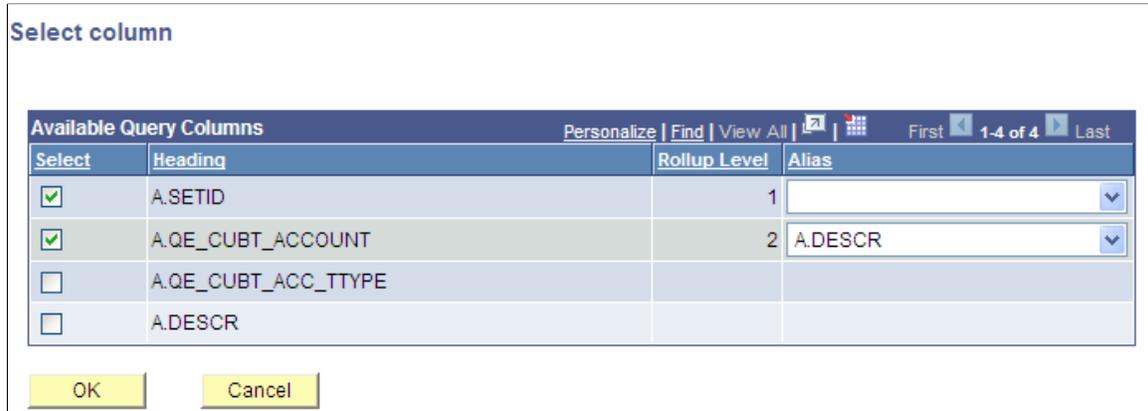
You can search a partial name query. Note that the system displays only public queries to which the current user has authorization. In this example, you select the QE_CUBT_ACCOUNTS query and click the OK button. You also can assign a defined template. In this example, the results of the query yield leaf members, so use the query dynamically.

If the selected query has a prompt criteria defined, then a prompt page appears enabling you to enter values to satisfy the prompt requirement for the query.

The last step is to select a query column to fill the dimension structure using the Select column page. In this scenario, you select the A.SETID and A.QE_CUBT_ACCOUNT columns as the desired fields. You also select the A.DESCR option as alias value for the A.QE_CUBT_ACCOUNT column

Image: Example of the Select column page

Example of the Select column page, A.SETID and A.QE_CUBT_ACCOUNT as desired fields, DESCR column as aliases.

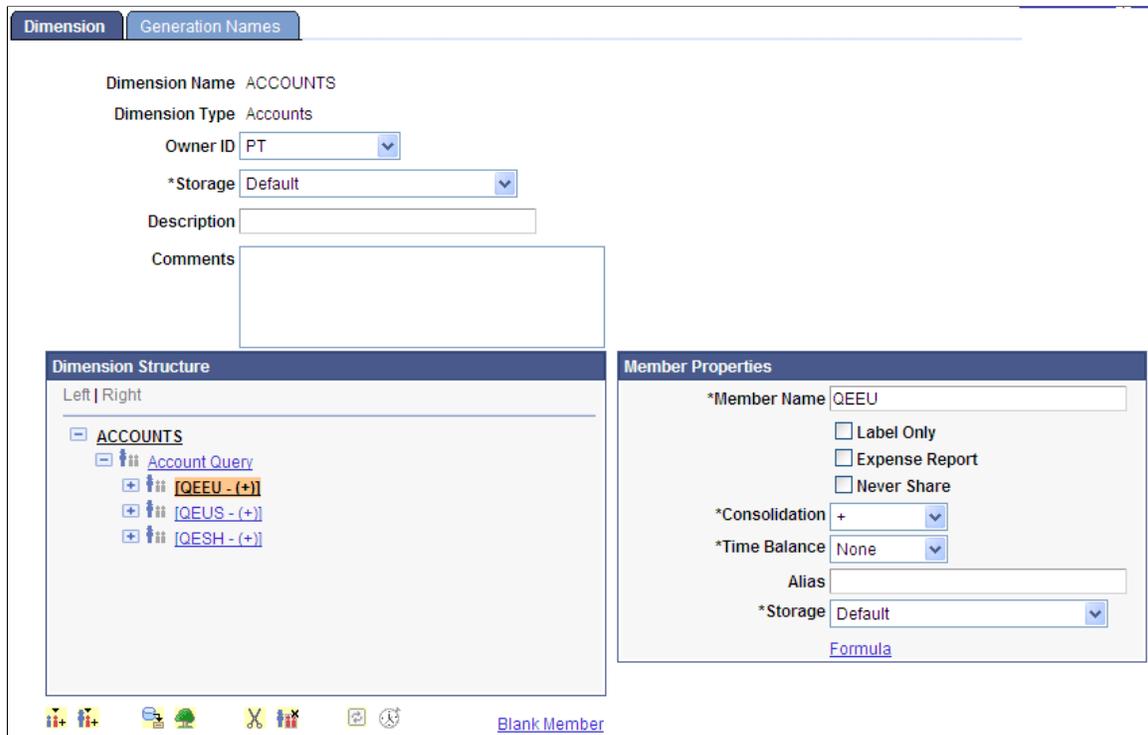


Note: If you are creating a multi-level dimension cube, you are able to define the dimension structure as well as a member alias for each level.

After selecting the desired values, click the OK button to run the query. The results appear as part of the dimensional structure.

Image: Example of the Dimension page with results displayed as part of the dimensional structure

This example illustrates the Dimension page with results displayed as part of the dimensional structure.



Note: The results of running this query should yield unique values for the leaf members. All leaf values under the same parent must always be unique for the dimension to be valid.

A query icon is in the member that references the dynamic query. If the query was not run as dynamic, then a normal parent icon exists and—from that point on—no indication exists that the members were added as part of running a query. As with static trees, it is as if you had manually typed in all the members.

Defining Time Dimensions

Use the Time Dimension Wizard page (CUB_TIMEDIMWIZ) to define the date and time ranges of the time dimension, and to automatically populate the members of a time dimension.

Navigation

Access the Time Dimension Wizard page by clicking the Time Dimension Wizard icon on the Dimension page.

Note: The Time Dimension Wizard icon is available only for time dimensions and not for other dimension types.

Image: Time Dimension Wizard page

This example illustrates the fields and controls on the Time Dimension Wizard page. Definitions for the fields and controls appear following the example.

- First Day of the Week** Select the first day of the calendar week on which the intended calendar starts.

- Begin Date** Select the begin date for the time range for which members are created.

- End Date** Select the end date for the time range for which members are created.

- Time Depth** Select the option next to a hierarchy level to include it in the hierarchy.

Note: Shaded levels cannot be used in combination with other selections. For example, Semester and Trimester hierarchies are mutually exclusive. Time-depth selections define the members that are created in the time dimension.

Year Semantic Rules

Select to define the month and day when semantic rules start the year.

Time Dimension Members

Time Dimension members are created as manual members (not dynamic) and they are in PeopleSoft Date-Time format, which is compatible with Essbase format. However, you should create the data query sources in a standard PeopleSoft Date-Time format as well; otherwise, the data-load process will fail because it will not recognize matching members.

During the data-load process, Essbase verifies that the passed cells match the members that are already in the dimension. For example, if the time members for the dates of a year were created in the dimension with format *YYYY-MM-DD* and the quarters in the data source query are set as *mm/dd/yy*, then Essbase returns an error message to inform you that the members were not encountered in the time dimension.

On the Time Dimension Wizard page, after you select the time dimension and click the Create Time Dimension button, the Time Dimension members appear on the Dimension page.

Image: Dimension page, Dimension Structure section

This example illustrates the fields and controls on the Dimension page, Dimension Structure section.



Note: To prevent date-time format conflicts, Essbase eliminates previously created members before it automatically displays new members.

You also can add new members to the time dimension posterior to the automatic population of members either by PeopleSoft trees, PeopleSoft queries, or manually. However, the members must be in the same PeopleSoft date-time format to be consistent with the automatically populated members.

Adding Generation Names

Use the Generation Names page (CUB_DIMENSION2) to enter the generation names for the Essbase dimension.

Navigation

Access the Generation Names page by selecting the Generation Names tab on the Dimension page.

Image: Generation Names page

This example illustrates the fields and controls on the Generation Names page. Definitions for the fields and controls appear following the example.

| Dimension | | Generation Names | |
|------------------|-----------------|------------------|---|
| Dimension: | PRODUCT | | |
| Generation Names | | | |
| Generation | Generation Name | | |
| 1 | WORLD | + | - |
| 2 | CONTINENT | + | - |

Generation

Displays the generation of the dimension.

The top parent of a dimension is called generation 1, each child of generation 1 is called generation 2, each child of generation 2 is called generation 3, and so on.

For example, as shown on the Generation Names page, the PRODUCT dimension has two generation names: WORLD and CONTINENT. These names are not relevant to the graphical user interface (GUI), but they can be used on the Essbase side when Essbase references an entire generation in a member formula.

Note: In Essbase terminology, a generation refers to a level within a dimension. A root branch of the tree is generation 1.

Generation numbers increase as you count from the root toward the leaf member.

Level also refers to a branch within a dimension; however, levels reserve the numerical ordering used for generations.

Generation Name

Enter the generation name for each generation.

Defining Essbase Cube Outlines

Defining Cube Outlines

This topic discusses how to add cube dimensions and cube data queries to cube outline.

Adding Cube Dimensions to a Cube Outline

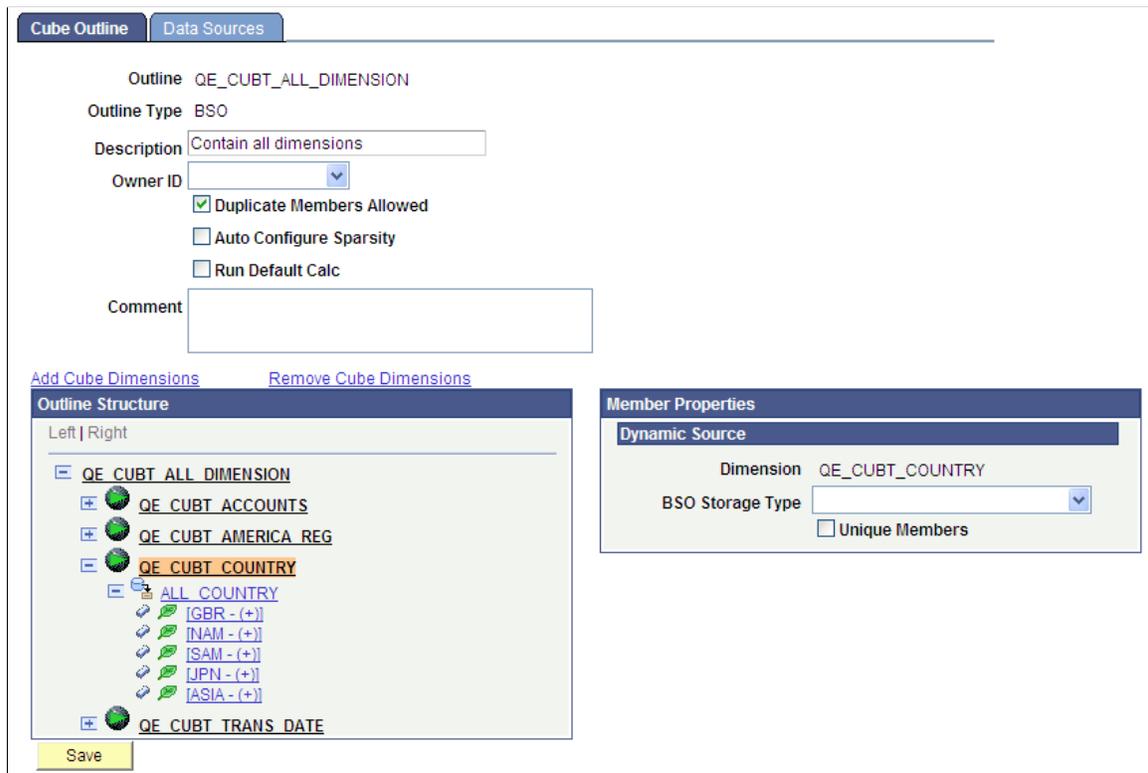
Use the Cube Outline page (CUB_OUTLINE) to add dimensions to the current outline.

Navigation

PeopleTools > Cube Builder > Outlines.

Image: Cube Outline page

This example illustrates the fields and controls on the Cube Outline page. Definitions for the fields and controls appear following the example.



Note: Depending on the outline type, some of the outline may not have all of the available properties or fields that are discussed in this topic.

| | |
|---|--|
| Outline Type | <p>Displays the type of outline. Available types are:</p> <ul style="list-style-type: none"> • <i>ASO</i>: aggregate storage. • <i>BSO</i>: block storage. |
| Description | Enter the description that you want to associate with the cube outline. |
| Owner ID | <p>Select the owner of the cube outline.</p> <p>The owner ID helps to determine the product, feature, or application team that last made changes to this outline.</p> |
| Duplicate Member Allowed | <p>Select to indicate that duplicate member names are allowed in the dimensions.</p> <hr/> <p>Note: To support duplicate members, users must create a data source query using the fully qualified name of each member; otherwise, the Essbase server will not be able to identify the member to load data into. See Essbase documentation for more information about Fully Qualified Name Members.</p> <hr/> |
| Auto Configure Sparsity | <p>Select to enable Essbase to automatically set sparsity levels.</p> <p>Deselect to manually set the sparsity levels. After you deselect this option, you can set the sparsity of each dimension in the template to either <i>Sparse</i> or <i>Dense</i>.</p> <hr/> <p>Note: Dimensions marked as <i>Sparse</i> might take up significantly more space in the resulting cube than those marked as <i>Dense</i>. If you mark every dimension as sparse, you may end up with an enormous cube that is not practical or even possible to store on even the most powerful server. That is why the Time and Account dimensions are <i>Dense</i> level by default. You can change these settings in Essbase after you create your cube. When you change the sparsity settings from Cube Builder, be careful when adjusting them and refer to the Oracle Essbase documentation.</p> <hr/> |
| Run Default Calc (run default calculation) | <p>Select this option to have the system run the default calculation script in Essbase when the cube is created; Essbase aggregates the tree in the given rollups, but it does nothing else.</p> <p>Deselect this option to create a custom calculation script.</p> |
| Add Cube Dimensions | <p>Click to access the Add Cube Dimensions page, where you can add dimensions to the current outline.</p> <hr/> <p>Note: A dimension can be used only once in an outline.</p> <hr/> |
| Remove Cube Dimensions | Click to access the Current Dimension List page, where you can remove dimensions from the current outline. |

Outline Structure

In this section, each dimension is highlighted by a bright green icon. The dimension contents are shown in the outline.

Member Properties

Located at the right of the tree, this section displays the corresponding attributes.

Note: You can change these values using the Dimension page.

Dynamic Source

This section appears only for dynamic queries.

It displays only a frame size of members with general information about the contents of the dimension.

Inserting Cube Dimensions

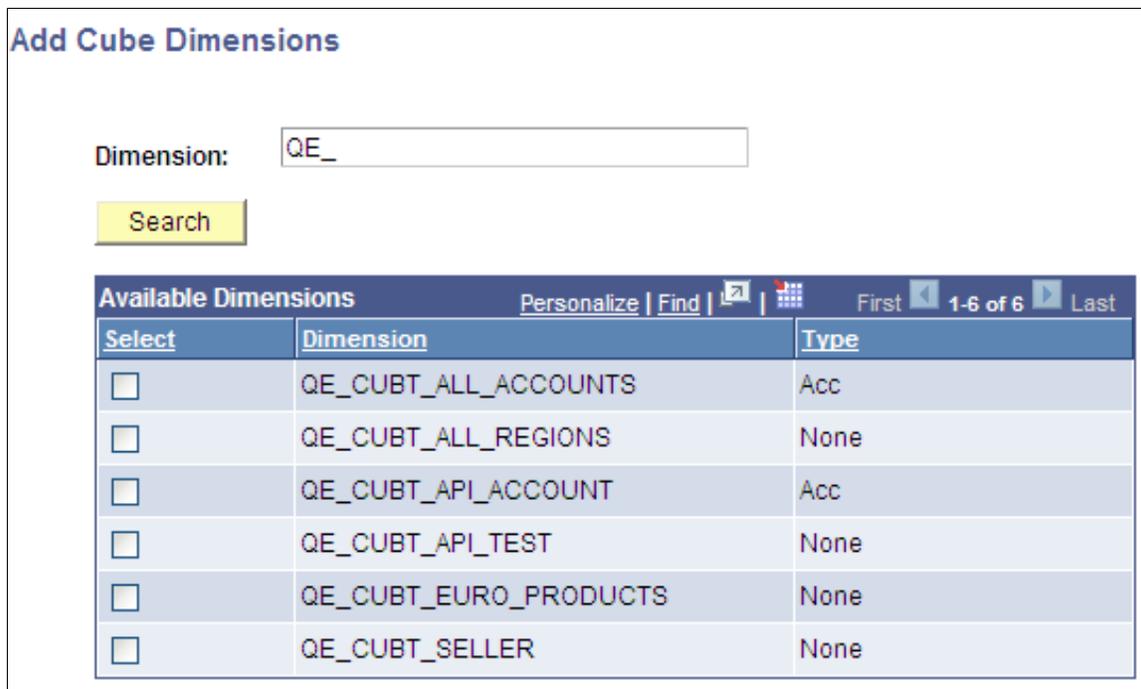
Use the Add Cube Dimensions page (CUB_OUTLINE_SEC) to select dimensions to add to the current outline.

Navigation

Access the Add Cube Dimensions page by clicking the Add Cube Dimensions link on the Cube Outline page.

Image: Add Cube Dimensions page

This example illustrates the fields and controls on the Add Cube Dimensions page. Definitions for the fields and controls appear following the example.



Search

Click to display all dimensions that are not already inserted into the dimension.

Available Dimensions

Select a dimension and click the OK button to return to the Cube Outline page.

Deleting Cube Dimensions

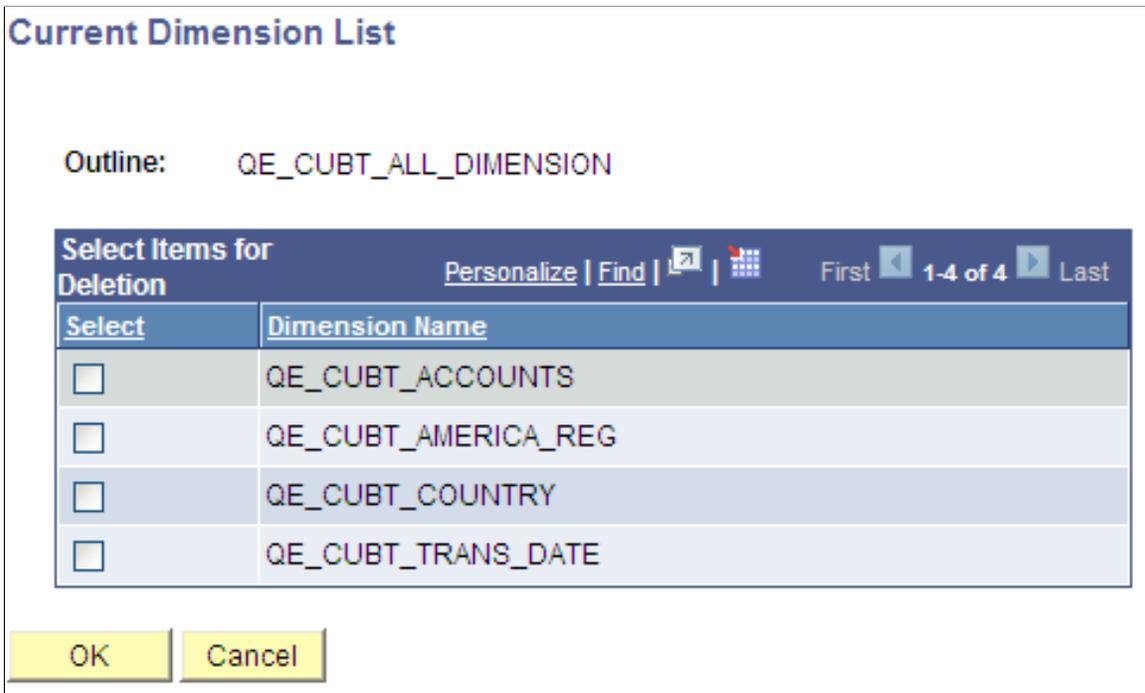
Use the Current Dimension List page (CUB_OUTDIM_SEC) to remove dimensions from the current outline.

Navigation

Access the Current Dimension List page by clicking the Remove Cube Dimensions link on the Cube Outline page.

Image: Current Dimension List page

This example illustrates the fields and controls on the Current Dimension List page. Definitions for the fields and controls appear following the example.



Select Items for Deletion

Select a dimension that you want to delete from the cube outline.

OK

Click to delete the selected dimension and return to the Cube Outline page.

Example: Adding Cube Dimensions to a Cube Outline

Access the Outlines component by selecting PeopleTools > Cube Builder > Outlines.

Image: Cube Outline page

This example shows the Cube Outline page with a new cube outline named OUTLINE_CUBE.

The screenshot displays the 'Cube Outline' configuration page. At the top, there are two tabs: 'Cube Outline' (selected) and 'Data Sources'. The main content area shows the following fields and options:

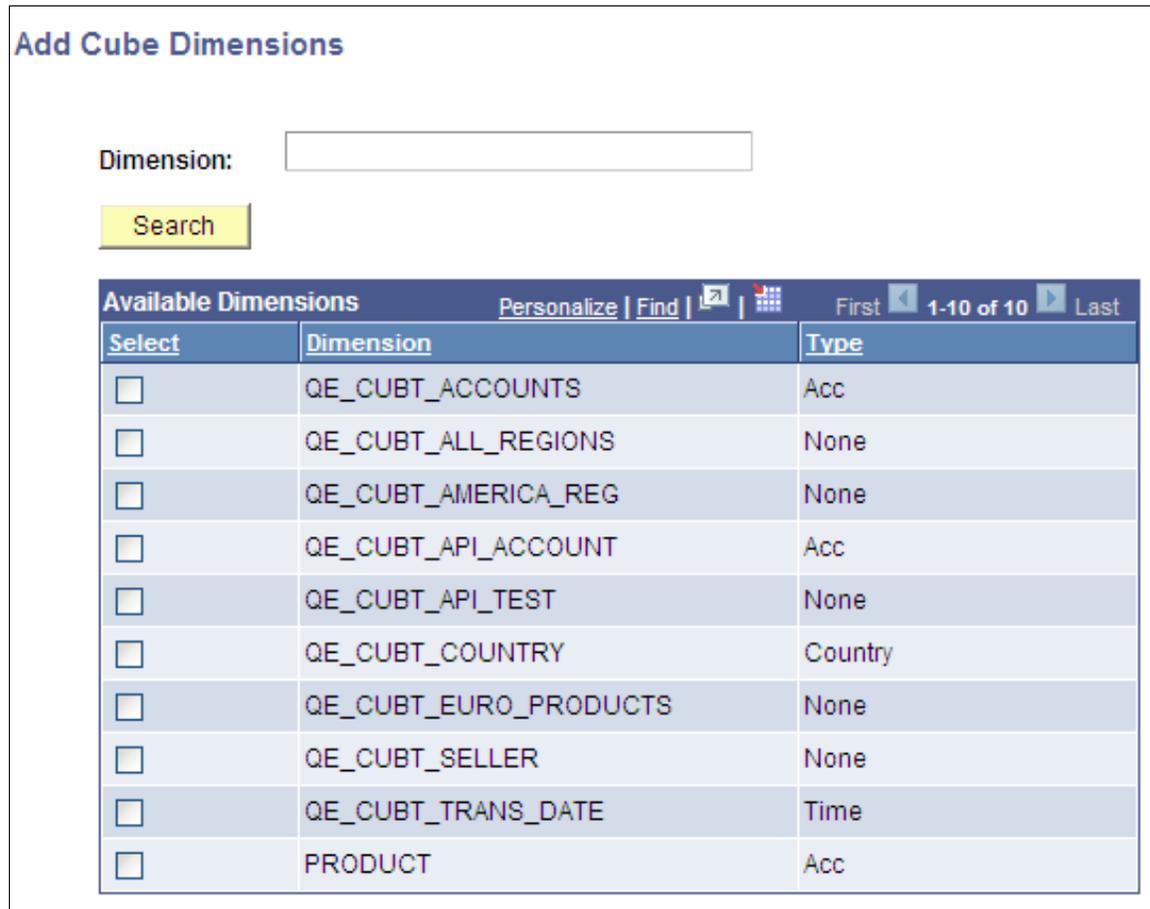
- Outline:** OUTLINE_CUBE
- Outline Type:** BSO
- Description:** An empty text input field.
- Owner ID:** A dropdown menu with a downward arrow.
- Options:**
 - Duplicate Members Allowed
 - Auto Configure Sparsity
 - Run Default Calc
- Comment:** A large empty text area.

Below these fields are two links: [Add Cube Dimensions](#) and [Remove Cube Dimensions](#). Below the links is a section titled 'Outline Structure' with a dark blue header. Inside this section, there is a 'Left | Right' label and a list containing one item: **+ OUTLINE_CUBE**. At the bottom of the page is a yellow 'Save' button.

On the Cube Outline page, click the Add Cube Dimensions link to access the Add Cube Dimensions page, where you can add dimensions to the current outline. Note that a dimension can be used once in an outline.

Image: Add Cube Dimensions page

This example illustrates the fields and controls on the Add Cube Dimensions page.



Click the Search button. The Available Dimensions section displays dimensions that are not already inserted into the dimension.

Select a dimension and click the OK button. The Cube Outline page reappears displaying the selected dimension.

Image: Cube Outline page displaying dimensions

In this example, the Cube Outline page displays the QE_CUBT_ALL_ACCOUNTS and ACCOUNTS dimensions.

The screenshot displays the 'Cube Outline' configuration interface. At the top, there are two tabs: 'Cube Outline' (selected) and 'Data Sources'. The main area shows the following configuration:

- Outline:** ACCOUNT_CUBE
- Outline Type:** BSO
- Description:** [Empty text box]
- Owner ID:** [Dropdown menu]
- Duplicate Members Allowed
- Auto Configure Sparsity
- Run Default Calc
- Comment:** [Empty text box]

Below the configuration fields, there are two links: 'Add Cube Dimensions' and 'Remove Cube Dimensions'. The 'Outline Structure' section is expanded, showing a tree view:

- ACCOUNT_CUBE**
 - QE CUBT ALL ACCOUNTS**
 - [E100 - (+)]
 - [E101 - (+)]
 - [E102 - (+)]
 - [A201 - (+)]
 - [E202 - (+)]
 - [E301 - (+)]
 - [E401 - (+)]
 - [E402 - (+)]
 - ACCOUNTS**
 - Account Query
 - [QEEU - (+)(QEEU)]
 - [QEUS - (+)(QEUS)]
 - [QESH - (+)(QESH)]

A yellow 'Save' button is located at the bottom left of the interface.

Each dimension is highlighted by a bright green icon. The dimension contents are shown in the outline.

Note: For dynamic queries, only a frame size of members is shown to give an idea of the contents of the dimension. The other frames are not viewable from this component.

Adding Cube Data Queries to a Cube Outline

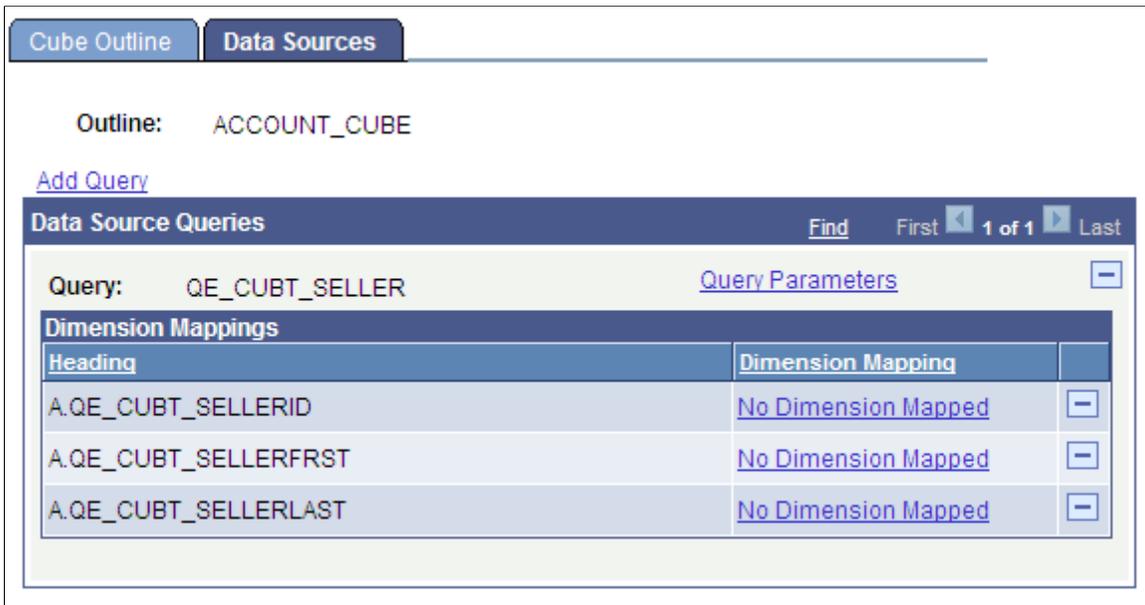
Use the Data Sources page (CUB_OUTLINE2) to add the cube data query or queries. You can supply one or more queries to fill the cube.

Navigation

Access the Data Sources page by selecting the Data Sources tab on the Cube Outline page.

Image: Data Sources page

This example illustrates the fields and controls on the Data Sources page. Definitions for the fields and controls appear following the example.



Add Query

Click to access the Find Query page, where you can search for and select a query from available queries.

Note: Each query must have one column for each dimension in the outline and one column for the measure. The measure is the numeric data that is the key piece of analysis in the online analytical processing (OLAP) cube, which is a set of multidimensional cells that hold numeric data.

Query Parameters

Click to access the Query Prompt Values page, where you can view the current prompt values for a query and modify them as needed.

Dimension Mappings

Displays a set of columns that will build your cube.

No Dimension Mapped

Click to access the Dimension Mapping page, where you can map the column to a dimension.

Save

Click to save the outlines.

You can only save valid outlines that have each data source query conforming to all of the following rules:

- Only one column is mapped as the measure.
- One column is mapped to each dimension that is part of the current outline.
- Each dimension exists in the current database.
- No duplicate mappings of dimensions or to the measure exist.
- At least one dimension exists in the outline.
- At least one data source query is defined in the outline.
- All columns are mapped.

Searching for Query

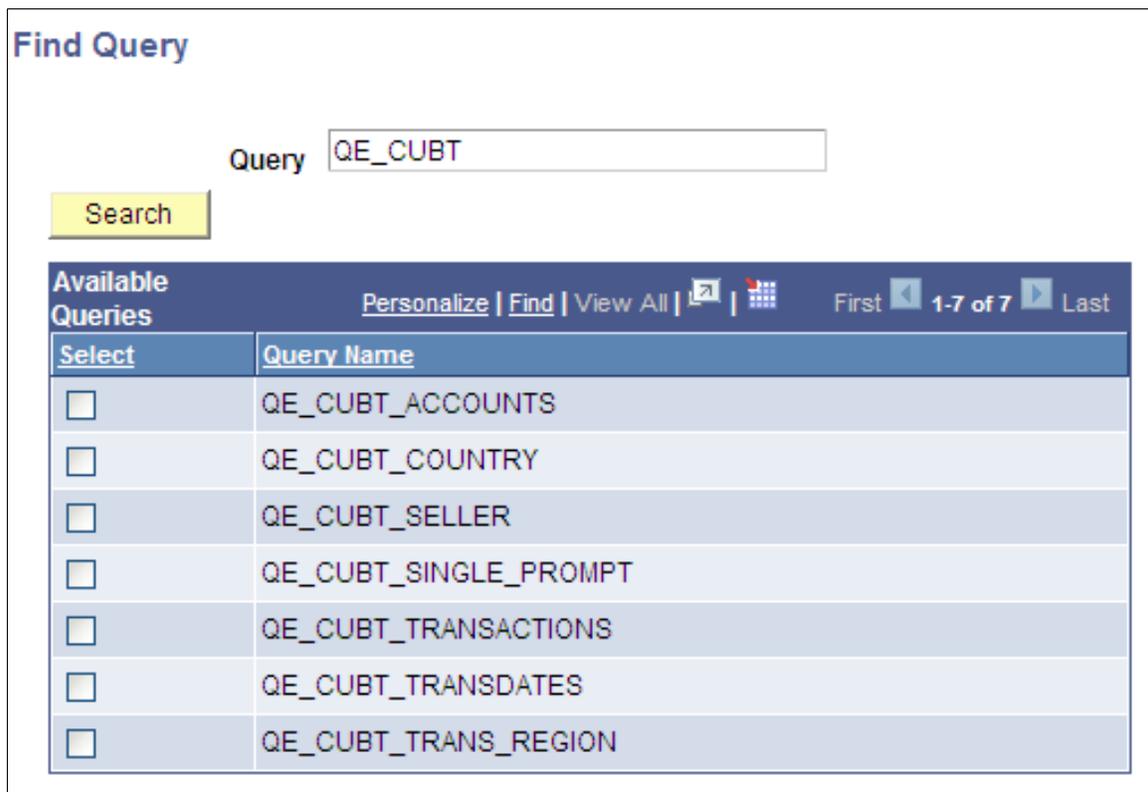
Use the Find Query page (CUB_QRYSEARCH_SEC) to search for available queries.

Navigation

Access the Find Query page by clicking the Add Query link on the Data Sources page.

Image: Find Query page

This example illustrates the fields and controls on the Find Query page. Definitions for the fields and controls appear following the example. The Available Queries section displays a list of existing queries that are defined in the system after you click the Search button. Note that only public queries to which the current user has authorization appear in this section.



Mapping to Dimensions

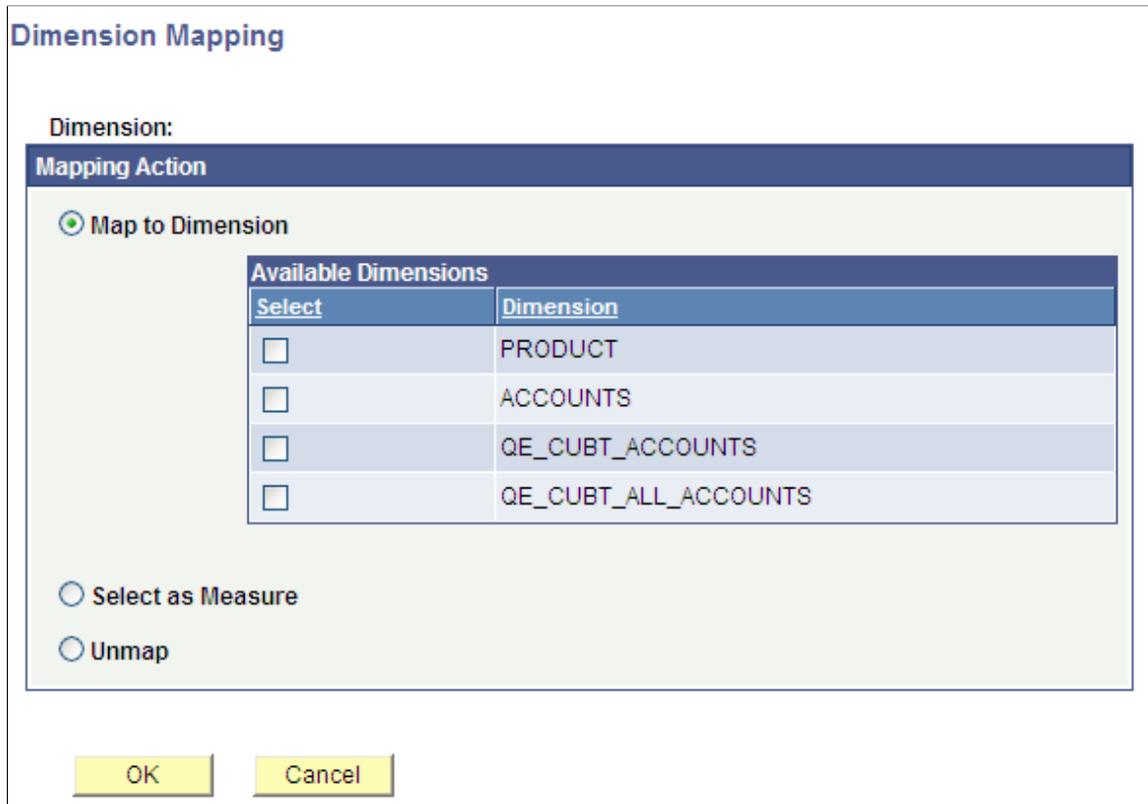
Use the Dimension Mapping page (CUB_OUTLINE2_SEC) to map columns from data extraction queries to dimensions, select a column from data extraction as the measure, and unmap previously mapped columns.

Navigation

Access the Dimension Mapping page by clicking the No Dimension Mapped link for the selected outline heading on the Date Sources page.

Image: Dimension Mapping page

This example illustrates the fields and controls on the Dimension Mapping page. Definitions for the fields and controls appear following the example.



Map to Dimension

Select to populate the Available Dimensions region, where you can select a dimension to map to your query.

Select as Measure

Select to define a column as a measure.

Note: Only one column can be selected as the measure.

Unmap

Select to unmap a column that is already mapped to a dimension.

OK

After mapping, click to return to the Data Sources page.

Inserting Query Prompts Values

Use the Query Prompt Values page (CUB_QRYPROMPT_SEC) to view the current prompt values for a query and change them if necessary.

Navigation

Access the Query Prompt Values page by clicking the Query Parameters link for the selected outline heading on the Date Sources page.

Image: Query Prompt Values page

This example illustrates the fields and controls on the Query Prompt Values page. Definitions for the field and control appear following the example.

Query Prompt Values

Query: CUBT_ACCOUNTS [Assign new prompt values](#)

| Prompt Field Values | | |
|---------------------|------|-------|
| Field Name | Type | Value |
| SETID | CHAR | EUR |

OK Cancel

Assign new prompt values

Click to access the New Prompt Values page, where you can change the input values of each query for each run.

Note: All changes on the New Prompt Values page are saved in the outline definition, so all subsequent runs will always use the current set of prompt values.

Prompt Field Values

Displays all data source queries that are assigned to the outline.

Example: Adding Cube Data Queries to a Cube Outline

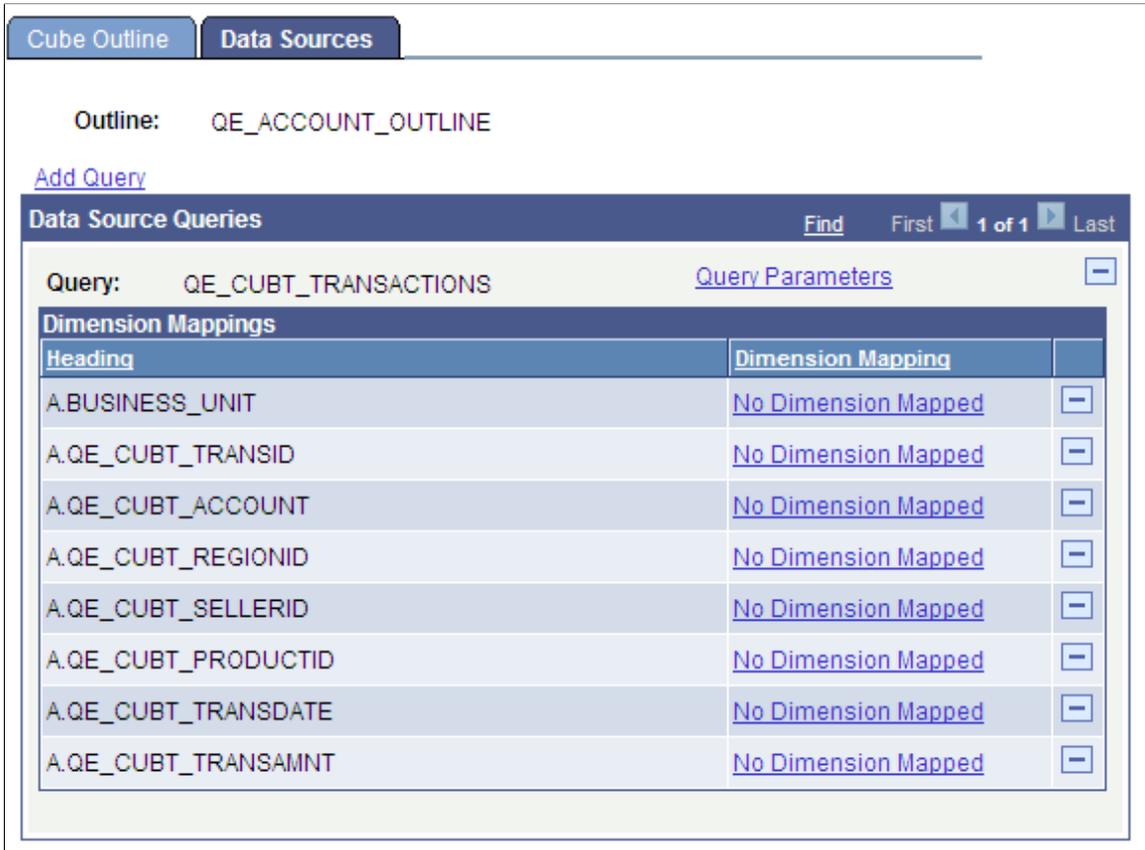
After entering the desired dimensions using the Cube Outline page, you need to supply one final key information: a query to use to supply the cube contents. Until now, you have provided information to build the cube structure. Now you use the Data Sources page in the outline component to add the cube data queries.

In the Data Sources page, click the Add Query link to access the Find Query page where you can select the desired query.

Note: On the Data Sources page, you can supply one or more queries to fill the cube.

Image: Data Sources page

This example shows the Data Sources page with the selected QE_CUBT_TRANSACTIONS query, which returns a full set of columns.



The query returns a number of columns, which would be typical for a real-life scenario. In this example, each of these columns needs a dimension:

- Account—QE_CUBT_ACCOUNT
- Region—QE_CUBT_REGIONID
- Seller—QE_CUBT_SELLERID
- Product—QE_CUBT_PRODUCTID
- Time—QE_CUBT_TRANSDATE
- Measure—QE_TRANSMOUNT

Click the Delete Row button (minus sign) to remove the business unit (BUSINESS_UNIT) and transaction ID (QE_CUBT_TRANSID) from the list because you do not need those two columns in this particular case.

Click the No Dimension Mapped link to access the Dimension Mapping page, where you can map the column to a dimension.

Image: Dimension Mapping page

This example illustrates the fields and controls on the Dimension Mapping page.

Dimension Mapping

Dimension:

Mapping Action

Map to Dimension

| Select | Dimension |
|--------------------------|----------------------|
| <input type="checkbox"/> | QE_CUBT_ALL_ACCOUNTS |

Select as Measure

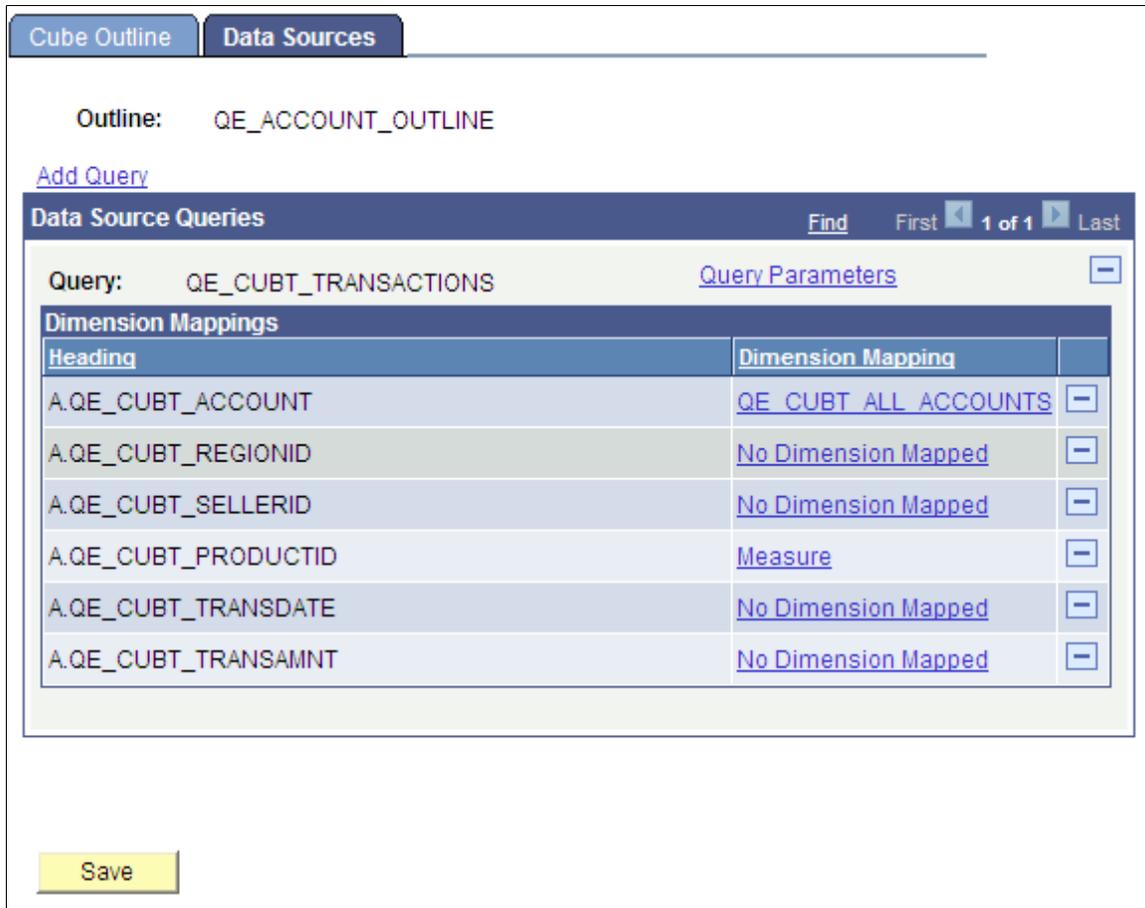
Unmap

OK Cancel

Select the Map to Dimension option to map the column to a dimension, and click the OK button. The Data Sources page reappears.

Image: Data Sources page with column mapped to dimension

This example illustrates the Data Sources page that has the QE_CUBT_ACCOUNT column mapped to the QE_CUBT_ALL_ACCOUNTS dimension and the QE_CUBT_PRODUCTID column tagged as the measure of the outline.



Note: Only one column can be selected as the measure. For the outline to be valid, all remaining columns in the Data Source Queries section must be mapped to a dimension or tagged as the measure.

Creating Dynamic Dimension Templates

Understanding Dynamic Dimension Templates

A dynamic dimension template is a set of default values to be used automatically for dynamic query and tree members. Because dynamic objects are not stored within the dimension, using a template enables the dimension builder to define what properties would be applied to the majority of members within the dynamic query and tree.

For members that are stored within the dimension—for example, having a dynamic query with all the consolidation properties in one location, and another dynamic query with all the consolidation properties in a different location—using templates enables you to set those values without setting them manually.

Using templates is optional. If you do not use templates, the system uses a default built-in template that assigns default values to each of the member's attributes. The default values are displayed in the Dynamic Dimension Templates page. If these default values will always be used, then templates should not be used.

Creating Dynamic Dimension Templates

Use the Dynamic Dimension Templates page (CUB_TEMPLATES) to create Essbase dynamic dimension templates.

Navigation

PeopleTools > Cube Builder > Dynamic Dimension Templates

Image: Dynamic Dimension Templates page

This example illustrates the fields and controls on the Dynamic Dimension Templates page. Definitions for the fields and controls appear following the example.

Dynamic Dimension Templates

Template ID QE_DYNAMIC_TEST

Description

Comments

Object Owner ID ▼

*Consolidation Property ▼ Expense Report

*Time Balance ▼ Never Share

*Skip Option ▼

Updated By VP1 Updated on 03/10/2009 3:39:47PM

Object Owner ID

Select the owner of the template.

The owner ID helps to determine the product, feature, or application team that last made changes to this template.

Consolidation Property

Define the mathematical operator used for rolling up members. Most often, you expect that data is added (using the + operator) when rolled up. However, you might occasionally need to specify another operator.

See [Defining an Essbase Dimension](#).

Expense Report

Essbase has built-in formulas that take advantage of the knowledge that an item is an expense. To pass this knowledge on to Essbase, select this check box .

Time Balance

Time balance affects how the parent time value is calculated. Select from these values:

- 0: none
- 1: first
- 2: last
- 3: average

Skip Option

If the Time Balance value is *first*, *last*, or *average*, then select a skip option that Essbase will use when it encounters missing values or 0 values.

Available options are:

- *Missing*
- *Missing and Zero*
- *None*
- *Zeros*

Building Essbase Cubes

Building Essbase Cubes

Use the Create Cubes (CUB_RUNCNTL) page to run the process that sends all of the defined information to the Essbase database to build Essbase cubes.

Navigation

Access the Create Cubes page by selecting PeopleTools > Cube Builder > Create Cube.

Image: Create Cubes page

This example illustrates the fields and controls on the Create Cubes page. Definitions for the fields and controls appear following the example.

Create Cubes

Run Control ID: RUNCNTRL1 [Report Manager](#) [Process Monitor](#)

Outline QE_CUBT_TRANS_DATE

Essbase Connection ID QE_CUBT_API_TEST

Application Name GEN

Database Name GEN1

*Meta-Data Update Action

*Restructuring options

*Aggregation On Load

Log level

Dimension

Query

Last Update Dttm 07/12/2011 5:22:36PM

- Outline** Displays the outline that is used to create this cube.
- Essbase Connection ID** Displays the Essbase connection ID that is used to create this cube.
- Application Name** Displays the Essbase application name that is used in the Essbase system to create the online analytical processing (OLAP) cube.

| | |
|--------------------------------|--|
| Database Name | Displays the database name that is used in the Essbase system to create the OLAP cube. |
| Meta-Data Update Action | <p>Select an update action to use to build the metadata (the cube shape). The selected update action controls what to do in case this cube creation process is not the first time the cube is built.</p> <p>Available options are:</p> <ul style="list-style-type: none"> • <i>Create or Re-create</i>: Create or recreate metadata from the beginning. • <i>Incremental Replace All</i>: Update metadata incrementally and remove unspecified members. • <i>Incremental Update</i>: Merge updates incrementally with existing metadata. • <i>No Action</i>: Do not perform an action on the metadata. |
| Restructuring Options | <p>Select an option to build the data (the cube contents).</p> <p>Available options are:</p> <ul style="list-style-type: none"> • <i>Discard all data</i>: Discard all previously loaded or manually modified data. • <i>Keep all data</i>: Do not modify the data. • <i>Keep only input data</i>: Keep only input data or data blocks created by data loading. • <i>Keep only level 0 data</i>: Delete upper level data incrementally. |
| Aggregations on Load | <p>Three available options are:</p> <ul style="list-style-type: none"> • Replace <p>Select this option to overwrite or replace the currently stored value in the data cell with the value being passed. Basically, if the data file has multiple rows with the same dimensional intersection, then the last value in the data cell is stored and the previous ones are ignored.</p> • Addition <p>Select this option to add all the values for the stored data cell to the currently stored value. Use this option for instances in which loads are always additive, for example, an implementation that loads transactions to an already stored summary amount.</p> • Subtraction |

Select this option to subtract the values passed from the currently stored value. This option is rarely used, but you can use it to perform an additive load that then needs to be backed out.

See “Using Data Source Queries“ in [Creating Queries](#).

Log Level

Enter the log level that pertains to the level of logging while the process is running.

These are the logging levels that can be selected to view specific messages or errors:

- 0 (eRequisite): no logging.
- 1 (eBasic): basic logging.
- 2 (eDetail): detailed logging.
- 4 (eDimensional): single dimension logging.
- 8 (eFactMapping): single factory logging.
- 16 (eAPITrace): Essbase APT traces.

For example, to see logging for Single Dimensions and Essbase APT, enter 20 (sum of eDimensional and eAPITrace = 4 + 16 = 20). To see all available logs, enter 31 (1 + 2 + 4 + 8 + 16 = 31).

Dimension

Select a dimension as the single dimension that loads data.

This selection sets the data update action to *No Action*. The only action that will occur in the current run is for the structure of dimension to be rebuilt. No data will be loaded.

Query

Select a query as the single query that loads data.

This selection sets the metadata action to *No Action*. While the process is loading in a single query, no structural changes are made to the cube.

Note: You should not provide both a dimension and a query because this selection would cause no actions to occur in the current run of the cube builder.

Last Update Dttm (last update date and time)

Displays the date and time that the cube creation run control ID was last modified.

Run

After saving the defined information, click this button to run the batch program.

Related Links

"Submitting Process Requests" (PeopleTools 8.58: Process Scheduler)

Integrating with Oracle Smart View

Using Oracle Smart View for Microsoft Excel, you can view the data and metadata of the recently built cube from Process Monitor.

Note: Currently, Chrome browser is not supported for opening SmartView for Cube Builder.

To view cube data using Smart View:

1. Build Essbase cube.

See [Building Essbase Cubes](#).

2. After successfully building an Essbase cube, click the Process Monitor link on the Create Cube page to access the Process List page.
3. Click the Details link for the selected process instance.

If an Essbase cube was built successfully, the Open Cube in Smart View link is available on the Process Detail page.

Image: Process Detail page and the Open Cube in Smart View link

This example illustrates the Process Detail page with the Open Cube in Smart View link.

| Process | |
|--|---|
| Instance 10507 | Type Essbase Cube Builder |
| Name PSBLDESS | Description Build Essbase Cube |
| Run Status Success | Distribution Status Posted |
| Run | Update Process |
| Run Control ID test | <input type="radio"/> Hold Request |
| Location Server | <input type="radio"/> Queue Request |
| Server PSNT | <input type="radio"/> Cancel Request |
| Recurrence | <input checked="" type="radio"/> Delete Request |
| | <input type="radio"/> Restart Request |
| Date/Time | Actions |
| Request Created On 08/10/2011 3:30:38PM PDT | Parameters Transfer |
| Run Anytime After 08/10/2011 3:30:35PM PDT | Message Log |
| Began Process At 08/10/2011 3:31:07PM PDT | Batch Timings |
| Ended Process At 08/10/2011 3:32:07PM PDT | View Log/Trace |
| | Open Cube in Smart View |

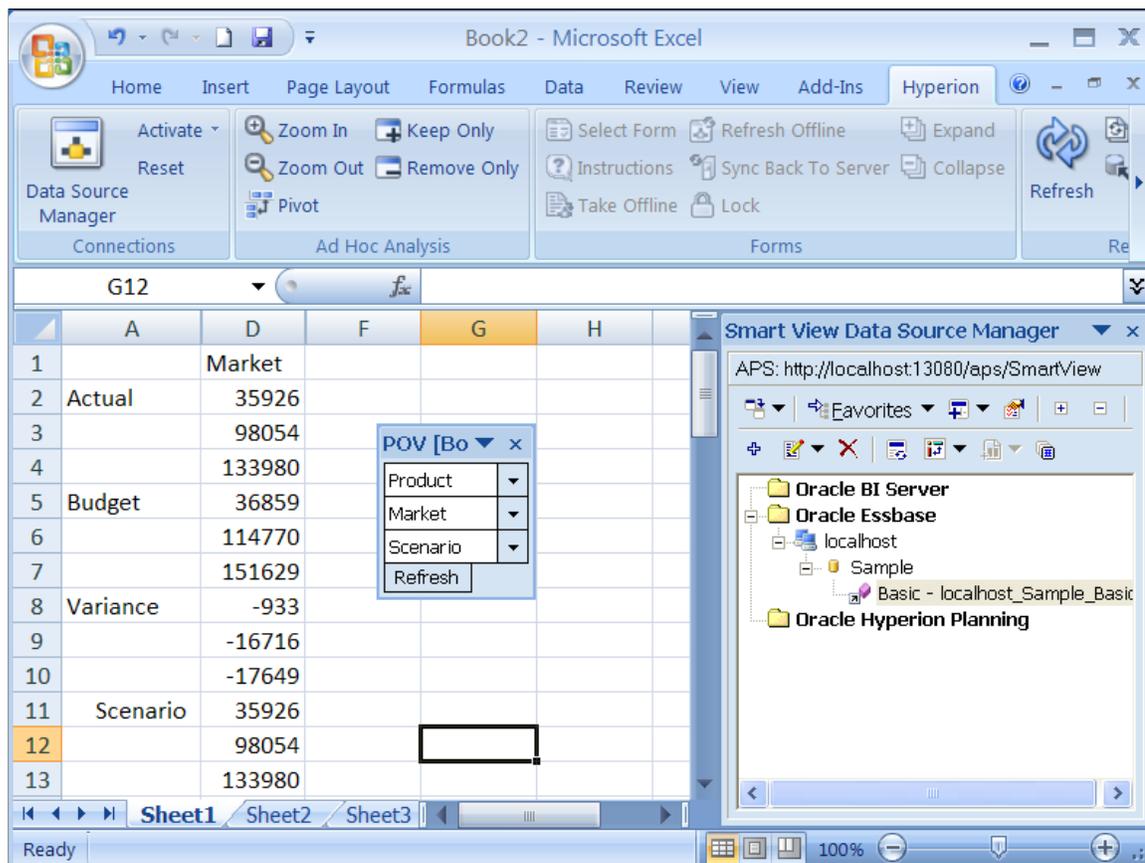
4. Click the Open Cube in Smart View link.

5. If required, enter the user name and password that was used to build the cube.

If your user name and password are authorized, Smart View displays the top members of the two dimensions created.

Image: Example of Smart View displaying the top members of the two dimensions created

Example of Smart View displaying the top members of the two dimensions created.



6. Optionally, perform drilldown or use the POV Smart View tool to manage the data.

Note: If you modify data or metadata using Smart View, these modifications affects the cube that is stored in Essbase but do not affect the PeopleSoft tree or query used by PS2Essbase to create the cube. If you make any modifications using Smart View, these modifications may be overwritten in the next run of PS2Essbase. If multiple process instances of PS2Essbase were run using Process Scheduler, the Process List page displays only one Open Cube in Smart View link for the last cube built.

Using API and Command-line Specifications

You can run PS2Essbase from the Create Cube page using Process Scheduler or the command line.

The following table lists the parameters that PS2Essbase uses when PS2Essbase runs from the command line.

| Parameter | Description |
|------------------|---|
| A, a | Outline Name. |
| R, r | Run Control ID. |
| I, i | Process Instance ID number, such as <i>I42</i> , usually passed by Process Scheduler. |
| CT, ct | Database type. For example, <i>-CTORACLE</i> . |
| CD, cd | Database name. For example, <i>-CDPT850RC1</i> . |
| CO, co | Operator ID. For example, <i>-COPTDMO</i> . |
| CP, cp | Password. For example, <i>-CPPASSWORD</i> . |
| EC, ec | Essbase Connection Name. |
| OP, op | Log files output destination. Note: When running PS2Essbase from the command line, in the <i>-OP</i> command line option, you must enclose the output destination patch in quotes. For example, <i>-OP "c:\tmp\ps"</i> . |
| M, m | Metadata action, such as <i>-mC</i> : <ul style="list-style-type: none"> • N - No action on metadata. • C - (Re)create metadata from scratch. • U - Incrementally merge updates with existing metadata. • R - Incrementally update existing metadata, remove unspecified members. |
| D, d | Outline Restructure option: <ul style="list-style-type: none"> • N - Keep all data. • R - Discard all data. • I - Reload all data. • D - Keep only level 0 data. • A - Keep only input data or data blocks created by data loading. |

| Parameter | Description |
|------------------|---|
| L, l | Logging level. For example, <i>L31</i> . |
| X, x (optional) | Dimension name for single dimension execution. |
| Y, y (optional) | PS Data Query name for single factory execution. |
| Z, z | Parameter used to load the aggregations on load option. Available options are: <ul style="list-style-type: none"> • R,r - Replace. • A,d - Addition. • S,s - Subtraction. |

For example, the parameters list passed using this command line:

```
ps2essbase.exe -CX -CTORACLE -CDPT850RC1 -COPTDMO
-CPPASSWORD -I42 -OP "c:\tmp\ps" -REMPLOYMENT -aEG_DIVISION
-mC -dR-L15 -XGL_ACCOUNT -YQRY_MEASURES
```

- Run Control ID = EMPLOYMENT
- Outline = EG_DIVISION
- Metadata Action = Create

Note: The log files will be deposited in the `-OP` location. For this example, the `-OP` location is “`c:\tmp\ps`”.

- Restructure Option = Discard all data

These metadata and restructure options indicate that metadata will be fully recreated or created if it did not exist, and all previous data, if it existed, will be discarded.

- Logging level includes Basic, Detail, Member, and Fact map logging; we specified L15, where $15 = 1$ (Basic) + 2 (Detail) + 4 (Member) + 8 (Fact).
- Single Dimension = GL_ACCOUNT
- Single Data Query = QRY_MEASURES

Logging

You can pass the log level with the L option to the command line of `psolap.exe`; for example, L2 or L6, and so on.

The following table lists the log level and the data recorded in each log level.

| Log Level | What is recorded |
|------------------|-------------------------|
| 1 | Basic logging |

| Log Level | What is recorded |
|------------------|--------------------------|
| 2 | Detail logging |
| 4 | Dimension member logging |
| 8 | Fact map logging |
| 16 | API logging |

When you run the process using Process Scheduler, the logging level is set on the Create Cube page. When you run the process from the command line, the logging level is set using the `-L` parameter. To calculate these logging settings, Cube Builder adds the options that you define. For example:

- To see logs for Single Dimensions and Essbase API, enter 20.

20 is the sum of Dimension member logging + API logging = 4 + 16 = 20.

- To see all the logs that are available, enter 31.

31 is the sum of Basic logging + Detail logging + Dimension member logging + Fact map logging + API logging = 1 + 2 + 4 + 8 + 16 = 31.

Chapter 8

Securing Cubes

Securing Cube Builder Cubes

This topic discusses how to secure cube builder cubes.

Controlling User Operations

Use the Component Permissions page (ACL_COMPONENT2) to control the operations that a user is allowed to perform within Essbase Cube Builder.

Navigation

To access the Component Permissions page:

1. Select PeopleTools > Security > Permissions & Roles > Permission Lists.
2. Add a new permission list or search for an existing one.
3. Select the Pages tab and then click an appropriate Edit Components link.

Image: Component Permissions page

This example illustrates the fields and controls on the Component Permissions page. Definitions for the fields and controls appear following the example.

The screenshot shows the 'Component Permissions' page. At the top, there is a 'Cube Manager' breadcrumb. Below it is a table with the following columns: 'Authorized?' (checkbox), 'Component Name', 'Item Label', 'Edit Pages' (link), and 'View Content References for this Component' (link). The table contains 12 rows of data. To the right of the table are 'Select All' and 'Deselect All' buttons. At the bottom of the table are 'OK' and 'Cancel' buttons. The table data is as follows:

| Authorized? | Component Name | Item Label | Edit Pages | View Content References for this Component |
|-------------------------------------|-------------------|------------------------------|----------------------------|--|
| <input checked="" type="checkbox"/> | ANALYSIS_DB | Cube Instances | Edit Pages | View |
| <input checked="" type="checkbox"/> | ANALYSIS_MODEL | Cube Definitions | Edit Pages | View |
| <input checked="" type="checkbox"/> | ATTRIBUTE_NAME | Attribute Names | Edit Pages | View |
| <input checked="" type="checkbox"/> | BUILD_ANALYSIS_DB | Build Cubes | Edit Pages | View |
| <input type="checkbox"/> | CUB_CONNECTID | Connections | Edit Pages | View |
| <input type="checkbox"/> | CUB_DIMDELETE | Cube Utilities | Edit Pages | View |
| <input type="checkbox"/> | CUB_DIMENSION | Dimension Builder | Edit Pages | View |
| <input type="checkbox"/> | CUB_GUISETUP | Cube Dimension Builder Setup | Edit Pages | View |
| <input type="checkbox"/> | CUB_OUTLINE | Cube Outline | Edit Pages | View |
| <input type="checkbox"/> | CUB_RUNCNTL | Create Cube | Edit Pages | View |
| <input type="checkbox"/> | CUB_TEMPLATES | Dynamic Cube Templates | Edit Pages | View |
| <input checked="" type="checkbox"/> | DIMENSION | Dimensions | Edit Pages | View |

| | |
|---|---|
| Authorized? | Displays the components to which the current permission list has access. |
| Component Name | <p>Displays the name of components beneath the authorized menu. Cube Builder components are:</p> <ul style="list-style-type: none"> • CUB_CONNECTID, which is the Connections component. (Select PeopleTools > Cube Builder > Connections.) • CUB_DIMDELETE, which is the Utilities component. (Select PeopleTools > Cube Builder > Utilities.) • CUB_DIMENSION, which is the Dimensions component. (Select PeopleTools > Cube Builder > Dimensions.) • CUB_GUISETUP, which is the Cube Dimension Builder Setup component. (Select PeopleTools > Cube Builder > Cube Dimension Builder Setup.) • CUB_OUTLINE, which is the Outlines component. (Select PeopleTools > Cube Builder > Outlines.) • CUB_RUNCNTL, which is the Create Cube component. (Select PeopleTools > Cube Builder > Create Cube.) • CUB_TEMPLATES, which is the Dynamic Dimension Templates component. (Select PeopleTools > Cube Builder > Dynamic Dimension Templates.) |
| Item Label | Displays the menu item label that you can click to access a component. |
| Edit Pages | <p>This link appears for every component listed under the current menu.</p> <p>Click to access the Page Permissions page, where you can set the actions that a user can complete on a page.</p> <p>See Setting User Actions.</p> |
| View Content References for this Component | <p>This link appears for every component listed under the current menu.</p> <p>Click the View link to access the Content References page, where you can view all the content references in the database that point to the component.</p> |

See [Viewing Content References](#).

Select All

Click to grant access to all or selected items.

To grant access to *most* menu items, click this button and then clear the items to which you do not want to grant access.

Deselect All

Click to deselect items that were previously selected.

Setting User Actions

Use the Page Permissions page (ACL_PAGES2) set the actions that a user can complete on a page.

Navigation

To access the Page Permissions page:

1. Select PeopleTools > Security > Permissions & Roles > Permission Lists.
2. Add a new permission list or search for an existing one.
3. Select the Pages tab and then click an appropriate Edit Components link.
4. Click an appropriate Edit Pages link.

Image: Page Permissions page

This example illustrates the fields and controls on the Page Permissions page. Definitions for the fields and controls appear following the example.

Page Permissions

CUBE_MANAGER / Design / Dimension Builder

| Panel Item Name | Authorized? | Display Only |
|------------------|--------------------------|--------------------------|
| Dimension | <input type="checkbox"/> | <input type="checkbox"/> |
| Generation Names | <input type="checkbox"/> | <input type="checkbox"/> |

Actions

- Add
- Update/Display
- Update/Display All
- Correction

Select All

Deselect All

OK Cancel

Panel Item Name

Lists the pages of the selected component.

| | |
|---------------------|--|
| Authorized? | Select to enable a user to access the page. After selecting this option, you must decide the degree to which a user is authorized on a page by selecting the Display Only option or at least one of the other available options in the Actions section. |
| Display Only | Select this option to enable the user to view the information on the page but not to modify any of the data. To enable write access to a page, deselect this option. |
| Actions | To allow users to modify data on the page, select one option from those that appear in the Actions section, such as <i>Add</i> , <i>Update/Display</i> , and <i>Correction</i> . The options that are available in the Actions section depend on the options that were selected when the page was initially developed in PeopleSoft Application Designer. |
| Select All | Click to grant access to all pages and to all the actions for each page. |
| Deselect All | Click to deselect all previously selected items. |

Viewing Content References

Use the Content References page (PLIST_CREF) to view all of the content references in the database that point to the selected component. If the Accessible option is selected, then the associated permission list includes access to all parent folders of the content reference.

Navigation

To access the Content References page:

1. Select PeopleTools > Security > Permissions & Roles > Permission Lists.
2. Add a new permission list or search for an existing one.
3. Select the Pages tab and then click an appropriate Edit Components link.
4. Click an appropriate View link.

Image: Content References page

This example illustrates the fields on the Content References page.

Content References

Component: **Cube Utilities**

The grid below shows all the content references in this database that point to this component.
If the "Accessible" column is checked, this Permission List includes access to all parent folders of the content reference.

| Portal Name | Name | Label | Accessible | Path |
|-----------------|------------------|-----------|--------------------------|--|
| CUSTOMER | PT_CUB_DIMDELETE | Utilities | <input type="checkbox"/> | Root(PORTAL_ROOT_OBJECT).PeopleTools(PT_PEOPLETOOLS).Cube Builder (PT_ESSBASE_CUBES) |
| EMPLOYEE | PT_CUB_DIMDELETE | Utilities | <input type="checkbox"/> | Root(PORTAL_ROOT_OBJECT).PeopleTools(PT_PEOPLETOOLS).Cube Builder (PT_ESSBASE_CUBES) |
| MOBILE | PT_CUB_DIMDELETE | Utilities | <input type="checkbox"/> | Root(PORTAL_ROOT_OBJECT).PeopleTools(PT_PEOPLETOOLS).Cube Builder (PT_ESSBASE_CUBES) |
| PARTNER | PT_CUB_DIMDELETE | Utilities | <input type="checkbox"/> | Root(PORTAL_ROOT_OBJECT).PeopleTools(PT_PEOPLETOOLS).Cube Builder (PT_ESSBASE_CUBES) |
| PS_SITETEMPLATE | PT_CUB_DIMDELETE | Utilities | <input type="checkbox"/> | Root(PORTAL_ROOT_OBJECT).PeopleTools(PT_PEOPLETOOLS).Cube Builder (PT_ESSBASE_CUBES) |
| SUPPLIER | PT_CUB_DIMDELETE | Utilities | <input type="checkbox"/> | Root(PORTAL_ROOT_OBJECT).PeopleTools(PT_PEOPLETOOLS).Cube Builder (PT_ESSBASE_CUBES) |

Related Links

"Understanding Permission Lists" (PeopleTools 8.58: Security Administration)

"Setting Up User Profile Types" (PeopleTools 8.58: Security Administration)

Maintaining Essbase Cubes

Maintaining Essbase Cubes

This topic provides an overview of Essbase cubes maintenance.

Deleting Dimensions

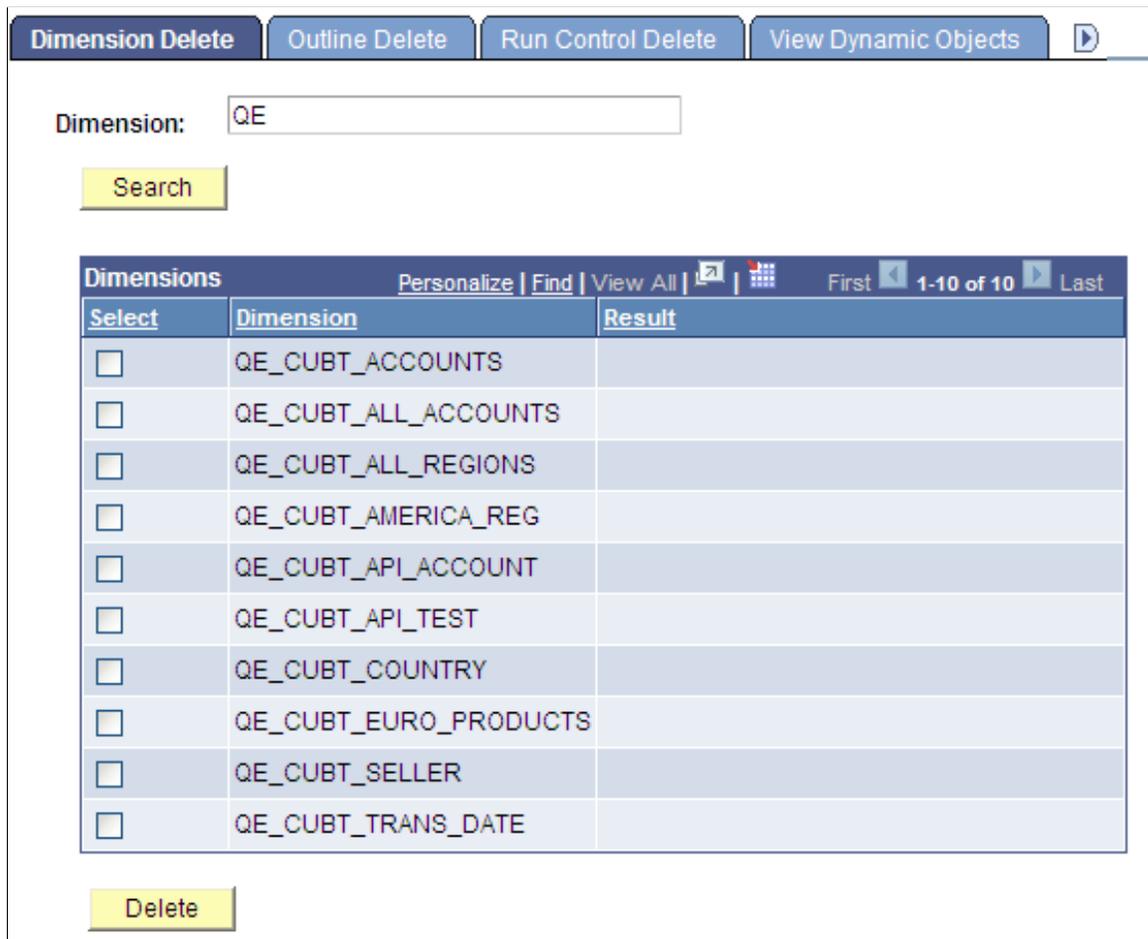
Use the Dimension Delete page (CUB_DIMDELETE) to delete any dimension that was created in the system as long as the dimension is not referenced in any outline.

Navigation

Access the Dimension Delete page by selecting PeopleTools > Cube Builder > Utilities.

Image: Dimension Delete page

This example illustrates the fields and controls on the Dimension Delete page.



Deleting Outlines

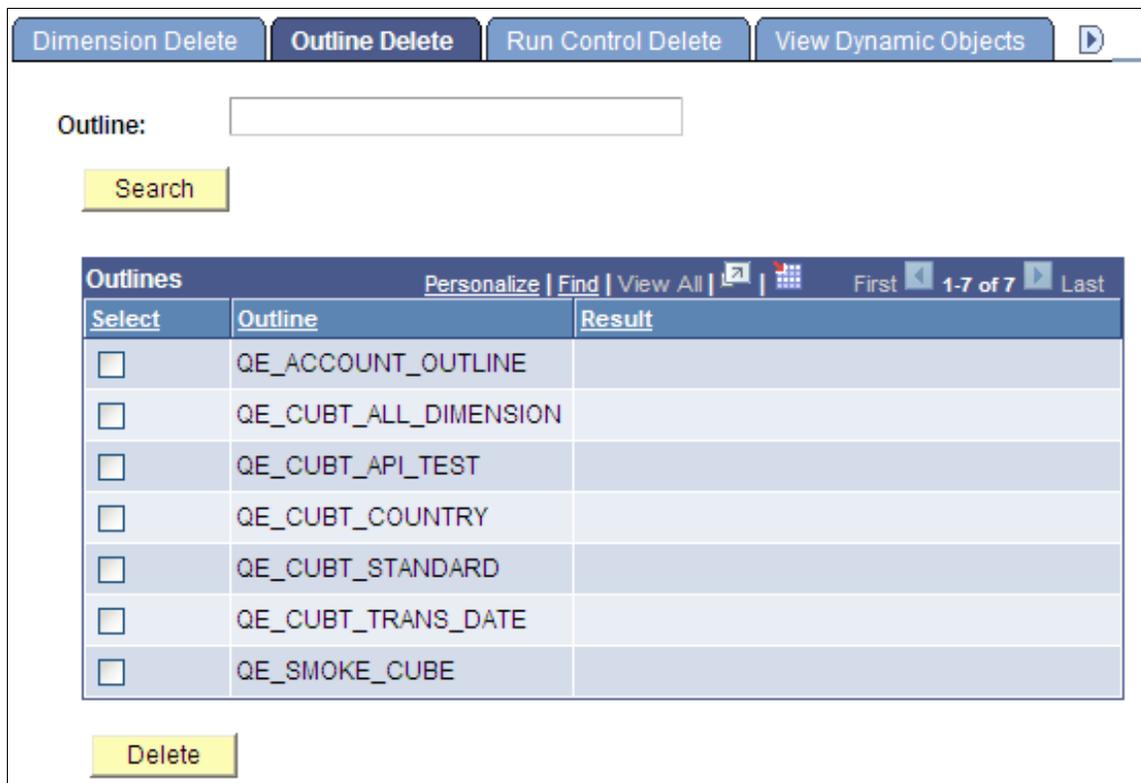
Use the Outline Delete page (CUB_OUTDELETE) to delete any created outline in the system as long as the outline is not referenced in any cube creation run control.

Navigation

Access the Outline Delete page by selecting the Outline Delete tab on the Dimension Delete page.

Image: Outline Delete page

This example illustrates the fields and controls on the Outline Delete page.



Deleting Run Control IDs

Use the Run Control Delete page (CUB_RUNDELETE) to delete any cube creation run control.

Navigation

Access the Run Control Delete page by selecting the Run Control Delete tab on the Dimension Delete page.

Image: Run Control Delete page

This example illustrates the fields and controls on the Run Control Delete page.

Run Control ID:

Search

| Select | Run Control ID | Result |
|--------------------------|----------------|--------|
| <input type="checkbox"/> | RUNCNTRL1 | |

Delete

Viewing Dynamic Objects

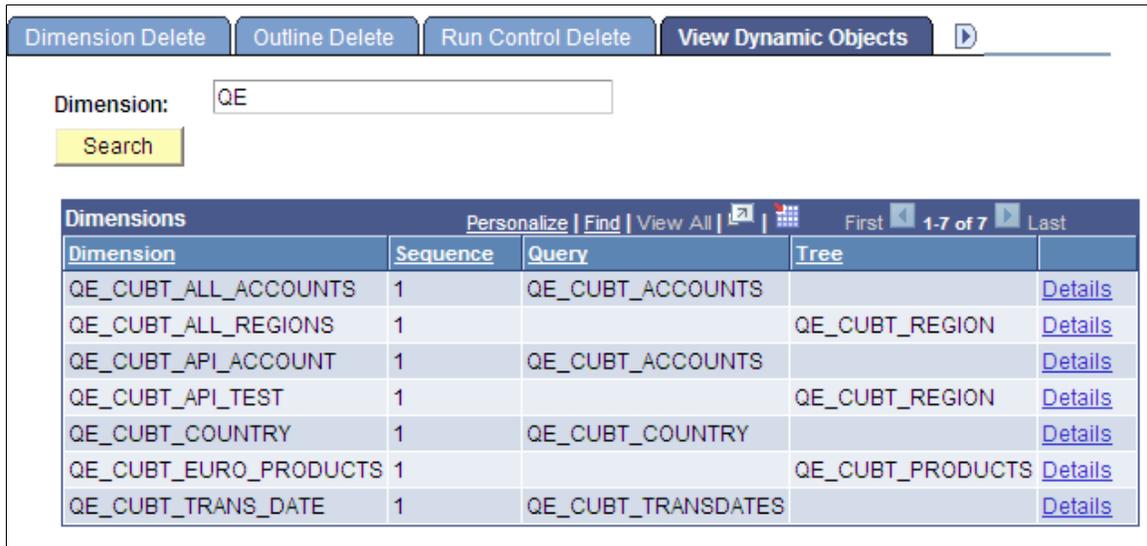
Use the View Dynamic Objects page (CUB_DIMDYNVIEW) for the selected dimension to view all dynamic objects (queries and trees) that are used in the definition. This page also displays runtime parameters that are used when you run queries. If you are having problems opening the dimension definition, you can view the queries and trees that are part of the dimensional structure. This page is useful in diagnosing permission problems with dynamic objects.

Navigation

Access the View Dynamic Objects page by selecting the View Dynamic Objects tab on the Dimension Delete page.

Image: View Dynamic Objects page

This example illustrates the fields and controls on the View Dynamic Objects page.



Viewing Outline Structures

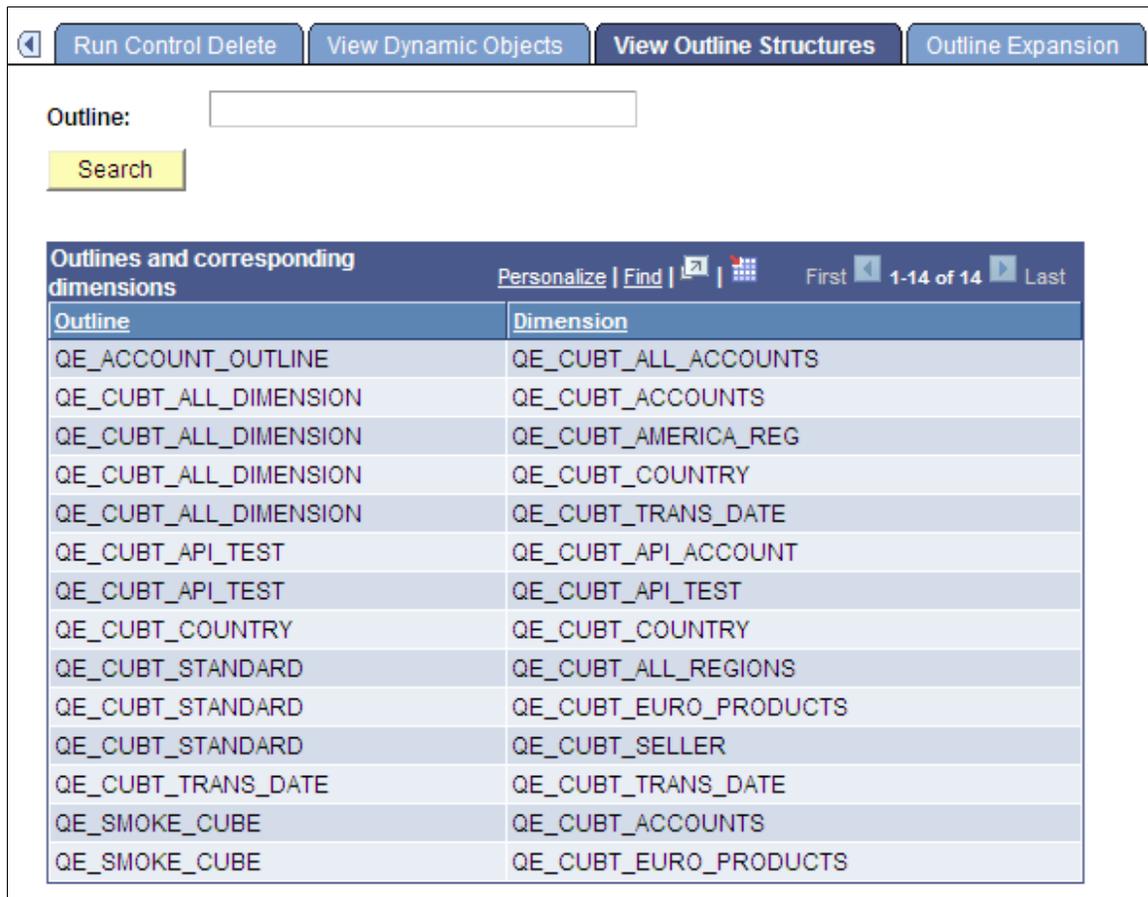
Use the View Outline Structures page (CUB_OUTSTRUCT) to view all the dimensions that are part of the outline for the selected outline.

Navigation

Access the View Outline Structures page by selecting the View Outline Structures tab on the Dimension Delete page.

Image: View Outline Structures page

This example illustrates the fields and controls on the View Outline Structures page.



Expanding Dimensions

Use the Outline Expansion page (CUB_DYNEXPAND_SEC) to perform the full dimension expansion for every dimension that is part of the given outline. This page runs all dynamic queries and trees in each dimension and displays the total number of rows processed for each dynamic object. For non-dynamic members of the dimension, this page shows an overall total only. All errors encountered during the expansion appear, along with relevant information.

Navigation

Access the Outline Expansion page by selecting the Outline Expansion tab on the Dimension Delete page.

Image: Outline Expansion page

This example illustrates the fields and controls on the Outline Expansion page.

Run Control Delete | View Dynamic Objects | View Outline Structures | **Outline Expansion**

Outline: 🔍

Expand

Result: Return code from expansion - 0.

| Expansion Results | | | | |
|----------------------|------------------|------|---------------|----------------|
| Dimension | Query | Tree | Non-dynamic | Processed Rows |
| QE_CUBT_ALL_ACCOUNTS | QE_CUBT_ACCOUNTS | | | 8 |
| QE_CUBT_ALL_ACCOUNTS | | | (non-dynamic) | 0 |

Notify

Viewing Query Results That Have Large Result Sets

Query Chunking and the Chunking Class

Query chunking provides a valuable methodology for displaying query run results of queries with very large result sets.

Using application package class `PT_CUBQUERYCHUNK:QueryChunker` enables you to display query result sets as frames of rows. The number of rows in a frame can be configured to be between 10 and 1000. This chunking object maintains an internal buffer of 10 frames and it enables the user to navigate the query results frame by frame, not as it refreshes the internal buffer dynamically during navigation.

During navigation, the query is opened only as needed and is closed after the relevant rows are retrieved into the internal buffer area. No open cursors are on the query object during navigation.

To instantiate a chunking object, you would run the following code:

```
&NewQueryChunk = create PT_CUBQUERYCHUNK:QueryChunker(&myrecordname, &myseqnum,  
&qpromptrec, &COMP_FRAME_SIZE, &fheading);
```

Instantiation requires a query name, a unique sequence number, the prompt record associated with the query (if any), the frame size desired, and a select column heading. This last parameter is used to uniquely identify a select column from the query definition. The query chunker application supports only a single column from the select clause of the query being chunked. The frame size must be between 10 and 1000. If higher or lower values of the frame size are passed in, the number is modified internally to meet this requirement.

Post instantiation, the validity of the instantiation must be checked as shown here:

```
If Not &NewQueryChunk.valid Then  
  /*report an error here */
```

The class also contains two properties to report error conditions: `errorcode` and `erroropencode`. The second property can be checked for any query open error. It reports if the query does not exist or if the user does not have permission to access the query.

If the process is working fine, then this code can get a reference to the current frame:

```
&retarray = &NewQueryChunk.GetFrame();
```

The following table lists the error codes for the chunker.

| Code | Error |
|-------------|---------------------|
| -1 | Missing query name. |

| Code | Error |
|-------------|---|
| -2 | Missing field heading. |
| -3 | Error opening the query. Check <code>erroropencode</code> for more information. |
| -4 | Mismatch between the prompt record that was passed in and the current query definition prompt values. |
| -5 | Unable to find the field heading that was passed in within the current query definition. |
| -6 | Error reopening query for a new chunk. |

This chunking capability can be used anywhere for any query. Page `CUB_QUERYNAV_SEC` can be added to any component to display chunking results.

These are the rules to use the `CUB_QUERYNAV_SEC` page:

1. The following objects must be defined before you access the page:

```
Component string &DYN_QUERYNAME;
```

```
Component number &DYN_SEQNUM;
```

```
Component PT_CUBQUERYCHUNK:QueryChunker &cChunker;
```

These values hold the query name and the unique sequence number. Remember that the sequence number is used to uniquely identify the query.

Some components could potentially be using the same query multiple times; the sequence number is used to uniquely identify a query instance. The page activate code makes sure that the current chunking object matches the current query and sequence number that were passed in.

2. The rowset in the `CUB_QUERYNAV_SEC` page is populated with the current frame of the chunking object as the page is displayed.

If this is the first time the chunking object is displayed, then the first frame is shown. This same rowset could be referenced in the calling component.

```
&rs = GetLevel0().GetRow(1).GetRowset(Scroll.PSCUBCHUNKING);
```

You can navigate back and forth within the chunking page. When the secondary page is dismissed, the rowset contains the current frame. You could also just get the current frame from the chunking object itself.

```
&retarray = &cChunker.GetFrame(); /* Get the current frame. */
&cframe = &cChunker.currentframe; /* Get the current frame number. */
```

On the `CUB_DIMENSION` component, an array of chunking classes is instantiated upon component load. Each dynamic query results in a new entry in this array. When you use chunking, the array is traversed to find the correct chunking object to pass into the secondary page. Object `&cChunker` is set to the appropriate entry from the array and the secondary page then can navigate back and forth.

Dynamic Query Expansion

Application class `PT_CUBQUERYCHUNK:Utilities:ExpandStructure` is used to expand dynamic queries from dimension definitions. The dimension definition holds values only for dynamic query entries to which you have added new information that is, information not found as a default value for the dimension members.

During cube population, the query is rerun and its members extracted to always send to Essbase the most up-to-date member list. This expansion of dynamic queries is completed with this class, which is called in the following way:

```
&ExpandStruct = create
PT_CUBQUERYCHUNK:Utilities:ExpandStructure(PSCUBWRK2.CUB_OUTLINEID, "");
```

The first parameter is the outline to expand. This parameter expands every dimension in the outline. Alternatively, you can expand a single dimension by leaving the first parameter blank and passing in the dimension ID in the second parameter.

Post expansion, the return code of the expansion should be checked in the following way:

```
If &ExpandStruct.errorcode <> 0 then
/* report error */
```

Message catalog entry 79,1103 contains the current list of expansion error codes. If new codes are added, this entry should be kept up to date.

The following tables include the current codes.

| Code | Error |
|-------------|--|
| -21 | Error opening a query. |
| -22 | Error retrieving presaved prompt values for a query. |
| -23 | Error obtaining selected column for a query. |
| -42 | Error tagging an entry in the internal dynamic buffer for a query. |
| -30 | Security error on a tree definition. |
| -31 | Error opening a tree. |
| -32 | Error finding the required root node on a tree definition. |
| -41 | Error: found dynamic tree leaf. |
| -43 | Error tagging an entry in the internal dynamic buffer for a tree. |

| <i>/* Leaf SQL Errors */</i> | Error |
|-------------------------------------|----------------------------|
| -51 | Empty tree structure name. |
| -52 | Invalid tree structure. |

| <i>/* Leaf SQL Errors */</i> | Error |
|------------------------------|-----------------------------|
| -53 | Empty detail search record. |
| -54 | Empty detail field name. |

| <i>/* Duplicate checks */</i> | Error |
|-------------------------------|--|
| -101 | Level 0 members with same parent and identical names. |
| -102 | Invalid shared member reference. Either the first duplicate name had the shared option selected, or a subsequent duplicate name did not have the shared option selected. Shared members must reference a prior nonshared member. |

The expansion routing populates table PSCUBDIMSTRCOUT based on the dimension structural information found in table PSCUBDIMSTRUCT. The latter table contains references to dynamic queries and any of the saved override values for specific members.

After the dimension is expanded, PSCUBDIMSTRCOUT will contain a row for every returned row for every dynamic query, so it can grow quite large. Every time a dimension is expanded, any rows from prior runs of the expansion routine are purged. Because this table can grow quite large, its contents should not be displayed on a regular PIA page.

A good example of PeopleCode used to expand an outline can be seen in the Expand Outline utility page, as described previously.

For the actual process of creating cubes, the C++ program PS2ESSBASE calls the expansion class to always perform a clean expansion prior to sending in anything to Essbase.

Appendix B

Using Essbase Cube Builder Demo Data

Understanding Essbase Cube Builder Demo Data

Essbase Cube Builder demo data is being provided as part of PeopleTools so that you can get an idea of the structures involved in creating cubes.

Project CUBE_DEMO_PROJ is delivered in the PROJECTS folder of the installation. Data mover and data files, ps_cubebuilder.dat and ps_cubebuilder.dms, are also being delivered. These files load the necessary data to show prebuilt dimensions and outlines as well as demo data that you can add to, as needed.

Applying Project CUBE_DEMO_PROJ

Project CUBE_DEMO_PROJ contains tables for a cube test suite. This project also includes a new menu —CUBE_TEST—and portal registry entries for a new folder and components. All demo-date portal-registry entries appear under the Essbase Cube Builder folder.

Project CUBE_DEMO_PROJ also includes queries CUBT_ACCOUNTS, CUBT_TRANSACTIONS, CUBT_TRANSDATES, and CUBT_TRANS_ALL.

After importing project CUBE_DEMO_PROJ, create all tables and then build the view PSCUBTTRNSD_VW.

Project CUBE_DEMO_PROJ is delivered in the PROJECTS folder as part of the PeopleTools.

Granting Permissions to All Menu CUBE_TEST Components

Granting permissions to all menu components is often completed to permission list PTPT1200.

To grant permissions:

1. Select PeopleTools > Security > Permissions & Roles > Permission Lists.
2. Open permission list *PTPT1200*.
3. In the Pages page, add new menu *CUBE_TEST*.
4. Click the Edit Components link of menu CUBE_TEST, and grant permission to every component.

The Component Permissions page for Cube Manager appears.

After permission is granted, a new folder called Cube Tester Sample Data appears under the Essbase Cube Builder folder. Items in the Cube Tester Sample Data folder are used for *Accounts*, *Products*, *Regions*, *Sellers*, and *Transactions*. You can add new items to any of these components to create more demo data.

Importing PS_CUBEBUILDER.DAT File Using Data Mover Script PS_CUBEBUILDER.DMS

The data mover script PS_CUBEBUILDER.DMS includes the following items:

- Tree structures.

Includes two tree structures: CUBT_REGIONSTRUCT and CUBT_PRODUCTS.

- Trees.

Includes two trees: CUBT_PRODUCTS and CUBT_REGIONS.

The tree CUBT_PRODUCTS is linked to the product table and makes use of setID controls to show how these controls work in trees.

The tree CUBT_REGIONS is linked to the region table.

- Data for all the tables that are imported in project CUB_TESTDATA.

The data import file PS_CUBEBUILDER.DAT includes all this set control data that is used for the trees and the sample data:

- SetID

Includes three setIDs: EUR, SHARE, and USA.

- Record groups

Includes one record group: CUBETEST.

- Tableset controls

Includes some tableset rows for set IDs: EUR, MODEL, SHARE, and USA.

Both PS_CUBEBUILDER.DAT and PS_CUBEBUILDER.DMS files are delivered as part of PeopleTools.

Note: The data mover file (PS_CUBEBUILDER.DMS) contains the *IGNORE_DUPS* option, which imports the data and will not overlay any possible user data that has the same keys. When data mover contains this *IGNORE_DUPS* option, it works only if you log into Data Mover in bootstrap mode.

Granting Query Security

To grant query security:

1. Select PeopleTools > Security > Query Security > Query Access Manager.

2. Open the tree *QUERY_TREE_PT*.
3. Add the following records under the *OLAP - Cube Manager Record* node:
 - *PSCUBTTRANS*
 - *PSCUBTSELLER*
 - *PSCUBTREGION*
 - *PSCUBTPRODUCT*
 - *PSCUBTACCOUNT*
 - *PSCUBTTRANSD_VW*
4. View the queries that were imported from the project.

Note that:

- Query *CUBT_ACCOUNTS* can be used to build a dimension structure.
- Queries *CUBT_TRANSACTIONS* and *CUBT_ALL_TRANSACTIONS* can be used to populate the cube structure.
- Query *CUBT_TRANSDATES* can be used to create a time dimension, and it displays all unique transaction dates from the transaction table.

Adding Data to Tables

You can add data to the tables after you apply project *CUBE_DEMO_PROJ*, grant permissions to all menu *CUBE_TEST* components, import *.DAT* file, and grant query security.

The *PSCUBTTRANS* table contains all the transactional data that is used to populate the cube. The other tables contain definitional data with:

- Sellers—*PSCUBTSELLER*
- Regions—*PSCUBTREGION*
- Products—*PSCUBTPRODUCT*
- Accounts—*PSCUBTACCOUNT*

All tables are accessible from the new components.

Using Sample Dimensions and Outlines

These sample dimensions are also included in the project *QE_CUBE_DEMO_PROJ*:

- *QE_CUBT_ACCOUNTS*

- QE_CUBT_ALL_REGIONS
- QE_CUBT_EUROPEAN_PRODUCTS
- QE_CUBT_TRANSACTION_DATES

The sample outline QE_CUBT_OUTLINE makes use of the preceding dimensions. It also uses query QE_CUBT_ALL_TRANSACTIONS as the query to populate the outline.