
Oracle's PeopleTools PeopleBook

PeopleTools 8.52: PeopleSoft Cube Builder

October 2011

Copyright © 1988, 2011, Oracle and/or its affiliates. All rights reserved.

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions Warranty/Consequential Damages Disclaimer

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.

Warranty Disclaimer

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.

Restricted Rights Notice

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software and documentation may provide access to or information on 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. 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.

Contents

Preface

PeopleSoft Cube Builder Preface	vii
PeopleSoft Cube Builder	vii
PeopleBooks and the PeopleSoft Online Library	vii

Chapter 1

Getting Started with PeopleSoft Cube Builder	1
PeopleSoft Cube Builder Overview	1
Essbase Cube Builder Business Processes	1
Other Sources of Information	2
Essbase Cube Builder Implementation	2

Chapter 2

Designing Cube Metadata	5
Understanding Cubes	5
Hierarchy	7
Understanding PeopleSoft Metadata	9
Understanding Metadata Types	10
Understanding Oracle Essbase Properties	10
Understanding PeopleSoft Process Scheduler Integration	11
Using Trees	11
Creating Queries	12
Understanding Query Types	12
Using Dimension Queries	13
Using Data Source Queries	13

Chapter 3

Defining Essbase Cube Builder Connection Definitions	21
Defining Essbase Cube Builder Connections	21

Chapter 4

Creating Dimensions for Essbase Cubes	25
Understanding Dimensions of Essbase Cubes	25
Setting Up Cube Dimension Builder	25
Creating Essbase Dimensions	27
Pages Used to Create Essbase Dimensions	27
Defining an Essbase Dimension	28
Inserting a Tree Structure into the Dimension	38
Inserting a Query into the Dimension	44
Defining Time Dimensions	52
Adding Generation Names	55

Chapter 5

Defining Essbase Cube Outlines	57
Defining Cube Outlines	57
Pages Used to Define Cube Outlines	57
Adding Cube Dimensions to a Cube Outline	58
Adding Cube Data Queries to a Cube Outline	65

Chapter 6

Creating Dynamic Dimension Templates	73
Understanding Dynamic Dimension Templates	73
Creating Dynamic Dimension Templates	73

Chapter 7

Building Essbase Cubes	77
Building Essbase Cubes	77
Creating Essbase Cubes	77
Integrating with Oracle Smart View	80
Using API and Command-line Specifications	82

Chapter 8

Securing Cubes	87
Securing Cube Builder Cubes	87
Pages Used to Secure Cube Builder Cubes	87
Controlling User Operations	88
Setting User Actions	90
Viewing Content References	91

Chapter 9

Maintaining Essbase Cubes	93
Maintaining Essbase Cubes	93
Pages Used to Maintain Essbase Cubes	93
Deleting Dimensions	94
Deleting Outlines	95
Deleting Run Control IDs	96
Viewing Dynamic Objects	97
Viewing Outline Structures	98
Expanding Dimensions	98

Appendix A

Viewing Query Results That Have Large Result Sets	101
Query Chunking and the Chunking Class	101
Dynamic Query Expansion	103

Appendix B

Using Essbase Cube Builder Demo Data	105
Understanding Essbase Cube Builder Demo Data	105
Applying Project CUBE_DEMO_PROJ	105
Granting Permissions to All Menu CUBE_TEST Components	106
Importing PS_CUBE_BUILDER.DAT File Using Data Mover Script PS_CUBE_BUILDER.DMS	106
Granting Query Security	107
Adding Data to Tables	107
Using Sample Dimensions and Outlines	108

Index 109

PeopleSoft Cube Builder Preface

This PeopleBook refers to PeopleSoft Cube Builder.

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.

PeopleBooks and the PeopleSoft Online Library

A companion PeopleBook called *PeopleBooks and the PeopleSoft Online Library* contains general information, including:

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.
- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.
- How to manage the locally installed PeopleSoft online library, including web site folders.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Application abbreviations found in application fields.

You can find *PeopleBooks and the PeopleSoft Online Library* in the online PeopleBooks Library for your PeopleTools release.

Chapter 1

Getting Started with PeopleSoft Cube Builder

This chapter provides an overview of PeopleSoft Cube Builder and discusses:

- PeopleSoft Cube Builder business processes.
- Other sources of information.
- Essbase Cube Builder implementation.

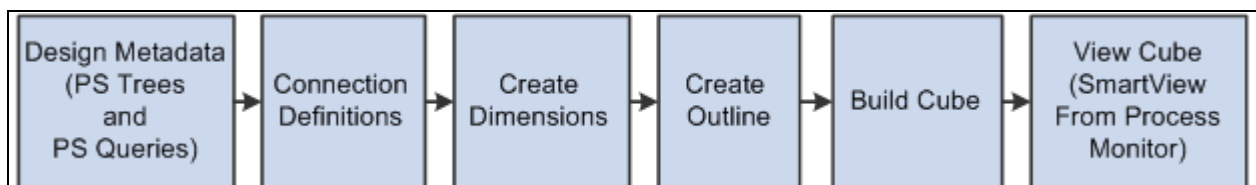
PeopleSoft Cube Builder Overview

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. 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

The following process flow illustrates the Cube Builder business processes:



Essbase Cube Builder business process flow

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

Other Sources of Information

This section 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, PeopleBooks, red papers, the Updates and Fixes area of My Oracle Support, and curriculum courses of Oracle PeopleSoft.

See ["PeopleSoft Cube Builder Preface," page vii.](#)

See *PeopleTools 8.52: Getting Started with PeopleTools*.

Essbase Cube Builder Implementation

Essbase Cube Builder implementation can be divided into the following phases:

- Designing cube metadata.
- Defining connection definitions.
- Creating dimensions.
- Creating an outline.
- Building cubes.

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. To design cube metadata, perform the following steps:

Step	Reference
Create new trees or use existing trees.	See <i>PeopleTools 8.52: PeopleSoft Tree Manager</i> , "Using PeopleSoft Tree Manager" and <i>PeopleTools 8.52: PeopleSoft Tree Manager</i> , "Creating Trees."
Create new queries or use existing queries.	See <i>PeopleTools 8.52: PeopleSoft Query</i> , "Creating and Running Simple Queries," Creating New Queries.

Defining Connection Definitions

The second phase of creating Essbase cubes is to create connectivity information. To set up the Essbase server connection information, perform the following step:

Step	Reference
Define Essbase Cube Builder connection definitions.	See Chapter 3, "Defining Essbase Cube Builder Connection Definitions," page 21.

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.

To create dimensions, perform the following steps:

Step	Reference
Set up cube dimension builder.	See Chapter 4, "Creating Dimensions for Essbase Cubes," page 25.
Create Essbase dimensions.	See Chapter 4, "Creating Dimensions for Essbase Cubes," Creating Essbase Dimensions, page 27.

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.

To create an outline, perform the following steps:

Step	Reference
Define cube outlines.	See Chapter 5, "Defining Essbase Cube Outlines," page 57.
Add cube dimensions to a cube outline.	See Chapter 5, "Defining Essbase Cube Outlines," Adding Cube Dimensions to a Cube Outline, page 58.
Add cube data queries to a cube outline.	See Chapter 5, "Defining Essbase Cube Outlines," Adding Cube Data Queries to a Cube Outline, page 65.

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.

To build Essbase cubes, perform the following step:

<i>Step</i>	<i>Reference</i>
Build Essbase cubes.	See Chapter 7, "Building Essbase Cubes," page 77.

Chapter 2

Designing Cube Metadata

This chapter provides overviews of cubes, PeopleSoft metadata, metadata types, Oracle Essbase properties, and PeopleSoft Process Scheduler Integration, and discusses how to:

- Use trees.
- Create queries.

See Also

PeopleTools 8.52: PeopleSoft Tree Manager, "Using PeopleSoft Tree Manager"

PeopleTools 8.52: PeopleSoft Query, "Creating and Running Simple Queries"

Understanding Cubes

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

Product	2001	2002	2003
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 preceding table 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 this diagram:

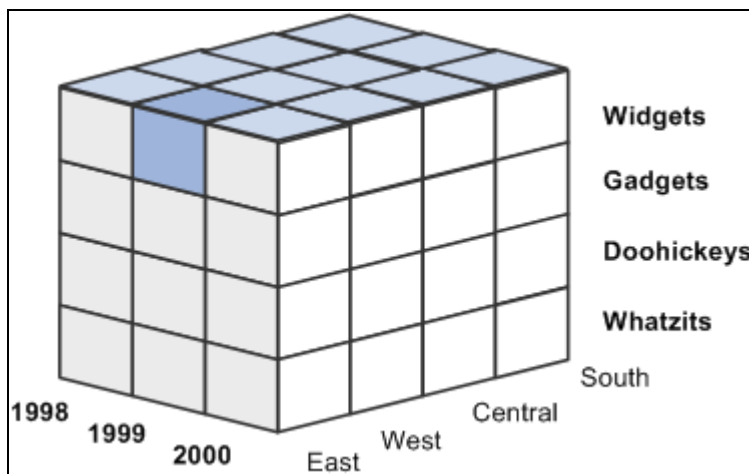


Diagram of a cube with three dimensions

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

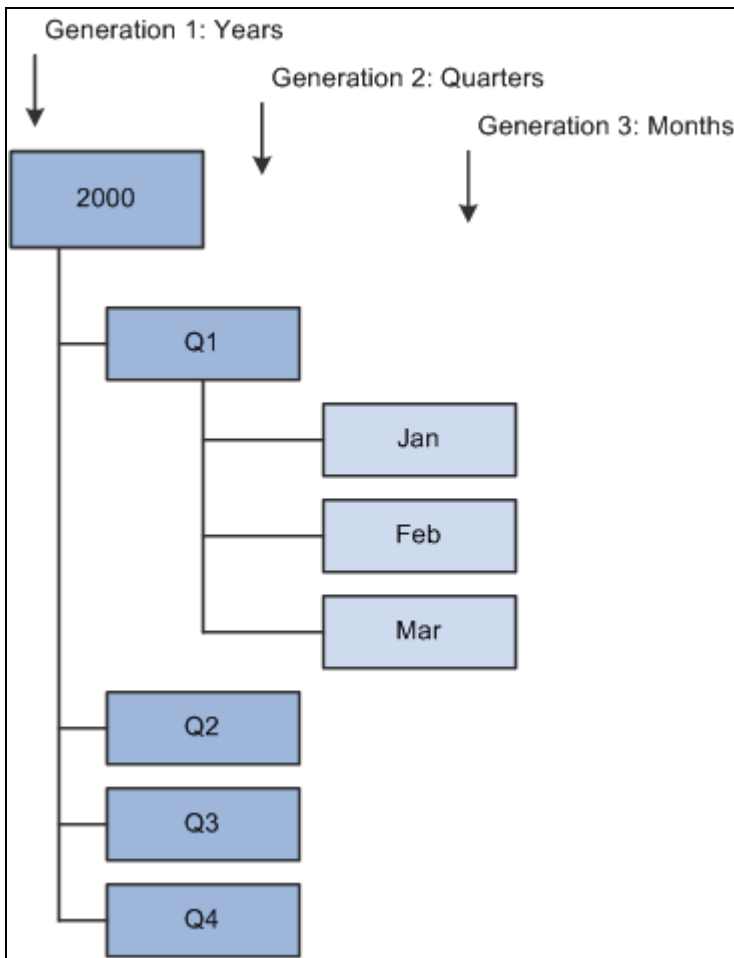
Hierarchy is the organization of cube data elements with their reporting structures. It represents both the hierarchy and the method of consolidation in a dimension level.

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.

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



Example of a time hierarchy diagram

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 *PeopleTools 8.52: PeopleSoft 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.)

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

Consolidation example in tabular format

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.

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 *PeopleTools 8.52: PeopleSoft 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 *PeopleTools 8.52: PeopleSoft 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.

Valid Value	Action
* (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.

See Also

Oracle Essbase documentation.

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 Also

[Chapter 7, "Building Essbase Cubes," page 77](#)

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

This section provides an overview of query types and discusses how to:

- Use dimension queries.
- Use data source queries.

Understanding Query Types

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.

See [Chapter 2, "Designing Cube Metadata," Using Dimension Queries, page 13](#) and [Chapter 2, "Designing Cube Metadata," Using Data Source Queries, page 13](#).

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 this table:

<i>Account (Dimension)</i>	<i>Department (Dimension)</i>	<i>Period (Dimension)</i>	<i>Amount (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:

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

Results of Query 2:

Account (Dimension)	Department (Dimension)	Period (Dimension)	Actual Amount (Measure 2)
1000	DEV	Q4 2003	3000
1100	SALES	Q4 2003	5000

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

Account (Dimension)	Department (Dimension)	Period (Dimension)	Budget Amount (Measure 1)	Actual Amount (Measure 2)
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)

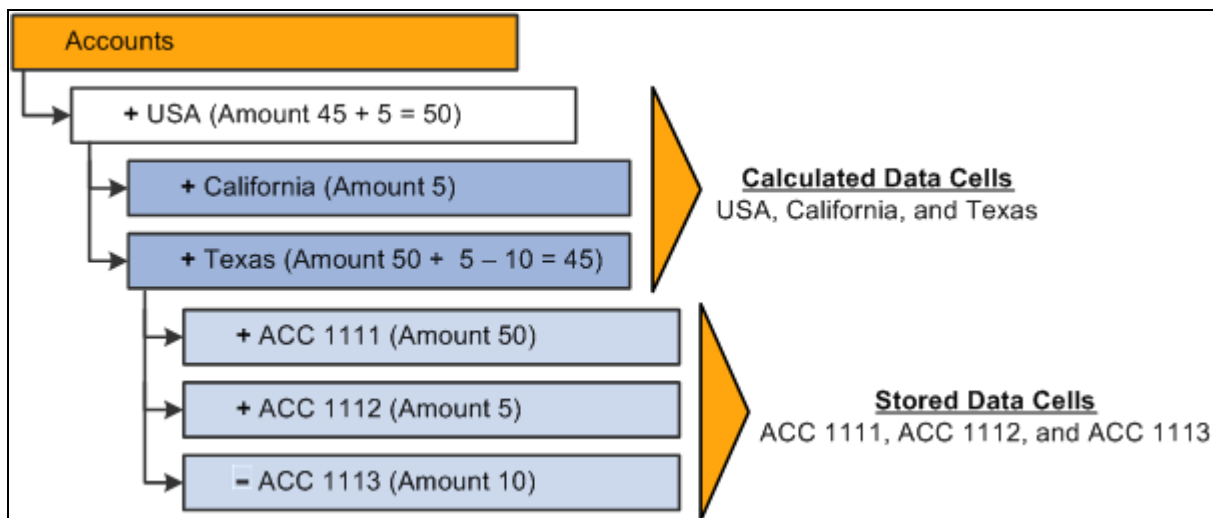
Calculated data cells store values of dimensional intersections of non-level zero members (branch nodes), which are calculated based on the consolidation properties of their children.

- 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.

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



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

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.

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

Create Cubes

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

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 07/07/2011 12:16:11PM

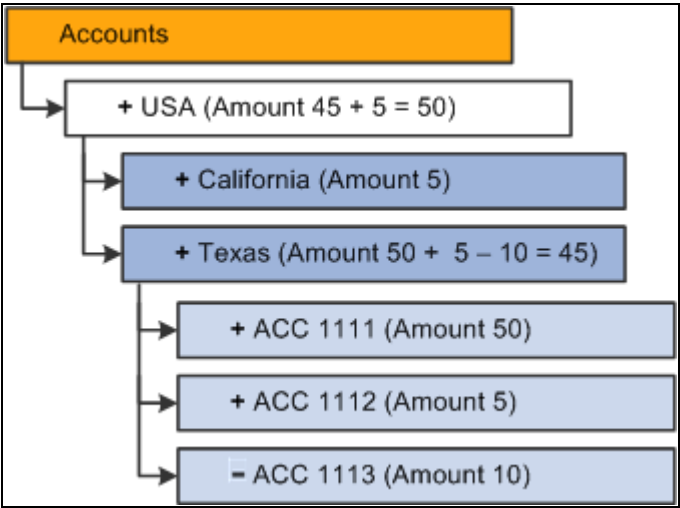
Create Cubes page with the Aggregations on Load options of Replace, Addition, and Subtraction

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

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



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

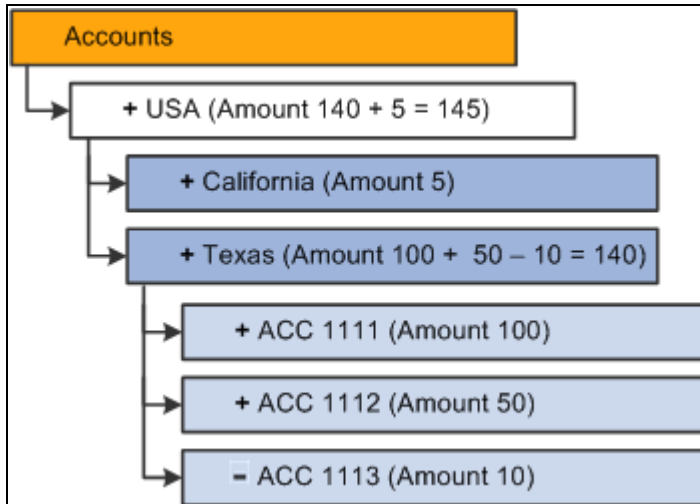
This example shows how to create data queries for each Aggregations on Load option, taking as a reference the next staging table that contains the data cells to be loaded:

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

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

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

1. If the *Replace* (replace data load) option is selected, this is the resulting dimension:



Results of replacing the existing data cells

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 *ReplaceAggregations 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, as in this example:

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

Example of summarizing repeated fields into one field

The SQL query is:

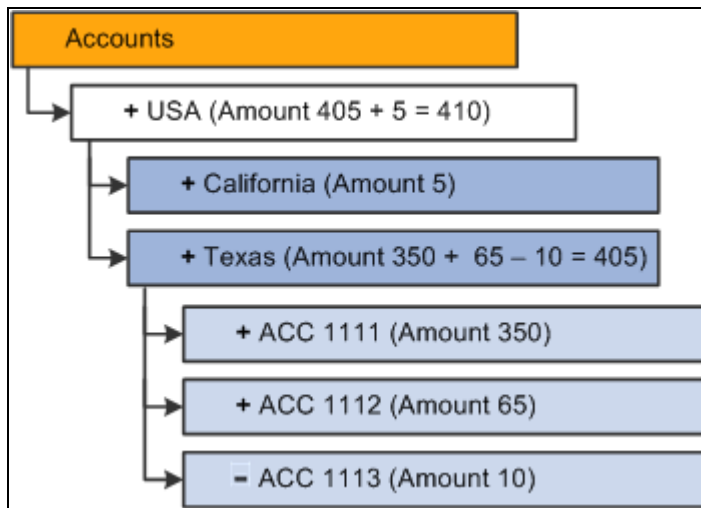
```

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

```

SQL query, summarizing repeated fields into one field

The resulting dimension is:

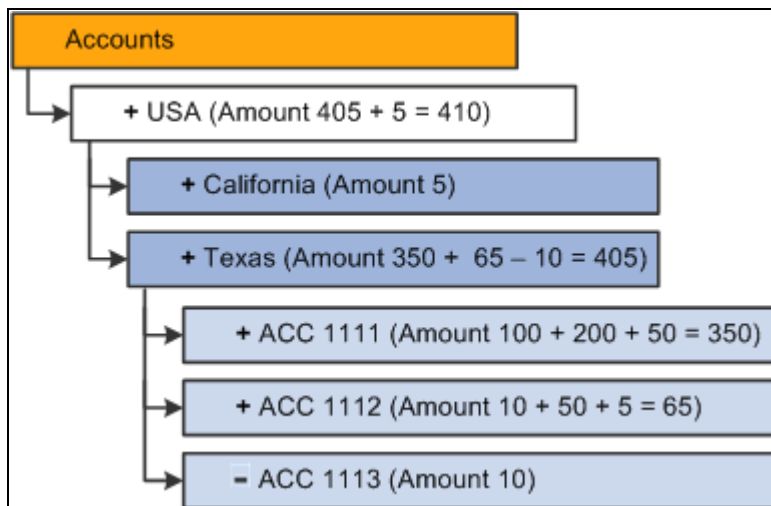


Results of replacing incremental updates

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.

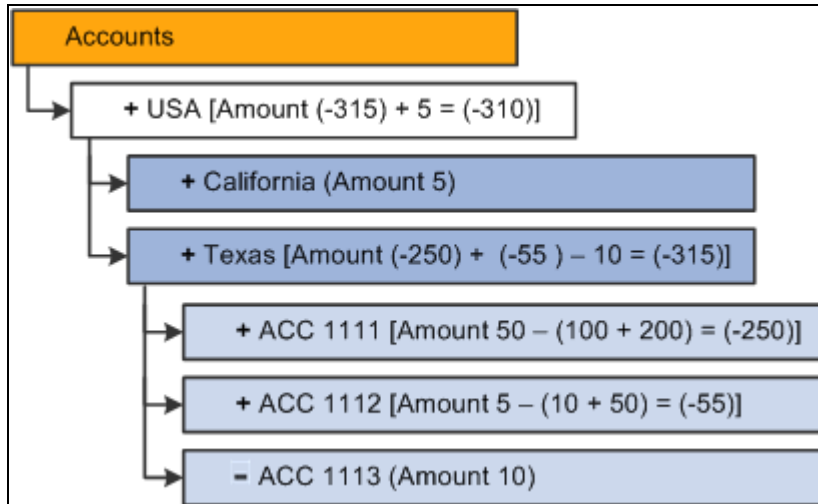
2. If the *Addition* (addition data load) option is selected, this is the resulting dimension:



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

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

3. If the *Subtraction* (subtraction data load) option is selected, this is the resulting dimension:



Results of subtractive incremental updates: 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

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 the 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.

Chapter 3

Defining Essbase Cube Builder Connection Definitions

This chapter discusses how to define Essbase Cube Builder connections.

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.

Access the Connections page by selecting PeopleTools, Cube Builder, Connections.

Connections

Essbase Connection ID **QE_CUBT_SMOKE**

*Description

Comment

Owner ID

*Server

*Database State

*User Name

*Password

Unicode Enabled: ☒

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

Connections page

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*
- *PeopleTools Demo*

Server

Enter the Essbase network server name or IP address.

Database State	<p>Set the default user access level to the database after it is created or modified.</p> <p>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:</p> <ul style="list-style-type: none"> • <i>Calculate</i> • <i>Design</i> • <i>None</i> • <i>Read</i> • <i>Write</i> <p>Oracle recommends that you set this option to <i>Read</i> at the least.</p>
User Name	Enter the Essbase user name.
Password	<p>Enter the Essbase password, which will be encrypted in the database table.</p> <hr/> <p>Note. Only users with the following permission lists are able to create the Essbase connectivity information: AEPNLS, ALLPAGES, CPPT1080, PTPT1200, and SMPPNLS.</p> <hr/>
Unicode Enabled	<p>Specify whether the database to be created will contain Unicode characters.</p> <hr/> <p>Note. Because selecting this option creates an Essbase Unicode database, the Essbase server must be in Unicode mode.</p> <p>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.</p> <hr/>
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

This chapter provides an overview of Essbase Cubes dimensions and discusses how to:

- Set up cube dimension builder.
- Create Essbase dimensions.

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

Access the Cube Dimension Builder Setup page (CUB_GUISETUP) by selecting PeopleTools, Cube Builder, Cube Dimension Builder Setup.

Cube Dimension Builder Setup

Dynamic Query Frame Size:

20

Static Query Maximum Rows:

10000

Last Updated:PTDOCKN

Last Update Date/Time:07/07/11 12:31:12PM

Cube Dimension Builder Setup page

Use the Cube Dimension Builder Setup page 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.

- 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 section discusses how to:

- Define an Essbase dimension.
- Insert a tree structure into the dimension.
- Insert a query into the dimension.
- Define time dimensions.
- Add generation names.

Pages Used to Create Essbase Dimensions

Note. Only users with these permission lists are able to create Essbase dimensions: AEPNLS, ALLPAGES, CPPT1080, PTPT1200, and SMPPNLS.

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Dimension	CUB_DIMENSION	PeopleTools, Cube Builder, Dimensions	Define an Essbase dimension.
Child Member Add	CUB_DIMENSION_SEC	PeopleTools, Cube Builder, Dimensions On the Dimension page, highlight a tree member, and click the Add Member Child icon.	Use on initial member creation to enter the initial dimension name and consolidation operator. Add a new member that is a child of the currently highlighted tree member.
Member Formula	CUB_FORMULA_SEC	PeopleTools, Cube Builder, Dimensions On the Dimension page, click the Formula link.	Enter formula information for a dimension member.
Find Tree	CUB_TREESEARCH_SEC	PeopleTools, Cube Builder, Dimensions On the Dimension page, click the Assign From Tree icon.	Search for and select a tree that is defined in the system.

Page Name	Definition Name	Navigation	Usage
Select Top Node	CUB_TREENODE_SEC	PeopleTools, Cube Builder, Dimensions On the Dimension page, click the Assign From Tree icon. On the Find Tree page, select trees and click the OK button.	Select an existing node of a highlighted tree.
Find Query	CUB_QRYSEARCH_SEC	PeopleTools, Cube Builder, Dimensions On the Dimension page, click the Assign From Query icon.	Select a query that is defined in the system.
Select a Column	CUB_QRYCOLUMN_SEC	PeopleTools, Cube Builder, Dimensions On the Dimension page, click the Assign From Query icon. On the Find Query page, select queries and click the OK button.	Select an existing column of a highlighted query. The columns displayed are from the select statement of the query.
Time Dimension Wizard	CUB_TIMEDIMWIZ	PeopleTools, Cube Builder, Dimensions On the Dimension page, select the Time Dimension Wizard icon.	Define the date and time ranges of the time dimension.
Generation Names	CUB_DIMENSION2	PeopleTools, Cube Builder, Dimensions On the Dimension page, select the Generation Names tab.	Enter the generation names for the Essbase dimension.

Defining an Essbase Dimension

Access the Dimension page by selecting PeopleTools, Cube Builder, Dimensions.

This is an example of the Dimension page when you first access it to create a dimension:

Dimension **Generation Names**

Dimension Name **PRODUCT**

Dimension Type **Accounts**

Owner ID

*Storage **Default**

Description

Comments

Dimension Structure

Left | Right

+ **PRODUCT**

Save

Dimension page when you first access it to create a 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.

This is an example of the Dimension page with some properties and active icons:

Dimension

Generation Names

Dimension Name

QE_CUBT_EURO_PRODUCTS

Dimension Type

None

Owner ID

PT

*Storage

Default

Description

Standard Dimension

Comments

Dimension Structure

Left | Right

QE_CUBT_EURO_PRODUCTS

TOP_NODE

VAT_FREE - (+)(VAT Free Products)]

EP602 - (+)]

EP603 - (+)]

EP605 - (+)]

HIGH_VALUE_PRODUCTS - (+)(High Value Products)]

Member Properties

*Member Name

TOP_NODE

Label Only

Never Share

Alias

*Storage

Default

Dynamic Source

Tree

QE_CUBT_PRODUCTS

SetID

QEEU

Set Control Value

Effective Date

02/27/2009

Template ID

Dimension page

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.

30

Copyright © 1988, 2011, Oracle and/or its affiliates. All Rights Reserved.

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 Insert 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.

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.

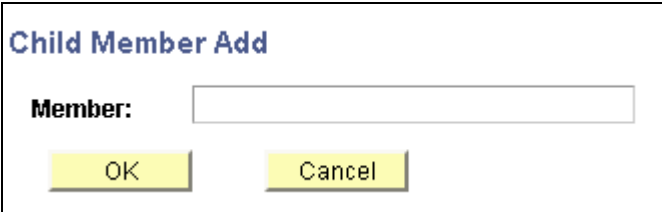
Dynamic Source

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.

Adding a Child or a Peer Member

Access the Child Member Add page (on the Dimension page, highlight a tree member and click the Add Member Child icon).



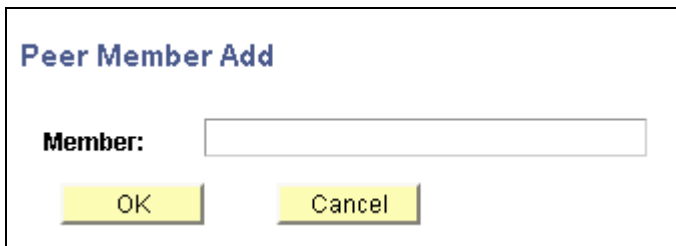
Child Member Add

Member:

OK Cancel

Child Member Add page

Access the Peer Member Add page (on the Dimension page, highlight a tree member and click the Insert Member Peer icon).

The screenshot shows a dialog box titled "Peer Member Add" in blue text. Below the title, there is a label "Member:" followed by a single-line text input field. At the bottom of the dialog, there are two buttons: "OK" and "Cancel", both with yellow backgrounds and black text.

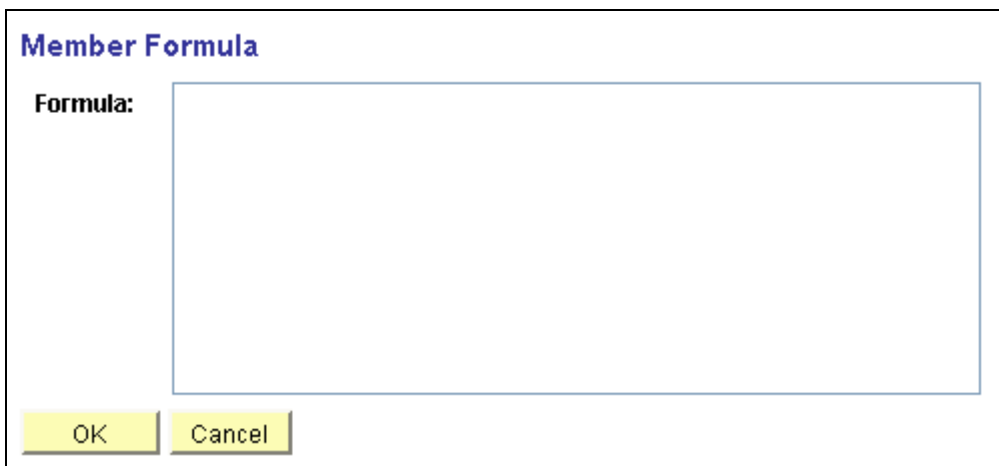
Peer Member Add page

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

Access the Member Formula page (on the Dimension page, click the Formula link).

The screenshot shows a dialog box titled "Member Formula" in blue text. Below the title, there is a label "Formula:" followed by a large, empty rectangular text area for entering a formula. At the bottom of the dialog, there are two buttons: "OK" and "Cancel", both with yellow backgrounds and black text.

Member Formula page

Formula Enter a formula that will be used to calculate aggregates for the specific member.

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 (select PeopleTools, Cube Builder, Dimensions) and add a dimension with a name of PRODUCT.

This is an example of the Dimension page when you first create a dimension:

The screenshot shows the 'Dimension' page in the Essbase administration tool. The 'Dimension' tab is selected, and the 'Generation Names' sub-tab is also visible. The form contains the following fields:

- Dimension Name:** PRODUCT
- Dimension Type:** Accounts
- Owner ID:** PT (selected from a dropdown)
- *Storage:** Default (selected from a dropdown)
- Description:** Standard Dimension
- Comments:** An empty text area.

Below the form is the 'Dimension Structure' section, which has a 'Left | Right' toggle. The structure tree shows a single root node labeled 'PRODUCT' with a plus icon to its left. At the bottom of the window, there is a row of icons for various actions (like copy, paste, delete, etc.) and a yellow 'Save' button.

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

The screenshot displays the 'Dimension' page with the 'Generation Names' tab selected. The main form contains the following fields:

- Dimension Name: PRODUCT
- Dimension Type: Accounts
- Owner ID: PT (dropdown)
- *Storage: Default (dropdown)
- Description: Standard Dimension
- Comments: (empty text area)

Below the main form are two panels:

- Dimension Structure:** Shows a tree view with 'PRODUCT' expanded, revealing a child member 'M1'.
- Member Properties:** Shows the properties for the selected member 'M1':
 - *Member Name: M1
 - ☐ Expense Report
 - ☐ Never Share
 - *Time Balance: None (dropdown)
 - Alias: (empty text field)
 - ☐ Shared Member
 - *Storage: Default (dropdown)
 - [Formula](#) (link)

At the bottom of the page, there is a row of icons for various actions (add, delete, etc.) and a 'Save' button.

Example of the Dimension page with M1 member and its properties

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.

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:

Dimension

Generation Names

Dimension Name

PRODUCT

Dimension Type

Accounts

Owner ID

PT

*Storage

Default

Description

Standard Dimension

Comments

Dimension Structure

Left | Right

PRODUCT

M1

VAT FREE

EP602 - (+)

EP603 - (+)

EP605 - (+)

Save

Member Properties

*Member Name

VAT_FREE

Label Only

Expense Report

Never Share

*Time Balance

None

Alias

*Storage

Default

Formula

Dynamic Source

Tree

QE_CUBT_PRODUCTS

SetID

QEEU

Set Control Value

Effective Date

02/27/2009

Template ID

Example of the attributes of the member VAT_FREE

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

Access the Find Tree page (on the Dimension page, select a tree name and click the Assign From Tree icon).

Find Tree

Tree:
☒ Dynamic

Template ID:

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

Find Tree page

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 [Chapter 6, "Creating Dynamic Dimension Templates," page 73](#).

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

Selecting the Top Node

Access the Select Top Node page (on the Dimension page, select a tree name and click the Assign From Tree icon; on the Find Tree page, select trees and click the OK button).

Select Top Node

Tree: QE_CUBT_PRODUCTS

Select	Tree Node
<input type="checkbox"/>	TOP_NODE
<input type="checkbox"/>	VAT_FREE
<input type="checkbox"/>	HIGH_VALUE_PRODUCTS

OK

Cancel

Select Top Node page

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, as shown in this example:

The screenshot displays the 'Dimension' page in Essbase. The top section, titled 'Dimension', contains the following fields:

- Dimension Name: **PRODUCT**
- Dimension Type: **Accounts**
- Owner ID: **PT**
- *Storage: **Default**
- Description: **Standard Dimension**
- Comments: (Empty text box)

Below the 'Dimension' section are two panels:

- Dimension Structure:** Shows a tree view with 'PRODUCT' as the root. Under 'PRODUCT', there are two nodes: 'VAT Ready Products' and 'US Tax Rate Products'. A child member 'T1 - (+)' is added under 'VAT Ready Products'.
- Member Properties:** Shows settings for the selected member 'T1'. The settings include:
 - *Member Name: **T1**
 - ☐ Expense Report
 - ☐ Never Share
 - *Consolidation: **+**
 - *Time Balance: **None**
 - Alias: (Empty text box)
 - ☐ Shared Member
 - *Storage: **Default**
 - Formula: (Link)


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

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. This example shows the Find Tree page:

Find Tree

Tree: ☒ Dynamic

Template ID: 

Available Trees				
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

Example of 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.

This is an example of the Select Top Node page:

Select Top Node

Tree: QE_CUBT_PRODUCTS

Available Tree Nodes	
Select	Tree Node
<input type="checkbox"/>	TOP_NODE
<input type="checkbox"/>	VAT_FREE
<input type="checkbox"/>	HIGH_VALUE_PRODUCTS

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

Dimension | **Generation Names**

Dimension Name **PRODUCT**

Dimension Type **Accounts**

Owner ID **PT**

*Storage **Default**

Description **Standard Dimension**

Comments

Dimension Structure

Left | Right

- PRODUCT**
 - VAT Ready Products
 - VAT FREE - (+)**
 - EP602 - (+)
 - EP603 - (+)
 - EP605 - (+)
 - US Tax Rate Products

Member Properties

*Member Name **VAT_FREE**

☐ Label Only

☐ Expense Report

☐ Never Share

*Consolidation **+**

*Time Balance **None**

Alias

*Storage **Default**

[Formula](#)

Dynamic Source

Tree **QE_CUBT_PRODUCTS**

SetID **QEEU**

Set Control Value

Effective Date **02/27/2009**

Template ID

Save

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

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

Access the Find Query page (on the Dimension page, highlight a tree member and click the Assign From Query icon).

Find Query

Query
☐ Dynamic
☒ Rollup

Search

Select	Query Name
<input type="checkbox"/>	QE_CUBT_ACCOUNTS
<input type="checkbox"/>	QE_CUBT_COUNTRY
<input type="checkbox"/>	QE_CUBT_SELLER
<input type="checkbox"/>	QE_CUBT_SINGLE_PROMPT
<input type="checkbox"/>	QE_CUBT_TRANSACTIONS
<input type="checkbox"/>	QE_CUBT_TRANSDATES
<input type="checkbox"/>	QE_CUBT_TRANS_REGION

OK

Cancel

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 [Chapter 6, "Creating Dynamic Dimension Templates," page 73](#).

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

Access the Select column page (on the Dimension page, highlight a tree member and click the Assign From Query icon; on the Find Query page, select queries and click the OK button).

Select column

Available Query Columns

Personalize | Find | View All | | First 1-4 of 4 Last

Select	Heading	Rollup Level	Alias
<input checked="" type="checkbox"/>	A.SETID	1	A.SETID
<input checked="" type="checkbox"/>	A.QE_CUBT_ACCOUNT	2	A.DESCR
<input type="checkbox"/>	A.QE_CUBT_ACC_TTYPE		
<input type="checkbox"/>	A.DESCR		

OK

Cancel

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

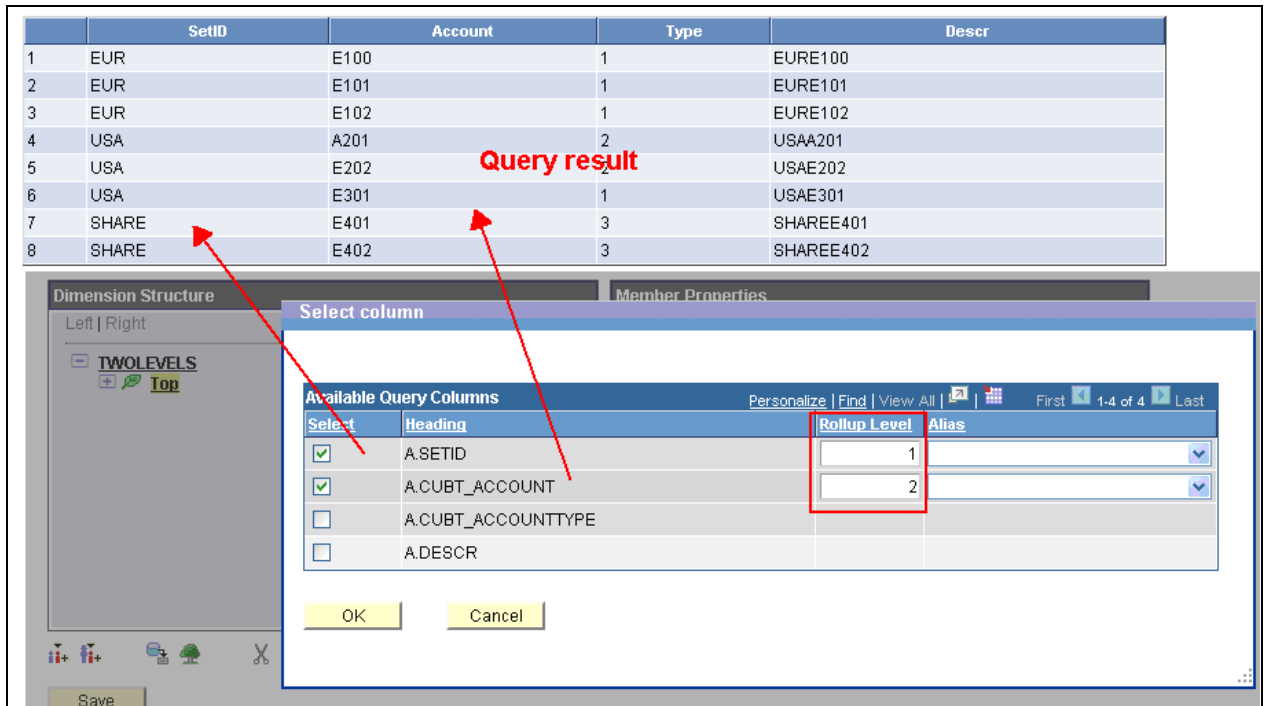
Optionally, select one or more columns from the query to be used to populate the alias for the selected member.

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.

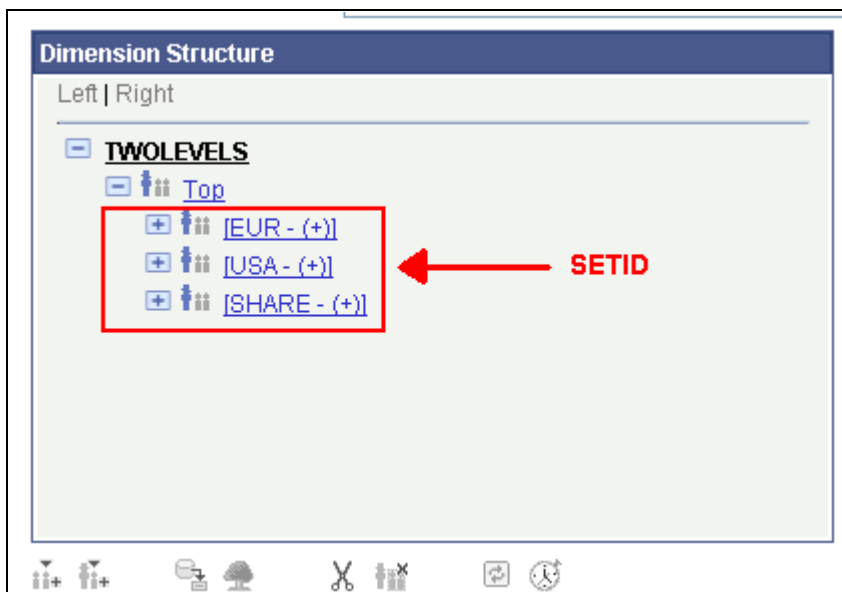
The following examples demonstrate the use of a wide query and rollup levels that define the hierarchical structure of a dimension:

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



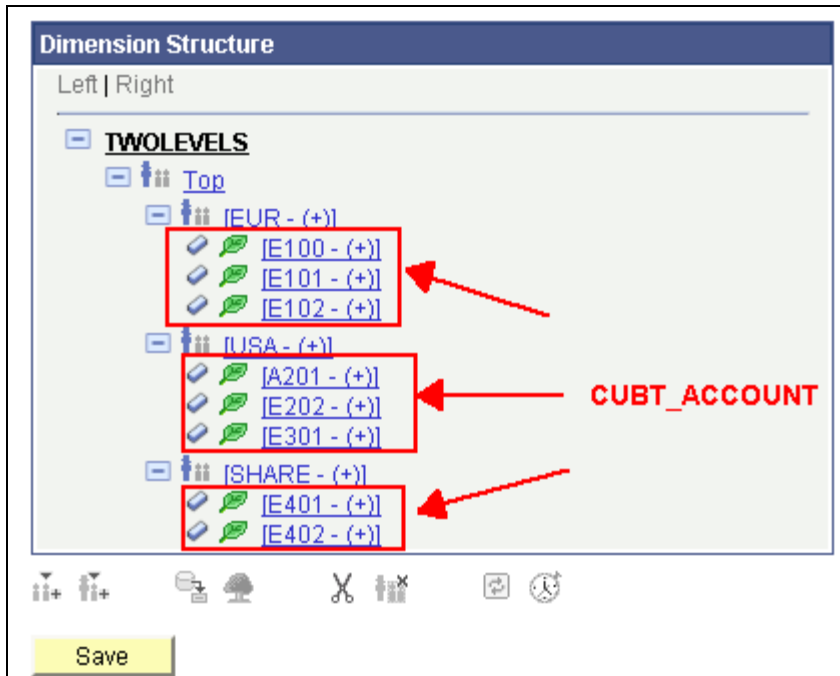
Setting the rollup levels

- 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:



Example of rollup level 1

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



Example of rollup level 2

Example: Inserting a Query into the Dimension

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

Dimension

Generation Names

Dimension Name

ACCOUNTS

Dimension Type

Accounts

Owner ID

*Storage

Default

Description

Comments

Dimension Structure

Left | Right

ACCOUNTS

Account Query

Member Properties

*Member Name

Account Query

Expense Report

Never Share

*Time Balance

None

Alias

Shared Member

*Storage

Default

Formula

Save

Example of the Dimension page with Account Query member

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

Find Query

Query ☒ Dynamic

Template ID ☒ Rollup

Select	Query Name
<input type="checkbox"/>	QE_CUBT_ACCOUNTS
<input type="checkbox"/>	QE_CUBT_COUNTRY
<input type="checkbox"/>	QE_CUBT_SELLER
<input type="checkbox"/>	QE_CUBT_SINGLE_PROMPT
<input type="checkbox"/>	QE_CUBT_TRANSACTIONS
<input type="checkbox"/>	QE_CUBT_TRANSDATES
<input type="checkbox"/>	QE_CUBT_TRANS_REGION

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:

Select column

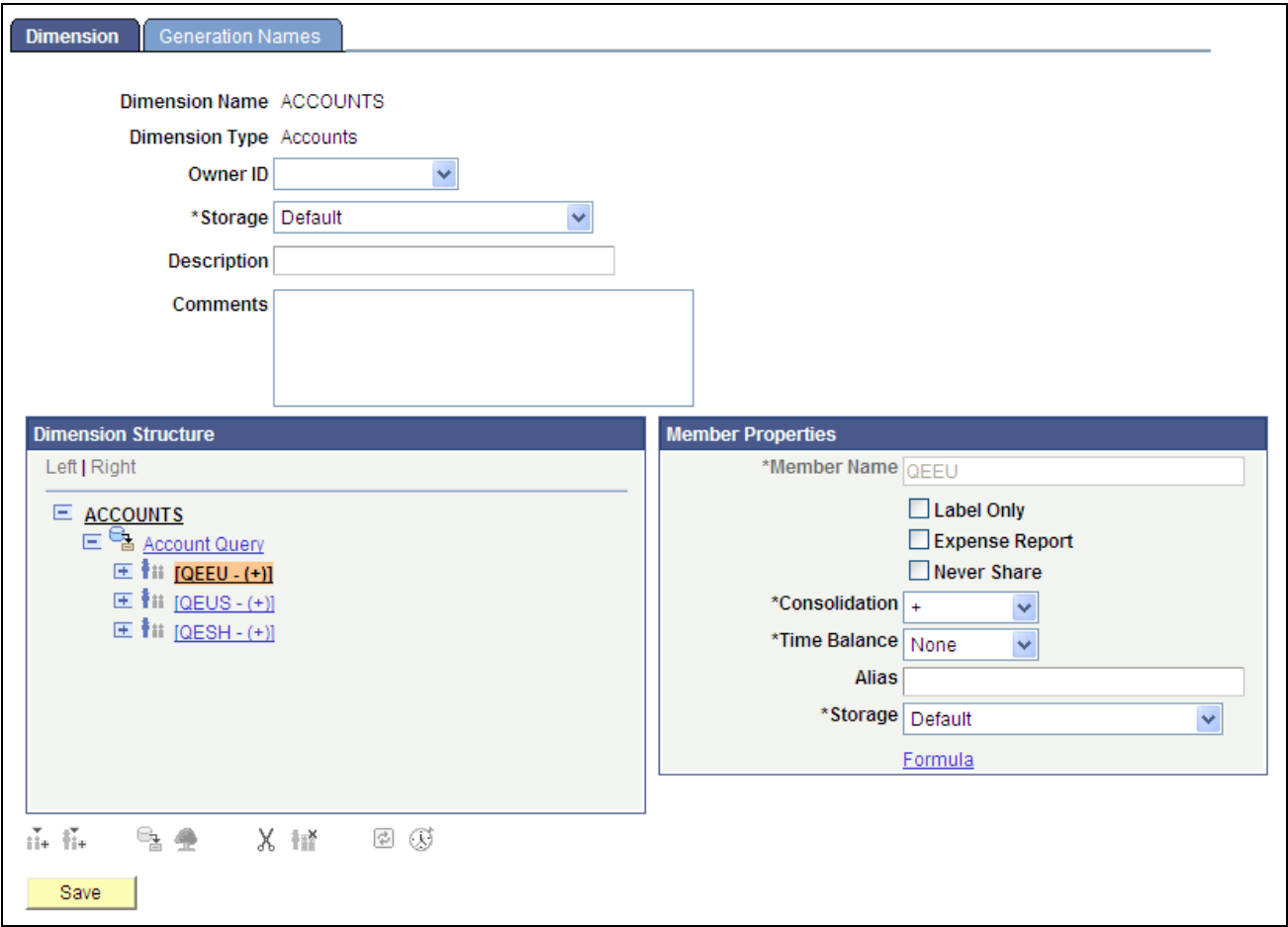
Available Query Columns			
Select	Heading	Rollup Level	Alias
<input checked="" type="checkbox"/>	A.SETID	1	
<input checked="" type="checkbox"/>	A.QE_CUBT_ACCOUNT	2	A.DESCR
<input type="checkbox"/>	A.QE_CUBT_ACC_TTYPE		
<input type="checkbox"/>	A.DESCR		

OK Cancel

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, as shown in this example:



Example of 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

Access the Time Dimension Wizard page (on the Dimension page, click the Time Dimension Wizard icon).

Time Dimension Wizard

Modeling Period

*First Day of the Week

1 - Sunday

*Begin Date

07/12/2011

*End Date

07/12/2011

Time Depths

☒ Year

☐ Period

☐ Semester

☒ Month

☐ Trimester

☒ Week

☒ Quarter

☒ Day

Year Semantic Rules

Year Starts On:

*Month

01 - January

*Day

01

Create Time Dimension

Cancel

Time Dimension Wizard page

Use the Time Dimension Wizard page to automatically populate the members of a time dimension.

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

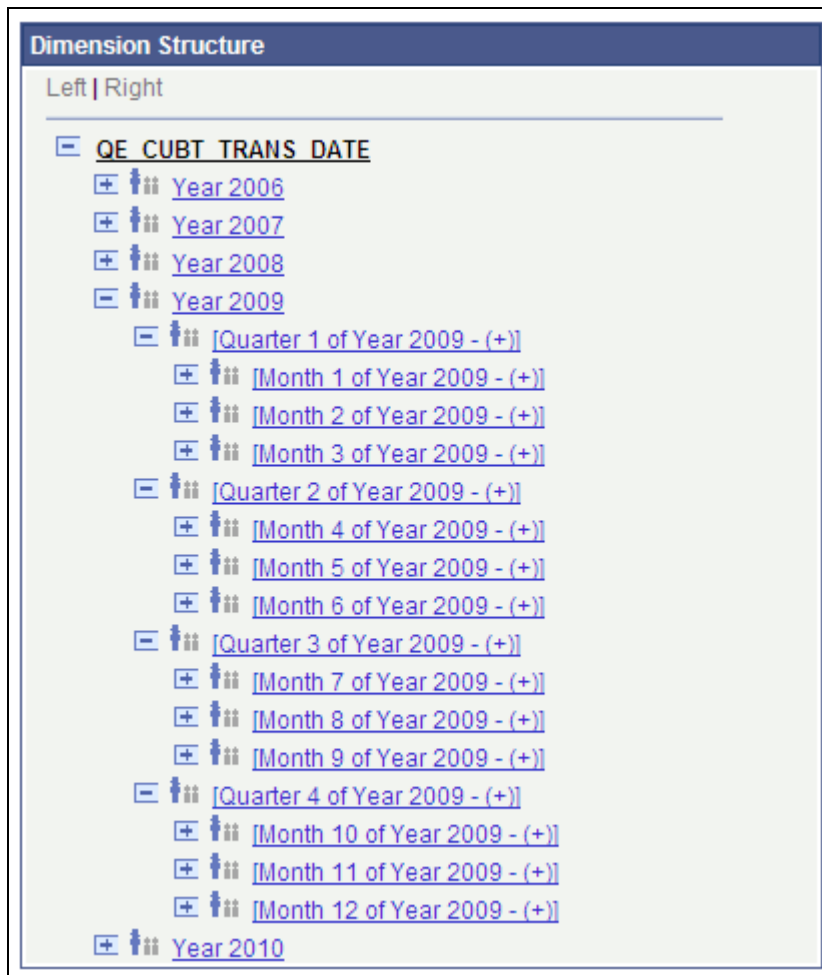
First Day of the Week	Select the first day of the calendar week on which the intended calendar starts.
Begin Date	Select the begin date for the time range for which members are created.
End Date	Select the end date for the time range for which members are created.
Time Depth	Select the option next to 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, as shown in this example:



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

Access the Generation Names page (on the Dimension page, select the Generation Names tab).

Dimension

Generation Names

Dimension: PRODUCT

Generation Names			
Generation	Generation Name		
1	WORLD	+	-
2	CONTINENT	+	-

Generation Names page

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.
	<i>Level</i> 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.

Chapter 5

Defining Essbase Cube Outlines

This chapter discusses how to define cube outlines.

Defining Cube Outlines

This section discusses how to:

- Add cube dimensions to a cube outline.
- Add cube data queries to a cube outline.

Pages Used to Define Cube Outlines

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Cube Outline	CUB_OUTLINE	PeopleTools, Cube Builder, Outlines	Add dimensions to the current outline.
Add Cube Dimensions	CUB_OUTLINE_SEC	PeopleTools, Cube Builder, Outlines On the Cube Outline page, click the Add Cube Dimensions link.	Select dimensions to add to the current outline.
Current Dimension List	CUB_OUTDIM_SEC	PeopleTools, Cube Builder, Outlines On the Cube Outline page, click the Remove Cube Dimensions link.	Remove dimensions from the current outline.
Data Sources	CUB_OUTLINE2	PeopleTools, Cube Builder, Outlines On the Cube Outline page, select the Data Sources tab.	Add the cube data query or queries. You can supply one or more queries to fill the cube.

Page Name	Definition Name	Navigation	Usage
Find Query	CUB_QRYSEARCH_SEC	PeopleTools, Cube Builder, Outlines On the Cube Outline page, select the Data Sources tab. On the Data Sources page, click the Add Query link.	Search for available queries.
Dimension Mapping	CUB_OUTLINE2_SEC	PeopleTools, Cube Builder, Outlines On the Cube Outline page, select the Data Sources tab. On the Data Sources page, click the No Dimension Mapped link for the selected outline heading.	Map columns from data extraction queries to dimensions, select a column from data extraction as the measure, and unmap previously mapped columns.
Query Prompt Values	CUB_QRYPROMPT_SEC	PeopleTools, Cube Builder, Outlines On the Cube Outline page, select the Data Sources tab. On the Data Sources page, click the Query Parameters link for the selected outline heading.	View the current prompt values for a query and change them if necessary.
Member Formula	CUB_FORMULA_SEC	PeopleTools, Cube Builder, Outlines On the Cube Outline page, click the Formula link.	Enter formula information for an outline member.

Adding Cube Dimensions to a Cube Outline

Access the Cube Outline page by selecting PeopleTools, Cube Builder, Outlines.

Cube Outline

Data Sources

Outline

QE_CUBT_ALL_DIMENSION

Outline Type

BSO

Description

Contain all dimensions

Owner ID

☒ Duplicate Members Allowed

☐ Auto Configure Sparsity

☐ Run Default Calc

Comment

[Add Cube Dimensions](#)

[Remove Cube Dimensions](#)

Outline Structure

Left | Right

[-] QE_CUBT_ALL_DIMENSION

- [+] QE_CUBT_ACCOUNTS
- [+] QE_CUBT_AMERICA_REG
- [-] QE_CUBT_COUNTRY
 - [-] ALL_COUNTRY
 - [+] [GBR - (+)]
 - [+] [NAM - (+)]
 - [+] [SAM - (+)]
 - [+] [JPN - (+)]
 - [+] [ASIA - (+)]
- [+] QE_CUBT_TRANS_DATE

Save

Member Properties

Dynamic Source

Dimension

QE_CUBT_COUNTRY

BSO Storage Type

☐ Unique Members

Cube Outline page

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 section.

Outline Type	Displays the type of outline. Available types are: <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	Select the owner of the cube outline. The owner ID helps to determine the product, feature, or application team that last made changes to this outline.

Duplicate Member Allowed	Select to indicate that duplicate member names are allowed in the dimensions.
	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.
Auto Configure Sparsity	Select to enable Essbase to automatically set sparsity levels.
	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> .
	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.
Run Default Calc (run default calculation)	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.
	Deselect this option to create a custom calculation script.
Add Cube Dimensions	Click to access the Add Cube Dimensions page, where you can add dimensions to the current outline.
	Note. A dimension can be used only once in an outline.
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

Access the Add Cube Dimensions page (on the Cube Outline page, click the Add Cube Dimensions link).

Add Cube Dimensions

Dimension:

Search

Available Dimensions			Personalize	Find		First	1-6 of 6	Last
Select	Dimension	Type						
<input type="checkbox"/>	QE_CUBT_ALL_ACCOUNTS	Acc						
<input type="checkbox"/>	QE_CUBT_ALL_REGIONS	None						
<input type="checkbox"/>	QE_CUBT_API_ACCOUNT	Acc						
<input type="checkbox"/>	QE_CUBT_API_TEST	None						
<input type="checkbox"/>	QE_CUBT_EURO_PRODUCTS	None						
<input type="checkbox"/>	QE_CUBT_SELLER	None						

Add Cube Dimensions page

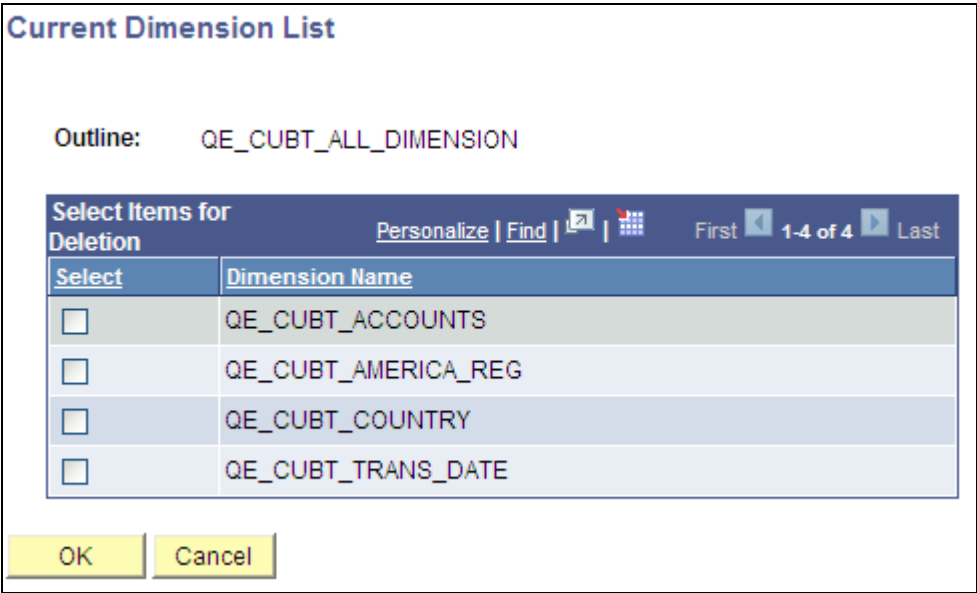
- Search

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

Access the Current Dimension List page (on the Cube Outline page, click the Remove Cube Dimensions link).



Current Dimension List page

- 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.

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

Cube OutlineData Sources

Outline

OUTLINE_CUBE

Outline Type

BSO

Description

Owner ID

☒ Duplicate Members Allowed

☐ Auto Configure Sparsity

☐ Run Default Calc

Comment

[Add Cube Dimensions](#)

[Remove Cube Dimensions](#)

Outline Structure

Left | Right

+ OUTLINE_CUBE

Save

Cube Outline page

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.

Add Cube Dimensions

Dimension:

Available Dimensions		
Select	Dimension	Type
<input type="checkbox"/>	QE_CUBT_ACCOUNTS	Acc
<input type="checkbox"/>	QE_CUBT_ALL_REGIONS	None
<input type="checkbox"/>	QE_CUBT_AMERICA_REG	None
<input type="checkbox"/>	QE_CUBT_API_ACCOUNT	Acc
<input type="checkbox"/>	QE_CUBT_API_TEST	None
<input type="checkbox"/>	QE_CUBT_COUNTRY	Country
<input type="checkbox"/>	QE_CUBT_EURO_PRODUCTS	None
<input type="checkbox"/>	QE_CUBT_SELLER	None
<input type="checkbox"/>	QE_CUBT_TRANS_DATE	Time
<input type="checkbox"/>	PRODUCT	Acc

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

Cube Outline

Data Sources

Outline ACCOUNT_CUBE
Outline Type BSO
Description
Owner ID
☒ Duplicate Members Allowed
☐ Auto Configure Sparsity
☐ Run Default Calc
Comment

[Add Cube Dimensions](#)
[Remove Cube Dimensions](#)

Outline Structure

Left | Right

ACCOUNT_CUBE

QE_CUBT_ALL_ACCOUNTS

ALL_ACCOUNTS

[E100 - (+)]

[E101 - (+)]

[E102 - (+)]

[A201 - (+)]

[E202 - (+)]

[E301 - (+)]

[E401 - (+)]

[E402 - (+)]

ACCOUNTS

Account Query

[QEEU - (+)](QEEU)]

[QEUS - (+)](QEUS)]

[QESH - (+)](QESH)]

Save

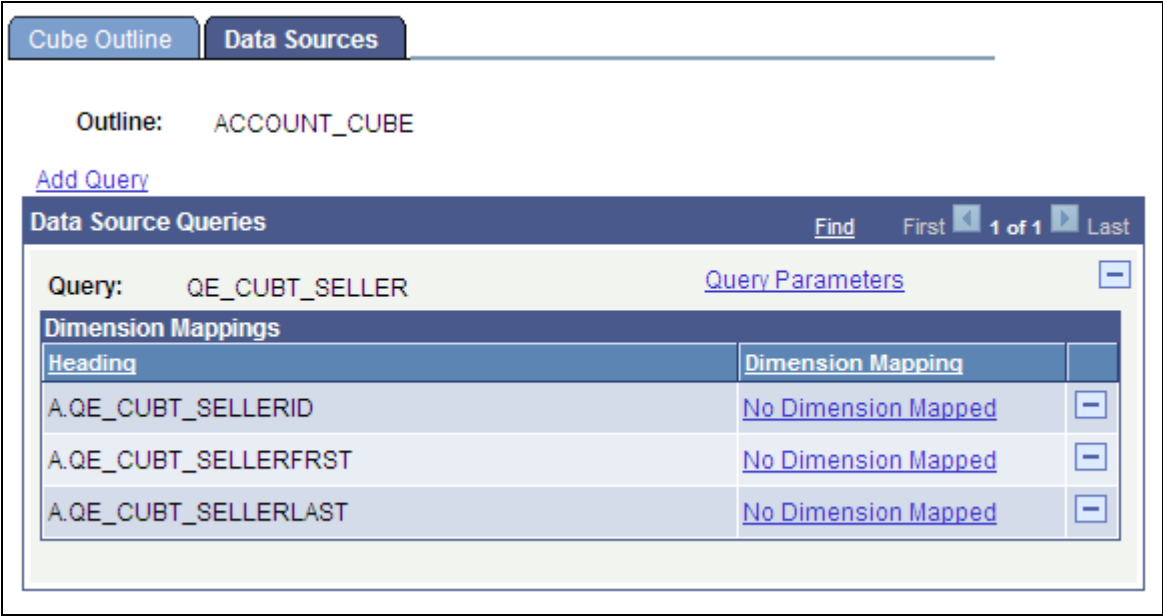
Cube Outline page displaying the QE_CUBT_ALL_ACCOUNTS and ACCOUNTS dimensions

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

Access the Data Sources page (on the Cube Outline page, select the Data Sources tab).



Data Sources page

- Add Query

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

Access the Find Query page (on the Data Sources page, click the Add Query link).

Find Query

Query

Select	Query Name
<input type="checkbox"/>	QE_CUBT_ACCOUNTS
<input type="checkbox"/>	QE_CUBT_COUNTRY
<input type="checkbox"/>	QE_CUBT_SELLER
<input type="checkbox"/>	QE_CUBT_SINGLE_PROMPT
<input type="checkbox"/>	QE_CUBT_TRANSACTIONS
<input type="checkbox"/>	QE_CUBT_TRANSDATES
<input type="checkbox"/>	QE_CUBT_TRANS_REGION

Find Query page

Available Queries	Displays a list of existing queries that are defined in the system after you click the Search button.
Note. Only public queries to which the current user has authorization appear in this section.	

Mapping to Dimensions

Access the Dimension Mapping page (on the Date Sources page, click the No Dimension Mapped link for the selected outline heading).

Dimension Mapping

Dimension:

Mapping Action

☒ Map to Dimension

Available Dimensions	
Select	Dimension
<input type="checkbox"/>	PRODUCT
<input type="checkbox"/>	ACCOUNTS
<input type="checkbox"/>	QE_CUBT_ACCOUNTS
<input type="checkbox"/>	QE_CUBT_ALL_ACCOUNTS

☐ Select as Measure

☐ Unmap

OK

Cancel

Dimension Mapping page

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

Access the Query Prompt Values page (on the Date Sources page, click the Query Parameters link).

Query Prompt Values

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

Prompt Field Values		
Field Name	Type	Value
SETID	CHAR	EUR

OK Cancel

Query Prompt Values page

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.

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

Cube Outline

Data Sources

Outline: QE_ACCOUNT_OUTLINE

[Add Query](#)

Data Source Queries

Find First 1 of 1 Last

Query: QE_CUBT_TRANSACTIONS

[Query Parameters](#)

[-]

Dimension Mappings		
Heading	Dimension Mapping	
A.BUSINESS_UNIT	No Dimension Mapped	[-]
A.QE_CUBT_TRANSID	No Dimension Mapped	[-]
A.QE_CUBT_ACCOUNT	No Dimension Mapped	[-]
A.QE_CUBT_REGIONID	No Dimension Mapped	[-]
A.QE_CUBT_SELLERID	No Dimension Mapped	[-]
A.QE_CUBT_PRODUCTID	No Dimension Mapped	[-]
A.QE_CUBT_TRANSDATE	No Dimension Mapped	[-]
A.QE_CUBT_TRANSAMNT	No Dimension Mapped	[-]

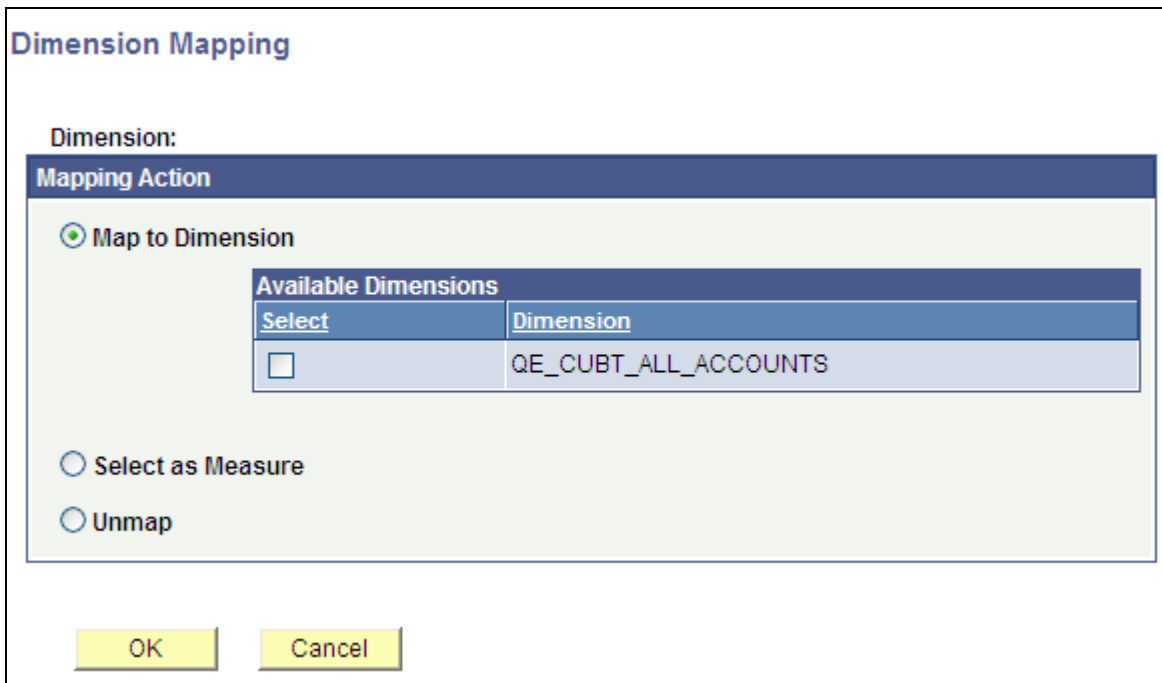
Data Sources page

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:



The image shows a 'Dimension Mapping' dialog box. At the top, it says 'Dimension:'. Below that is a section titled 'Mapping Action' with three radio button options: 'Map to Dimension' (which is selected), 'Select as Measure', and 'Unmap'. To the right of the 'Map to Dimension' option is a table titled 'Available Dimensions'. The table has two columns: 'Select' and 'Dimension'. There is one row in the table with a checkbox in the 'Select' column and the text 'QE_CUBT_ALL_ACCOUNTS' in the 'Dimension' column. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

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

Dimension Mapping page

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

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:

Cube Outline

Data Sources

Outline: QE_ACCOUNT_OUTLINE

[Add Query](#)

Data Source Queries

FindFirst1 of 1Last

Query: QE_CUBT_TRANSACTIONS

[Query Parameters](#)

Dimension Mappings

Heading	Dimension Mapping	
A.QE_CUBT_ACCOUNT	QE_CUBT_ALL_ACCOUNTS	<div></div>
A.QE_CUBT_REGIONID	No Dimension Mapped	<div></div>
A.QE_CUBT_SELLERID	No Dimension Mapped	<div></div>
A.QE_CUBT_PRODUCTID	Measure	<div></div>
A.QE_CUBT_TRANSDATE	No Dimension Mapped	<div></div>
A.QE_CUBT_TRANSMNT	No Dimension Mapped	<div></div>

Save

Data Sources page with column mapped to dimension

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.

Chapter 6

Creating Dynamic Dimension Templates

This chapter provides an overview of dynamic dimension templates and discusses how to create 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

Access the Dynamic Dimension Templates page (CUB_TEMPLATES) by selecting PeopleTools, Cube Builder, Dynamic Dimension Templates.

Dynamic Dimension Templates

Template ID

QE_DYNAMIC_TEST

Description

Test Dynamic Template

Comments

Object Owner ID

*Consolidation Property

+

☐ Expense Report

*Time Balance

None

☐ Never Share

*Skip Option

None

Updated By

VP1

Updated on

03/10/2009 3:39:47PM

Dynamic Dimension Templates page

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

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 [Chapter 4, "Creating Dimensions for Essbase Cubes," Defining an Essbase Dimension, page 28.](#)

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*

Chapter 7

Building Essbase Cubes

This chapter discusses how to:

- Build Essbase cubes.
- Integrate with Oracle Smart View.
- Use application programming interface (APT) and command-line specifications.

Building Essbase Cubes

This section discusses how to create Essbase cubes:

Creating Essbase Cubes

Access the Create Cubes page (CUB_RUNCNTL) by selecting PeopleTools, Cube Builder, Create Cube.

Create Cubes

Run Control ID: RUNCNTRL1

[Report Manager](#) [Process Monitor](#)

Run

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 07/12/2011 5:22:36PM

Create Cubes page

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

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

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.

Available options are:

- *Create or Re-create*: Create or recreate metadata from the beginning.
- *Incremental Replace All*: Update metadata incrementally and remove unspecified members.
- *Incremental Update*: Merge updates incrementally with existing metadata.
- *No Action*: Do not perform an action on the metadata.

Restructuring Options

Select an option to build the data (the cube contents).

Available options are:

- *Discard all data*: Discard all previously loaded or manually modified data.
- *Keep all data*: Do not modify the data.
- *Keep only input data*: Keep only input data or data blocks created by data loading.
- *Keep only level 0 data*: Delete upper level data incrementally.

Aggregations on Load

Three available options are:

- Replace

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.

- Addition

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.

- 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 [Chapter 2, "Designing Cube Metadata," Aggregations on Load Tool, page 15.](#)

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.

See Also

PeopleTools 8.52: PeopleSoft Process Scheduler, "Submitting and Scheduling Process Requests"

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. To view cube data using Smart View:

1. Build Essbase cube.

See [Chapter 7, "Building Essbase Cubes," Creating Essbase Cubes, page 77.](#)

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, as shown in this example:

Process Detail

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 Location Server Server PSNT Recurrence	<input type="radio"/> Hold Request <input type="radio"/> Queue Request <input type="radio"/> Cancel Request <input checked="" type="radio"/> Delete Request <input type="radio"/> Restart Request

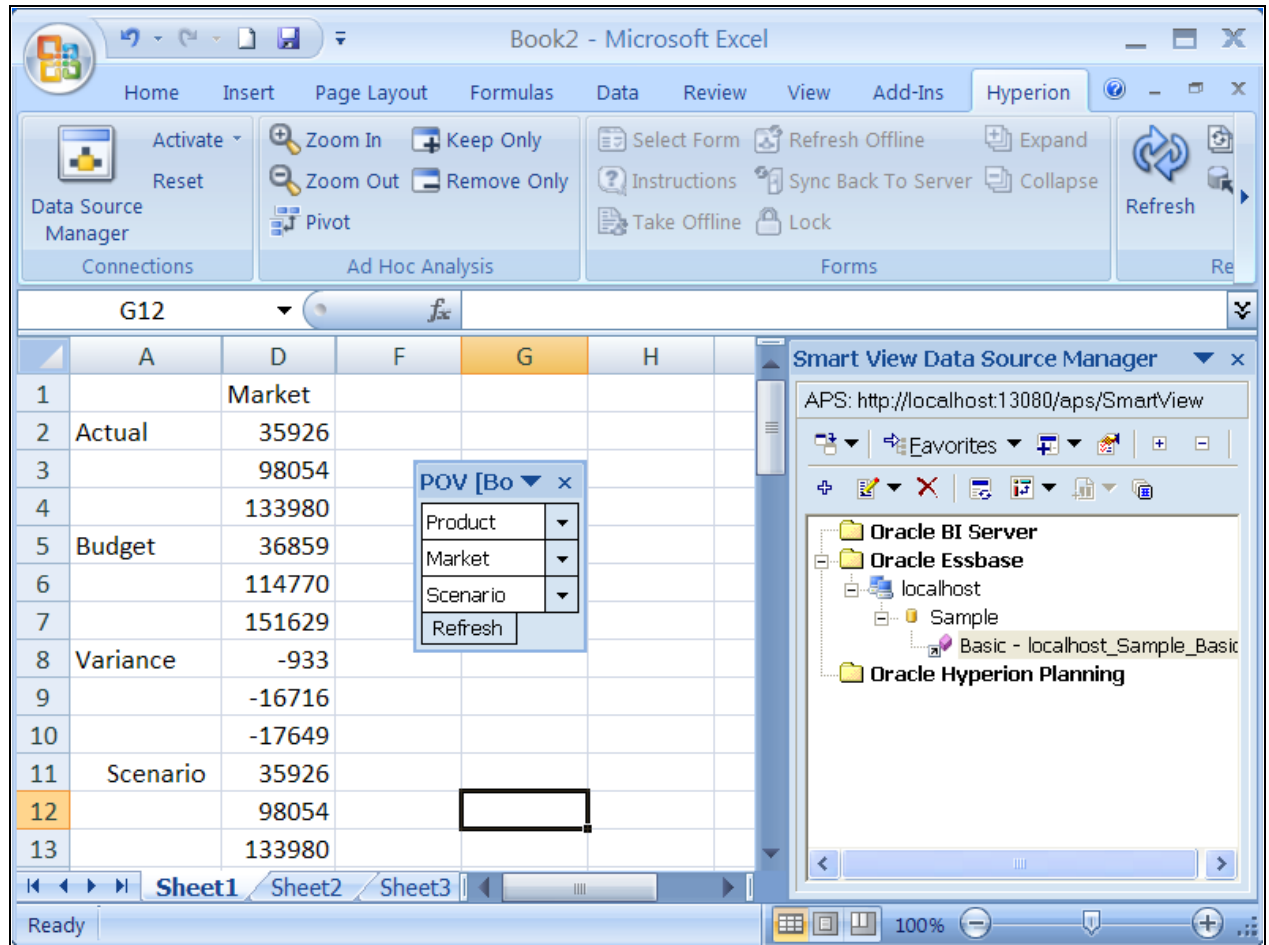
Date/Time	Actions
Request Created On 08/10/2011 3:30:38PM PDT	Parameters Transfer
Run Anytime After 08/10/2011 3:30:35PM PDT	Message Log
Began Process At 08/10/2011 3:31:07PM PDT	Batch Timings
Ended Process At 08/10/2011 3:32:07PM PDT	View Log/Trace
	Open Cube in Smart View

Example of the Process Detail page and the Open Cube in Smart View link

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, as shown in this example:



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.

This 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>-CPPTDMO</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.

Parameter	Description
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.
L, l	Logging level. For example, <i>L3I</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.

Example: the parameters list passed using this command line:

```
ps2essbase.exe -CX -CTORACLE -CDPT850RC1 -COPTDMO -CPPTDMO -I42 -OP "c:\tmp\ps" -REMPLOYMENT
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.

This table lists the log level and the data recorded in each log level:

Log Level	What is recorded
1	Basic logging
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, PeopleSoft 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

This chapter discusses how to secure Cube Builder cubes.

Securing Cube Builder Cubes

This section discusses how to:

- Control user operations.
- Set user actions.
- View content references.

Pages Used to Secure Cube Builder Cubes

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Component Permissions	ACL_COMPONENT2	PeopleTools, Security, Permissions & Roles, Permission Lists Create a new permission list or search for an existing one. Select the Pages tab and then click the appropriate Edit Components link.	Control the operations that a user is allowed to perform within Essbase Cube Builder.
Page Permissions	ACL_PAGES2	PeopleTools, Security, Permissions & Roles, Permission Lists Create a new permission list or search for an existing one. Select the Pages tab and click the Edit Components link. Click the Edit Pages link.	Set the actions that a user can complete on a page.

Page Name	Definition Name	Navigation	Usage
Content References	PLIST_CREF	PeopleTools, Security, Permissions & Roles, Permission Lists Create a new permission list or search for an existing one. Select the Pages tab and click the Edit Components link. Click the View link.	View all the content references in the database that point to the component.

Controlling User Operations

Use these steps to access the Component Permissions page where you can control user operations:

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.

Component Permissions

Cube Manager

Components					Personalize	Find	First	1-12 of 12	Last	Select All
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						

OK
 Cancel
 Deselect All

Component Permissions page

Authorized?

Displays the components to which the current permission list has access.

Component Name	<p>Displays the name of components beneath the authorized menu. PeopleSoft 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 Chapter 8, "Securing Cubes," Setting User Actions, page 90.</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> <p>See Chapter 8, "Securing Cubes," Viewing Content References, page 91.</p>
Select All	<p>Click to grant access to all or selected items.</p> <p>To grant access to <i>most</i> menu items, click this button and then clear the items to which you do not want to grant access.</p>
Deselect All	Click to deselect items that were previously selected.

Setting User Actions

Use these steps to access the Page Permissions page (ACL_PAGES2) where you can set user actions:

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.

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

Page Permissions page

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 *Add*, *Update/Display*, and *Correction*.

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 these steps to access the Content References page (PLIST_CREF) where you can view content references:

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.

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.

Personalize | Find |   First 1-6 of 6 Last

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}

Content References page

The Content References page displays 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.

See Also

PeopleTools 8.52: Security Administration, "Setting Up Permission Lists"

Chapter 9

Maintaining Essbase Cubes

This chapter discusses how to maintain Essbase cubes using the Essbase Utilities component (CUB_DIMDELETE).

Maintaining Essbase Cubes

This section discusses how to:

- Delete dimensions.
- Delete outlines.
- Delete run control IDs.
- View dynamic objects.
- View outline structures.
- Expand dimensions.

Pages Used to Maintain Essbase Cubes

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Dimension Delete	CUB_DIMDELETE	PeopleTools, Cube Builder, Utilities	Delete any dimension created in the system as long as the dimension is not referenced in any outline.
Outline Delete	CUB_OUTDELETE	PeopleTools, Cube Builder, Utilities On the Dimension Delete page, select the Outline Delete tab.	Delete any outline created in the system as long as the outline is not referenced in any cube creation run control.
Run Control Delete	CUB_RUNDELETE	PeopleTools, Cube Builder, Utilities On the Dimension Delete page, select the Run Control Delete tab.	Delete any cube creation run control.

Page Name	Definition Name	Navigation	Usage
View Dynamic Objects	CUB_DIMDYNVIEW	PeopleTools, Cube Builder, Utilities On the Dimension Delete page, select the View Dynamic Objects tab.	View all dynamic objects (queries and trees) that are used in the definition for the selected dimension. View any runtime parameters that are used when you run queries. View the queries and trees that are part of the dimensional structure if you are having problems opening the dimension definition. Diagnose permission problems with dynamic objects.
View Outline Structures	CUB_OUTSTRUCT	PeopleTools, Cube Builder, Utilities On the Dimension Delete page, select the View Outline Structures tab.	View all the dimensions that are part of the outline for the selected outline.
Outline Expansion	CUB_DYNEXPAND_SEC	PeopleTools, Cube Builder, Utilities On the Dimension Delete page, select the Outline Expansion tab.	Perform the full dimension expansion for every dimension that is a part of the selected outline. Run all dynamic queries and trees in each dimension, and display the total number of rows processed for each dynamic object.

Deleting Dimensions

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

Dimension Delete
Outline Delete
Run Control Delete
View Dynamic Objects

Dimension:

Dimensions		
Select	Dimension	Result
<input type="checkbox"/>	QE_CUBT_ACCOUNTS	
<input type="checkbox"/>	QE_CUBT_ALL_ACCOUNTS	
<input type="checkbox"/>	QE_CUBT_ALL_REGIONS	
<input type="checkbox"/>	QE_CUBT_AMERICA_REG	
<input type="checkbox"/>	QE_CUBT_API_ACCOUNT	
<input type="checkbox"/>	QE_CUBT_API_TEST	
<input type="checkbox"/>	QE_CUBT_COUNTRY	
<input type="checkbox"/>	QE_CUBT_EURO_PRODUCTS	
<input type="checkbox"/>	QE_CUBT_SELLER	
<input type="checkbox"/>	QE_CUBT_TRANS_DATE	

Dimension Delete page

Use the Dimension Delete page to delete any created dimension in the system as long as the dimension is not referenced in any outline.

Deleting Outlines

Access the Outline Delete page (on the Dimension Delete page, select the Outline Delete tab).

Dimension Delete
Outline Delete
Run Control Delete
View Dynamic Objects

Outline:

Outlines		
Select	Outline	Result
<input type="checkbox"/>	QE_ACCOUNT_OUTLINE	
<input type="checkbox"/>	QE_CUBT_ALL_DIMENSION	
<input type="checkbox"/>	QE_CUBT_API_TEST	
<input type="checkbox"/>	QE_CUBT_COUNTRY	
<input type="checkbox"/>	QE_CUBT_STANDARD	
<input type="checkbox"/>	QE_CUBT_TRANS_DATE	
<input type="checkbox"/>	QE_SMOKE_CUBE	

Outline Delete page

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

Deleting Run Control IDs

Access the Run Control Delete page (on the Dimension Delete page, select the Run Control Delete tab).

Dimension Delete Outline Delete **Run Control Delete** View Dynamic Objects

Run Control ID:

Search

Run Controls		Personalize Find View All	First 1 of 1 Last
Select	Run Control ID	Result	
<input type="checkbox"/>	RUNCNTRL1		

Delete

Run Control Delete page

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

Viewing Dynamic Objects

Access the View Dynamic Objects page (on the Dimension Delete page, select the View Dynamic Objects tab).

Dimension Delete Outline Delete Run Control Delete **View Dynamic Objects**

Dimension:

Search

Dimensions					Personalize Find View All	First 1-7 of 7 Last
Dimension	Sequence	Query	Tree			
QE_CUBT_ALL_ACCOUNTS	1	QE_CUBT_ACCOUNTS				Details
QE_CUBT_ALL_REGIONS	1		QE_CUBT_REGION			Details
QE_CUBT_API_ACCOUNT	1	QE_CUBT_ACCOUNTS				Details
QE_CUBT_API_TEST	1		QE_CUBT_REGION			Details
QE_CUBT_COUNTRY	1	QE_CUBT_COUNTRY				Details
QE_CUBT_EURO_PRODUCTS	1		QE_CUBT_PRODUCTS			Details
QE_CUBT_TRANS_DATE	1	QE_CUBT_TRANSDATES				Details

View Dynamic Objects page

Use the View Dynamic Objects page 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.

Viewing Outline Structures

Access the View Outline Structures page (on the Dimension Delete page, select the View Outline Structures tab).

Run Control Delete
View Dynamic Objects
View Outline Structures
Outline Expansion

Outline:

Search

Outlines and corresponding dimensions	
Outline	Dimension
QE_ACCOUNT_OUTLINE	QE_CUBT_ALL_ACCOUNTS
QE_CUBT_ALL_DIMENSION	QE_CUBT_ACCOUNTS
QE_CUBT_ALL_DIMENSION	QE_CUBT_AMERICA_REG
QE_CUBT_ALL_DIMENSION	QE_CUBT_COUNTRY
QE_CUBT_ALL_DIMENSION	QE_CUBT_TRANS_DATE
QE_CUBT_API_TEST	QE_CUBT_API_ACCOUNT
QE_CUBT_API_TEST	QE_CUBT_API_TEST
QE_CUBT_COUNTRY	QE_CUBT_COUNTRY
QE_CUBT_STANDARD	QE_CUBT_ALL_REGIONS
QE_CUBT_STANDARD	QE_CUBT_EURO_PRODUCTS
QE_CUBT_STANDARD	QE_CUBT_SELLER
QE_CUBT_TRANS_DATE	QE_CUBT_TRANS_DATE
QE_SMOKE_CUBE	QE_CUBT_ACCOUNTS
QE_SMOKE_CUBE	QE_CUBT_EURO_PRODUCTS

View Outline Structures page

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

Expanding Dimensions

Access the Outline Expansion page (on the Dimension Delete page, select the Outline Expansion tab).

Run Control DeleteView Dynamic ObjectsView Outline StructuresOutline Expansion

Outline: QE_ACCOUNT_OUTLINE

Expand

Result: Return code from expansion - 0.

Expansion Results

PersonalizeFindView AllFirst1-2 of 2Last

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

Notify

Outline Expansion page

Use the Outline Expansion page 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.

Appendix A

Viewing Query Results That Have Large Result Sets

This chapter discusses:

- Query chunking and the chunking class.
- Dynamic query expansion.

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();
```

This table lists the error codes for the chunker:

Code	Error
-1	Missing query name.
-2	Missing field heading.
-3	Error opening the query. Check erroropencode 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.

These tables include the current codes:

Code	Error
-21	Error opening a query.
-22	Error retrieving presaved prompt values for a query.

Code	Error
-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.
-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

This chapter provides an overview of Essbase Cube Builder demo data and discusses how to:

- Apply project CUBE_DEMO_PROJ.
- Grant permissions to all menu CUBE_TEST components.
- Import PS_CUBEBuilder.DAT file using Data Mover script PS_CUBEBuilder.DMS.
- Grant query security.
- Add data to tables.
- Use sample dimensions and outlines.

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.cat 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 PSCUBTTRANSD_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.

Index

A

- Add Cube Dimensions page 61, 63
- adding child member 34
- adding cube dimension to cube outline 58, 62
- Adding Data to Tables 107
- adding generation names 55
- adding peer member 34
- adding query prompts values 69
- Alias 46
- API
 - using 82
- attribute queries
 - Essbase attributes 10

B

- build cubes 77
- building cubes 3
- building Essbase cubes 77

C

- cells 6
- Child member
 - adding 34
- Child Member Add page 34
- Chunking
 - class 101
- Chunking Class 101
- column
 - selecting 45
- command-line specifications
 - using 82
- Component Permissions page 87, 88
- Connections page 21
- consolidation 8
- content references, viewing 91
- Content References page 88, 91
- Create Cubes page 77
- creating dimensions 3
- Creating Dynamic Dimension Templates 73
- creating outline 3
- creating template
 - dynamic dimension template 73
- CUBE_DEMO_PROJ 108
 - project 105
- CUBE_TEST
 - granting permissions 106
- Cube Builder *See* PeopleSoft Cube Builder
 - implementing: 2
- cube builder connection 21
- Cube Builder process type 11
- cube data queries
 - adding to cube outline 65, 69
- cube dimension
 - adding to cube outline 58, 62

- deleting 61
 - inserting 61
- cube dimension builder
 - setting up 25
- Cube Dimension Builder Setup page 25
- cube dimensions
 - designing using tree metadata 11
 - overview 5
- cube members 6
- cube outline
 - adding cube dimension to cube outline 58, 62
- Cube outline
 - adding cube data queries to cube outline 65, 69
- Cube Outline page 58, 64
- cubes 5
 - building 3
 - consolidating 8
 - hierarchies 7
 - securing 87
- Current Dimension List page 61

D

- data
 - adding to table 107
- data mover script 106
- Data Source page 65
- data source queries 13
- data sources
 - using multiple queries as 13
- data storage (Essbase attribute) 10, 33
- defining time dimensions 52
- deleting cube dimension 61
- deleting dimensions 94
- deleting outlines 95
- deleting run control ID 96
- demo data
 - Essbase cube builder demo data 105
- dimension
 - insert a tree structure into dimension 38, 41
- Dimension
 - Defining 28
 - inserting a query into dimension 44, 48
- Dimension Delete page 94
- dimension expansion
 - performing 98
- Dimension Mapping page 68
- dimension members 6
- Dimension page 28, 35, 41
- dimension queries 13
 - using parent/child relationship 13
- dimensions 5, 6
 - children within 8
 - creating 3
 - deleting 94
 - mapping 68
 - multiple 6
 - parents within 8
- Dimensions of Essbase cubes

- overview 25
- Duplicate checks 104
- Dynamic dimension templates
 - overview 73
- Dynamic Dimension Templates page 73
- Dynamic objects
 - viewing 97
- Dynamic Query Expansion 103

E

- effective-dated trees 11
- error code 103
- Essbase cube builder connection
 - defining 21
- Essbase cubes
 - building 77
- Essbase Dimension
 - Defining 28
- Essbase dimensions
 - defining 35
- Essbase Utilities 93
- expense item (Essbase attribute) 10

F

- fact 6
- fact data 6
- Find Query page 44
- Find Tree page 39, 41
- Fine Query page 67

G

- generation names
 - adding 55
- Generation Names page 55
- globalized formats 9
- granting query security 107

H

- hierarchies 7

I

- Inserting a query into dimension 44, 48
- inserting cube dimension 61

L

- Leaf SQL Errors 104
- link
 - open cube in smart view 81
- logging 85

- log level 85

M

- mapping dimensions 68
- measures
 - overview 6
- Member formula
 - adding 35
- Member Formula page 35
- members 6
 - case sensitivity and names 12
- Message catalog
 - error code 103
- metadata 9
 - cube 10
 - designing 5
 - dimensional 10
 - member property 10
- multilevel dimensions 7
- multiple dimensions 6

O

- Open Cube in Smart View 81
- Oracle Essbase
 - attributes 10
 - consolidation 10, 33, 74
 - rollups in 12
- Oracle Smart View 80
- outline
 - creating 3
- Outline
 - Cube Outline page 62
- Outline Delete page 95
- Outline Expansion page 98
- outlines
 - deleting 95
- Outline structures
 - viewing 98

P

- Page Permissions page 87, 90
- Parameter 83
- peer member
 - adding 34
- Peer Member Add page 34
- PeopleSoft Application Designer, available
 - actions 91
- PeopleSoft Cube Builder vii, 1
- PeopleSoft Process Scheduler
 - integrating with 11
- PeopleSoft queries
 - 9, *See Also* dimension queries, data source queries
 - creating 12
 - defining cubes with 9
 - overview types of 12
 - populating OLAP cubes with data using 9
- PeopleSoft Query 9
- PeopleSoft Tree Manager 8, 9

- PeopleSoft trees 9
 - copying 12
 - effective-dated 11
 - leaves 9
 - modifying 12
 - nodes 9
 - subsets of 12
 - using 11
- performing dimension expansion 98
- Process Scheduler
 - See* PeopleSoft Process Scheduler
- project
 - CUBE_DEMO_PROJ 108
- Project
 - CUBE_DEMO_PROJ 105
- PS_CUBEBUILDER.DAT
 - importing 106
- PS_CUBEBUILDER.DMS 106
- PS2Essbase 82

Q

- QE_CUBT_ACCOUNTS 108
- QE_CUBT_ALL_REGIONS 108
- QE_CUBT_EUROPEAN_PRODUCTS 108
- QE_CUBT_TRANSACTION_DATES 108
- queries *See* PeopleSoft queries
- query
 - searching 67
 - selecting 44
- Query Chunking 101
- Query Chunking and the Chunking Class 101
- query prompts values
 - adding 69
- Query Prompt Values page 69
- query security
 - granting 107

R

- Rollup Level 46
- rollups
 - in Essbase 12
- Run Control Delete page 96
- run control ID
 - deleting 96

S

- sample dimensions
 - using 108
- sample outlines
 - using 108
- security 87
 - granting query security 107
 - setting user actions 90
 - setting user options 88
- Select Column page 45
- selecting a column 45
- selecting a query 44
- selecting a tree 39
- selecting the top node 40

- SELECT statements 9
- Select Top Node page 40, 42
- Smart View
 - displays top member of dimensions 82
 - Integrating with Oracle Smart View 80
- SQL statements 9

T

- table
 - adding data 107
- template
 - dynamic dimension template 73
- time balance (Essbase attribute) 10, 33, 74
- time dimensions
 - defining 52
- Time Dimension Wizard page 52
- top node
 - selecting 40
- tree
 - selecting 39
- trees *See* PeopleSoft trees
- tree structure
 - inserting 38, 41

U

- unary operator (Essbase attribute) 10
- Using API 82
- Using Command-line Specifications 82

V

- View Dynamic Objects page 97
- Viewing Dynamic Objects 97
- Viewing Outline Structures 98
- View Outline Structures page 98

