



PeopleTools 8.12 PeopleSoft Cube Manager PeopleBook

SKU MTCMr8SP1B 1200

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Copyright © 2001 by PeopleSoft, Inc. All rights reserved.

Printed in the United States of America.

All material contained in this documentation is proprietary and confidential to PeopleSoft, Inc. and is protected by copyright laws. No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of PeopleSoft, Inc.

This documentation is subject to change without notice, and PeopleSoft, Inc. does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to PeopleSoft, Inc. in writing.

The copyrighted software that accompanies this documentation is licensed for use only in strict accordance with the applicable license agreement which should be read carefully as it governs the terms of use of the software and this documentation, including the disclosure thereof.

PeopleSoft, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, Vantive, and Vantive Enterprise are registered trademarks, and *PeopleTalk* and "People power the internet." are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners.

Contents

About This PeopleBook

Before You Begin	vii
Related Documentation	vii
Documentation on the Internet	viii
Documentation on CD-ROM	viii
Hardcopy Documentation	viii
Typographical Conventions and Visual Cues	ix
Comments and Suggestions	x

Chapter 1

Introduction to Cube Manager

Overview of Online Processing	1-1
Cubes	1-3
Dimensions and Members	1-3
Measures and Cells	1-3
Multiple Dimensions	1-4
Rollups	1-4
Understanding Cube Manager Architecture	1-7
Designing with Cube Manager	1-7
PeopleSoft Metadata	1-8
PeopleSoft Trees	1-8
PeopleSoft Queries	1-8
Process Scheduler Integration	1-9
Supported OLAP Tools	1-9
PowerPlay	1-10
Using Cube Manager in PeopleTools 7.x with PowerPlay 5.2	1-10
Using Cube Manager in PeopleTools 7.x with PowerPlay 6.6 Personal Server and Transformer	1-11
Using Cube Manager 8 with PowerPlay 6.6 Enterprise Server Edition	1-11
Essbase	1-13
PS/ROLAP	1-13
Designing and Building OLAP Cubes	1-13
Designing the metadata.	1-14

Using Cube Manager.....	1-14
-------------------------	------

Chapter 2

Designing Cube Metadata

Trees	2-1
Queries.....	2-2
Dimension Queries.....	2-3
Narrow Query Definition	2-4
Wide Query Definition.....	2-7
Attribute Queries	2-8
General Attributes	2-9
Essbase Attributes	2-9
PowerPlay Attributes	2-11
PS/ROLAP Attributes	2-11
Level Attribute Queries.....	2-12
Data Source Queries.....	2-13

Chapter 3

Designing Cubes

Overview of Cube Components.....	3-1
Designing a Cube.....	3-3
Dimensions - Dimension Page	3-3
Dimension Options Page.....	3-8
Rollup Options Page	3-9
Dimensions - Rollup Inputs Page.....	3-11
Dim Input Field Page	3-13
Designing Cube Definitions	3-14
Cube Definitions - Cube Definition Page.....	3-15
Cube Definitions - Cube Inputs page	3-17
Cube Input Field.....	3-18
Designing Attribute Names	3-20
Attribute Names - Attribute Name page.....	3-20

Chapter 4

Building Cubes

Building Cube Instances	4-1
Cube Instances - Cube Inst Def page	4-1
Cube Instances - Cube Inst Def page : Filter tab.....	4-3
Cube Filter Criteria	4-4
Platform Options	4-6

Hyperion Essbase Options	4-6
Cognos PowerPlay Options.....	4-12
Generic Star Schema Options	4-16
PS/ROLAP Options	4-16
Cube Instances - Aggregates page	4-18
Building Cubes	4-18
Build Cubes - Build Cubes page	4-19
Build Cubes - Bind Variables page	4-22
Example of Bind Variables Used in PeopleSoft General Ledger.....	4-24
Setting Up Process Scheduler	4-26
Process Definition page on Process Definitions page	4-27
Process Scheduler Request Page	4-28

Chapter 5

Securing Cubes

Setting Cube Manager Security	5-1
Component Permissions for Cube Manager.....	5-1
Page Permissions for Cube Manager	5-2
Troubleshooting Cube Manager	5-4
Troubleshooting Examples.....	5-5

Chapter 6

Glossary

Index

ABOUT THIS PEOPLEBOOK

This book discusses Cube Manager, a set of PeopleTools pages and processes that you use to create and maintain analytic data stores, also called OLAP (Online Analytical Processing) cubes. Its chapters explain the basic concepts of OLAP and the tools that are supported, as well as how to use PeopleTools to design analytic components, build, and troubleshoot cubes.

The following topics are discussed in the rest of the chapters of the Cube Manager PeopleBook.

Introduction to Cube Manager explains the basic concepts of OLAP, and provides an overview of how Cube Manager works.

Designing Cube Metadata discusses how Cube Manager uses two PeopleTools data constructs—trees and queries—to create cube dimensions.

Designing Cubes explains how to create the various components of cubes: Dimensions and Cube Definitions.

Designing Cubes covers how to create Instance Definitions. It also explains the final process of building a cube for viewing with your OLAP tool.

Securing Cubes discusses data warehouse tuning issues, how to apply security to a cube, and how to track down the causes of common Cube Manager errors.

Before You Begin

To benefit fully from the information covered in this book, you need to have a basic understanding of how to use PeopleSoft applications. We recommend that you complete at least one PeopleSoft introductory training course.

You should be familiar with navigating around the system and adding, updating, and deleting information using PeopleSoft windows, menus, and pages. You should also be comfortable using the World Wide Web and the Microsoft® Windows or Windows NT graphical user interface.

In addition to familiarity with PeopleTools, you must have specific knowledge of Tree Manager, Query, Process Scheduler, and to some extent, PS/nVision. We also assume a basic familiarity with OLAP concepts and a good working knowledge of at least one of the supported third-party tools: Cognos PowerPlay, Hyperion Essbase, or CA Strategy ROLAP engine. (MS OLAP Services is not supported for PT 8.1.)

Related Documentation

To add to your knowledge of PeopleSoft applications and tools, you might want to refer to the documentation of the specific PeopleSoft applications that your company uses. You can access additional documentation for this release from PeopleSoft Customer Connection

(www.peoplesoft.com). We post updates and other items on Customer Connection, as well. In addition, documentation for this release is available on CD-ROM and in print.



Important! Before upgrading, it is *imperative* that you check PeopleSoft Customer Connection for updates to the upgrade instructions. We continually post updates as we refine the upgrade process.

Documentation on the Internet

You can order printed, bound versions of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM. You can order additional copies of the PeopleBooks CDs through the Documentation section of the PeopleSoft Customer Connection Web site: <http://www.peoplesoft.com/>.

You can also find updates to the documentation for this and previous releases on Customer Connection. Through the Documentation section of Customer Connection, you can download files to add to your PeopleBook library. There is a variety of useful and timely materials, including updates to the full PeopleSoft documentation delivered on your PeopleBooks CD.

Documentation on CD-ROM

Complete documentation for this PeopleTools release is provided in HTML format on the PeopleTools PeopleBooks CD-ROM. The documentation for the PeopleSoft applications you have purchased appears on a separate PeopleBooks CD for the product line.

Hardcopy Documentation

To order printed, bound volumes of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM, visit the PeopleSoft Press Web site from the Documentation section of PeopleSoft Customer Connection. The PeopleSoft Press Web site is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), our book print vendor.

We make printed documentation for each major release available shortly after the software is first shipped. Customers and partners can order printed PeopleSoft documentation using any of the following methods:

Internet From the main PeopleSoft Internet site, go to the Documentation section of Customer Connection. You can find order information under the Ordering PeopleBooks topic. Use a Customer Connection ID, credit card, or purchase order to place your order.

PeopleSoft Internet site: <http://www.peoplesoft.com/>.

Telephone Contact Consolidated Publishing Incorporated (CPI) at 800 888 3559.

Email Email CPI at callcenter@conpub.com.

Typographical Conventions and Visual Cues

To help you locate and interpret information, we use a number of standard conventions in our online documentation.

Please take a moment to review the following typographical cues:

`monospace
font` Indicates PeopleCode.

Bold Indicates field names and other page elements, such as buttons and group box labels, when these elements are documented below the page on which they appear. When we refer to these elements elsewhere in the documentation, we set them in Normal style (not in bold).

Italics Indicates a PeopleSoft or other book-length publication title. We also use italics for *emphasis* and to indicate specific field values. When we cite a field value under the page on which it appears, we use this style: ***field value***.

We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number *0*, not the letter *O*.

KEY+KEY Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W.

Jump links Indicates a jump (also called a link, hyperlink, or hypertext link). Click a jump to move to the jump destination or referenced section.

Cross-
references The phrase For more information indicates where you can find additional documentation on the topic at hand. We include the navigational path to the referenced topic, separated by colons (:). Capitalized titles in *italics* indicate the title of a PeopleBook; capitalized titles in normal font refer to sections and specific topics within the PeopleBook. Cross-references typically begin with a jump link. Here's an example:

For more information, see [Documentation on CD-ROM](#) in *About These PeopleBooks*: Related Documentation.

- Topic list Contains jump links to all the topics in the section. Note that these correspond to the heading levels in the Contents window.



Name of
Page

Opens a pop-up window that contains the named page. Click the icon to display the image. Some screen shots may also appear directly in the text.



Text in this bar indicates information to which you should pay particular attention as you work with your PeopleSoft system. If the note is preceded by **Important!**, the note is crucial and includes information that concerns what you need to do for the system to function properly.



Text in this bar indicates For more information cross-references to related or additional information.



Text within this bar indicates a crucial configuration consideration. Pay very close attention to these warning messages.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like changed about our documentation, PeopleBooks, and other PeopleSoft reference and training materials. Please send your suggestions to:

PeopleTools Product Documentation Manager
PeopleSoft, Inc.
4460 Hacienda Drive
Pleasanton, CA 94588

Or send comments by email to the authors of the PeopleSoft documentation at:

<mailto:DOC@PEOPLESOFT.COM>

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions. We are always improving our product communications for you.

CHAPTER 1

Introduction to Cube Manager

This chapter discusses the basic concepts of both OLAP (Online Analytical Processing) and Cube Manager, a PeopleTool that you use to build OLAP databases, or cubes. The data in these cubes can then be viewed, analyzed, and modified using the following products:

- Hyperion Essbase
- Cognos PowerPlay
- CA Strategy PS/ROLAP



Some Cube Manager documentation and pages refer to the CA Strategy product as Sterling Eureka:StrategyPS and Sterling PS/ROLAP. Please be aware that both names refer to the third-party product: CA Strategy.

The information presented here is not a substitute for your PowerPlay, Essbase, or CA Strategy PS/ROLAP documentation, but will help you integrate your PeopleSoft data with one or more of these tools. Once your PeopleSoft data is integrated with one of these products, users need to be familiar with the product to effectively analyze and interact with the data.

Overview of Online Processing

Most business software users are familiar with OLTP (Online Transaction Processing) applications. These applications are used to create and maintain information about business operations. The transactions stored by applications of this type are the heart of any business software. At the database level, OLTP applications are designed to allow speedy creation of data and to reduce redundant information. Data structures of this design are not well-suited to analysis.

The nature of OLTP databases poses a problem: how to analyze data in a database that is not designed for analysis. One solution is to use a product such as PS/nVision. Such products perform analysis on selected characteristics of the database. However, using a powerful tool, such as PS/nVision, on top of an OLTP database takes time.

In contrast, OLAP applications are designed specifically for data analysis. The source of information for analysis is an OLTP database. To make the OLTP data available to analytical applications, data is extracted and transformed into a format that can be analyzed more easily. The resulting OLAP database can be stored in several different formats, depending on the tools used to access the data.

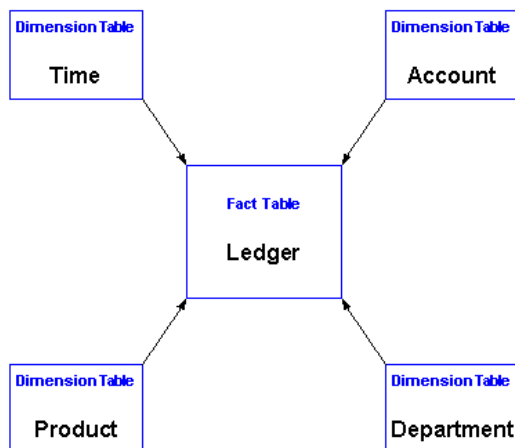
MOLAP (multidimensional OLAP) is a format in which all of the data is stored in a hierarchical format. This type of database is used mainly for small-to-medium data marts. MOLAP databases like PowerPlay and Essbase are very fast at summarizing and navigating through data. The only drawback of MOLAP systems is that as dimensional information, sizes, or numbers increase, the storage mechanism becomes less efficient.

ROLAP (Relational OLAP) is a format in which the analytical data is stored in relational tables. The main benefit of ROLAP data storage is its ability to store vast amounts of data. But ROLAP data storage is not as efficient in accessing aggregate information at higher levels of the hierarchy.

The structures of the data schema can be one of two types: snowflake or star schema. Snowflake schemas are designed to keep very little redundant data, whereas star schemas encourage duplicate data. This release of Cube Manager only supports the star schema, so here we concentrate on that format. Sterling PS/ROLAP can use data stored in ROLAP structures.

In a star schema, each dimension is represented in a single table. The fact data, data that is to be analyzed, is stored in a separate table. There is one column in the fact table that represents the dimension from which the data was created. The following diagram illustrates a typical star schema:

The following diagram illustrates a typical star schema:



Star Schema structure

HOLAP (Hybrid OLAP) is the latest type of analytical structure. As the name implies, the data is stored in a hybrid format. The base fact data is stored in a set of relational tables, while the summary data is stored in a multidimensional format. This strategy is very powerful, but can pose problems when combining data from the relational tables with the multidimensional summary data.

Generally speaking, multidimensional analysis is not a technology. Rather, it's a type of analysis that enables you to look easily at data from many different perspectives. These tools enable users to explore, interact with, and "slice and dice" complex data, guiding them to the multidimensional information they need, but can't easily discover with conventional reporting tools. This is all done in a way and at a speed not possible with traditional RDBMS (relational database management system) applications.

Cubes

The key concept of OLAP is that of a cube. In this document, we use the term cube when referring to any analytic data store, whether it is a MOLAP cube or a ROLAP star schema. An OLAP cube is a collection of related data—a database, really—having multiple dimensions. Cube dimensions are the rough 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 OLTP system.

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

Let's say you want to analyze your company's sales (in units). You could examine the total units sold in a particular year, but that wouldn't help you understand much about your business. Instead, you might want to see unit sales broken down by time and by products. It should be fairly easy to envision the matrix that you use to analyze this data. It might look something like this:

Product	1994	1995	1996
Widgets	3000	6500	8200
Gadgets	1200	1450	3000
Doohickeys	2500	3400	2000
Whatzits	500	670	1300

Illustration of a Cube With Two Dimensions

Dimensions and Members

In OLAP terminology, the preceding matrix 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 our example, each dimension is subdivided into categories, called cube members, that represent individual years and products. So, in the Time dimension, the members are 1994, 1995, and 1996. In the Product dimension, the members are Widgets, Gadgets, Doohickeys, and Whatzits.

Measures and Cells

In the preceding matrix, what values are we most interested in? Not years or products. The purpose of creating the matrix is to find the number of units sold. Units sold comprise the data element 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 matrix where members intersect with other members represent individual measure/fact values. These intersections are called cells. The shaded cell in the table above represents the number of Widgets sold in 1995: 6500 units.

Multiple Dimensions

The two-dimensional cube pictured above is rather basic, for reporting purposes. For example, it gives us no idea where any of the units were sold. We can provide this information by adding another dimension, Location, to the model:

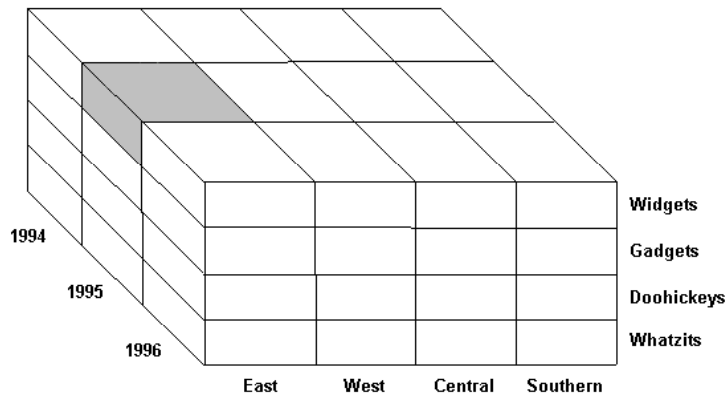


Illustration of a Cube With Three Dimensions

This OLAP cube represents units sold dimensioned by time, product, and location. (The location members are East, West, Central, and Southern.) The shaded cell represents the number of Widgets sold in the East region in 1995. You could find the number of units sold for any other product in any other region at any other time, simply by finding the cell at the intersection point of three members, one from each dimension.

Now, suppose you also want to factor customer accounts into the analysis. While depicting four dimensions graphically is a challenge, the result of this added dimension is clear. In this case, each cell of the OLAP cube represents the intersection of an account, a year, a region, and a product.

Rollups

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

Hierarchy

Our example cube has only one level in each dimension. The Time dimension consists of one level containing three members (years); 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, in a particular year. So the time dimension could be examined at three levels: month, quarter, or year. Likewise, a particular office could record each sale, located in a particular city, in a particular region. So the location dimension might also have three levels: office, city, and region.

As we said earlier, the categories found at each level of a dimension are called members. Multi-level dimensions might be envisioned 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. For example, here is a diagram representing a portion of a typical time dimension, with its various levels and members:

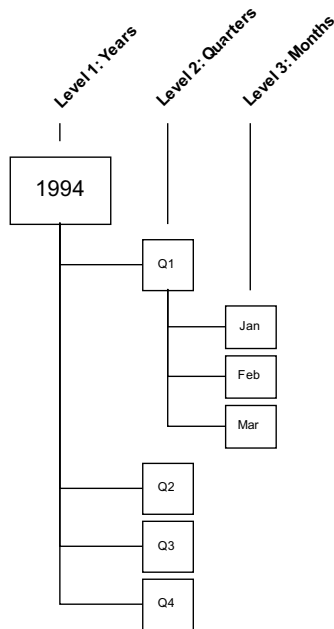


Illustration of a Time Hierarchy

Each box in the diagram represents a unique member. If you've used the PeopleSoft Tree Manager before, this diagram should look very familiar. In fact, PeopleSoft trees can play an important role in defining the hierarchy of an OLAP cube.



For more information about Tree Manager, see Introduction to Tree Manager.



For more information about using trees, see PeopleSoft Trees

Consolidation

Viewing a dimension's hierarchy tells you about the organization of its members; but there's another facet of the dimension to consider. We need to know how the values found under child members should be consolidated into the value of their parent members. For example, are the children added together to equal the parent? This 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.)

To give an example of this, we use our original cube, in which we show units sold dimensioned by time and product.

Product	1994	1995	1996
Widgets	3000	6500	8200
Gadgets	1200	1450	3000
Doohickeys	2500	3400	2000
Whatzits	500	670	1300

Units Sold Dimensioned by Time and Product

Now, lets add a second level, quarters, to the time dimension.

Product	1994				1995				1996			
	Q1/ 94	Q2/ 94	Q3/ 94	Q4/ 94	Q1/ 95	Q2/ 95	Q3/ 95	Q4/ 95	Q1/ 96	Q2/ 96	Q3/ 96	Q4/ 96
Widgets	400	700	1500	400	1200	1500	2000	1800	2200	3000	2500	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
Whatzits	125	125	125	125	125	200	225	120	170	230	400	500

Units Sold Dimensioned by Time Hierarchy and Product

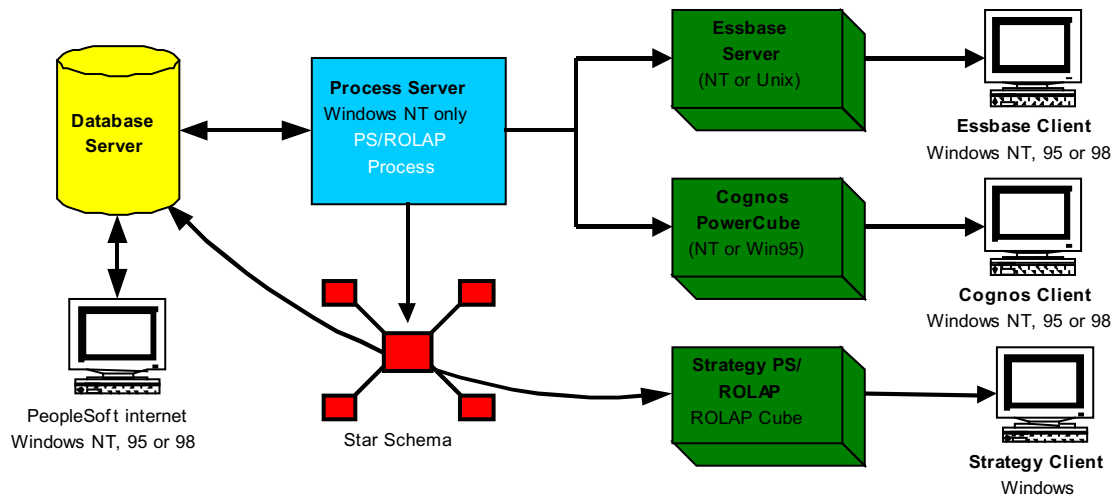
How does the data represented at the Quarter level consolidate into the Year level? It's simply added together. So the 1994 rollup is Q1/94 + Q2/94 + Q3/94 + Q4/94.

However, you might also find dimensions in which certain members are to be subtracted from their siblings, such as in a profit dimension. In such a dimension, let's say you have two members at the first level, margin and total expenses, both of which are reported as positive values. To find your total profits, you wouldn't add margin and total expenses; you would subtract total expenses from margin.

In Essbase a dimension could have two identical hierarchy rollups with two different totaling results even though both hierarchy rollups have identical tree structures, nodes and leaves. This would be possible if for example your first tree is set up ABC, while the second tree is ADC. For example, A, B, C, D have the fact data values: 2, 5, 10, 20 respectively. The total for the 1st hierarchy A,B,C will be 17 since the total = 2 + 5 + 10. The total for the 2nd hierarchy A, D, C will be 10, since each parent gets its total from its children. Since C has the fact value 10, the total for C is 10. Since 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.

Understanding Cube Manager Architecture

The following diagram shows what part of the cube building process happens in each part of the system.



The Cube Manager Big Picture

1. Cube Manager extracts data from the application database.
2. Cube Manager writes to the OLAP data store.
3. Analysis is performed directly against the OLAP data store using third-party software, or PS/ROLAP Decision Master, which is a PeopleSoft product, can also be used

Designing with Cube Manager

It's possible to build an OLAP database out of PeopleSoft data without using Cube Manager. In fact, customers have implemented custom OLAP solutions for use with previous PeopleSoft releases. However, Cube Manager provides several very important benefits when linking between your PeopleSoft application data and your OLAP platform.

Cube Manager provides a framework for modeling your OLAP cubes through its design tools. This framework supports the definition of all dimensions, attributes, measures, and cubes that you might want to build from PeopleSoft sources. All of these definitions can be shared across cubes to ensure that OLAP results are consistent across your enterprise.

Components designed in Cube Manager can also be used across all target OLAP platforms supported by PeopleSoft. Cube Manager provides all these benefits while achieving a nearly platform-independent solution. The great majority of a cube's design attributes can be applied to Hyperion Essbase, Cognos PowerPlay, or CA Strategy PS/ROLAP. Cube Manager also enables you to leverage your existing PeopleSoft metadata to define the cube structure.

PeopleSoft Metadata

Metadata is data that defines data. It conveys information about how data is formatted, structured, and stored. In an OLAP cube, metadata defines the cube's dimensions, levels, members, member attributes, and their interrelationships. Cube Manager uses two types of PeopleSoft structures, trees and queries, to help 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 identify individual departments in your organization. A tree could be built for the DEPTID field that defines the organizational hierarchy that specifies how each department relates to the others—that departments 10700 and 10800 report to the same manager, department 20200 is part of a different division, and so on.

It's easy to see how PeopleSoft trees can be used to define cube structure. Like cube dimensions, trees are composed of levels and members. (In Tree Manager, members are called nodes and leaves.)



For more information on how Cube Manager uses the metadata in trees, see [Trees](#).

PeopleSoft Queries

PeopleSoft queries are SQL statements created by our Query PeopleTool. These SELECT statements are used to return field values based on certain criteria. The data returned by a PeopleSoft Query can be secured by the standard PeopleSoft security mechanism. Also, data returned by Query can be returned in any of the database-supported globalized formats.

Queries are used in a number of ways to define an OLAP cube. They may be used to define rollup structure, although this process is not nearly as straightforward as using a tree. Queries can also be used to specify member attributes. For example, you can set an attribute that automatically flips the sign of member values (sometimes necessary for accounting purposes). Finally, queries are always used to populate OLAP cubes with data; the query results are the rows of data that fill the cube's cells.

Security Queries

Because the data in an analytic database is usually composed of all possible information that might be used in analysis, sometimes, sensitive data is included in the cube. To prevent prying eyes from seeing that sensitive data in a star schema, Cube Manager enables you to specify a query that secures the dimensional data. Defining a query with a particular structure and associating it with a dimensional rollup can secure certain members. In PS/ROLAP you can use security queries to restrict what is seen in the dimension when you view a cube by Administering Access Controls on the ROLAP Server.

Security queries are of the inclusion type. Only the value pairs returned by the query have access to the data. Cube Manager uses security queries by joining the resulting values of the query with a specified dimension in order to create a dynamically created temporary table of possible members visible to a given user group. To accomplish this, you must define a query that returns two fields. One field contains the member of the dimension that is to be included for viewing. The other field is the LDAP class that has access to this value.



In Release 8, security queries are only available with EPM workbenches. We plan to add this functionality to our other target platforms in future releases.



For more information about security using LDAP for EPM workbenches, see your workbench documentation.

Level Attribute Queries

The PS/ROLAP engine supports a series of functions that can be applied to time dimensions. In order to provide the metadata necessary to support these functions, the level attribute query is used.

Process Scheduler Integration

Process Scheduler includes a process type definition specifically for use with Cube Manager. This is the Cube Builder process type, and it is invoked whenever a user launches the process to create a cube from the standard run control page. Depending on the OLAP tool you specify, during this process, the data and metadata are translated into a format understood by Essbase, PowerPlay, or PS/ROLAP.



For more information about the specific steps for invoking a process for Cube Manager, see Setting Up Process Scheduler.

Supported OLAP Tools

Each PeopleSoft customer has unique reporting and analysis needs. To address these needs, PeopleSoft provides support for various OLAP databases and tools: Cognos PowerPlay, Hyperion Essbase, and Strategy PS/ROLAP.

If you haven't chosen an OLAP platform, the following descriptions should help you decide which one best suits your needs.

PowerPlay

PowerPlay is usually considered a desktop OLAP solution that is appropriate for individual users who might want to work offline. It includes a product called Transformer that builds the extremely portable data cube (it's easy to attach it as a file to electronic mail, and so on). PowerPlay includes two components: a database engine component (the PowerCube) and an end user component. This front-end component can be used not only for PowerPlay databases, but also for other OLAP databases, including Essbase.



You can use use PowerPlay 6.6, just you used PowerPlay 6.5, with Cube Manager 8.0.

For PeopleTools 8.12, we have enabled the use of PowerPlay time dimension. In PowerPlay it is possible to define a dimension to be a time dimension, a level can be set at year and so on. In this way the reports you create using this dimension, can actually do computations and comparisons based on the current year versus last year, or current period versus last period.



For more information about defining the time dimension for PowerPlay, see Setting Time Dimensions.

For PeopleTools 8 and 8.1, we have enabled the use of the PowerPlay 6.6 Enterprise Server software and its increased functionality. Cube Manager builds and adds cubes to the Enterprise Server, but only if you are using PeopleTools 8 or PeopleTools 8.1.

If you are using PeopleTools 7.x, you can still use the Enterprise Server software. However, Cube Manager can't register the cubes automatically to the Enterprise Server. You have to build the cubes using Cube Manager and then register them manually to the Enterprise Server.



For more information about PowerPlay 6.6 Enterprise Server software, see the HTML_readme files under each component directory located on your PowerPlay Install CD.

Using Cube Manager in PeopleTools 7.x with PowerPlay 5.2

The PowerPlay product consists of two main components for version 5.2: the Transformer and PowerPlay for Windows. The Transformer builds PowerPlay Cubes and creates the .MDC file. PowerPlay for Windows is the reporting tool for end users.

The biggest difference between PowerPlay version 5.2 and 6.6 is the addition of the Enterprise Server with version 6.5.

The following illustration shows how PowerPlay 5.2 worked with PeopleTools 7.x and continues to work with PeopleTools 8.1



Using Cube Manager in PeopleTools 7.x with PowerPlay 6.6 Personal Server and Transformer

You can use PowerPlay 6.6 Personal Server just as you used PowerPlay 5.2. There isn't any difference in the way you build cubes for PowerPlay 6.6 using Cube Manager in PT 7.x. You can consider this a new version of the PowerPlay software with the same architecture as PowerPlay 5.2.

Using Cube Manager 8 with PowerPlay 6.6 Enterprise Server Edition

There are several new components in PowerPlay 6.6 (Enterprise Server version) that work together to give you more functionality and to make PowerPlay cubes available to remote users. Only the pieces that Cube Manager uses are described in this section. Cube Manager in PeopleTools 8 has been modified to work with these new pieces.



You can still use PowerPlay 6.6 with PeopleTools 7.x, but you don't have the extra functionality of being able to add a cube automatically to the Enterprise Server through Cube Manager.

Enterprise Server

This is an NT service that maintains cubes at a given location. Users from remote locations connect to this service and open the cubes in their choice of PowerPlay for Windows, PowerPlay for Excel, or PowerPlay for Web. The cube must be added to the Enterprise Server to manage it. The Enterprise Server needs a port number, server/machine name, and a password, so that you can connect to it and add or remove cubes.

Like any other NT services, it is the responsibility of the user to start the EP Server before other users can connect to it. An NT service can be started manually by selecting Start, Control Panel, Services, or by setting the service to start automatically when the system starts.



For more information about starting the Enterprise Server, see the PowerPlay Enterprise Server documentation.



Cube Manager and the Enterprise Server must run on the same machine.

PP Enterprise Server Administrator

This is a Windows application that can talk to the Enterprise Server, either locally or remotely, and can display the cubes that the Enterprise Server is maintaining. Depending on the security privileges, you can add or remove cubes from the Enterprise Server using this application. You can also change passwords from this application.

PPAdmtool

This is a command-line program that can talk to the Enterprise Server. You can add or remove cubes from the Enterprise Server using this program. To execute any commands using this program, you need to specify the server/machine name, port number, and password.

The difference between the Enterprise Server Administrator and PPAdmtool is that PPAdmtool runs at the command prompt. The Enterprise Server Administrator is a Windows application that must be started before viewing or manipulating the cubes. You can use PPAdmtool from the prompt to connect to, view, add, or remove cubes, without having to perform the extra step of starting a program.

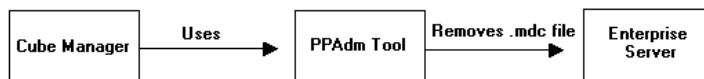
PPApplications

These are the actual Windows applications that can open the cube .mdc files. This can be any of the applications PowerPlay for Windows, PowerPlay for Excel, or PowerPlay for Web. These applications either need direct access to the .mdc file (locally or on network) or access to the Enterprise Server that is maintaining a particular cube.

How Cube Manager 8.0 Uses the Enterprise Server

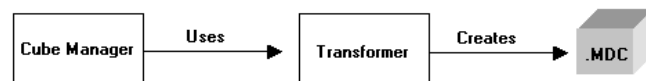
This example shows how Cube Manager uses the new functionality of PowerPlay 6.6 to manage a cube that exists on the Enterprise Server.

Step 1



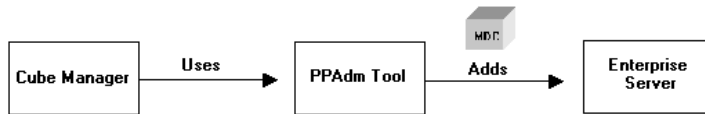
PPAdm Tool Removes the .mdc file From the Enterprise Server

Step 2



Cube Manager Uses the Transformer to Create the .mdc File

Step 3



Cube Manager Uses PPAdm Tool to Add the .mdc file to the Enterprise Server

Cube Manager's Platform Options panel for PowerPlay has been modified to take the extra information needed to talk to the Enterprise Server.

Essbase

Essbase is a robust, client/server-based product, the scope of which goes well beyond individual users to the level of a data mart. While it has many powerful features, including its own scripting language, it might be somewhat more difficult to implement than PowerPlay. However, because it supports metadata updates, Essbase enables you to keep a persistent data store, whereas PowerCubes have to be recreated whenever any metadata changes.

Essbase includes an Excel add-in with which you can view OLAP cubes using Microsoft Excel—very much like PS/nVision. Essbase cubes can also be accessed using PowerPlay as a front end. Essbase is shipped separately from PeopleTools (there are two licensing components for the integration and the end-user product).

PS/ROLAP

PS/ROLAP is the set of UNIX-based ROLAP services is used as a reporting and analysis tool. This engine is used to analyze trends and data within data warehouses. The PS/ROLAP integrated with PeopleSoft DecisionMaster was specifically designed for viewing analytic content delivered by PeopleSoft Workbenches.

Designing and Building OLAP Cubes

Identify your requirements. This is the most important part of any analytical application. An analytical application always produces results, regardless of whether those results are meaningful.

In this stage, you decide what aspects or processes of your business you want to capture. Some examples could be sales activity, claims processing, or marketing return on investment. It might be that there are several aspects of your business that you want to analyze that are unrelated. Do not try to make one cube for all of the aspects. Each set of related information should be treated as a single cube.

Next, you identify the measures that you use to quantify those results, such as sales amounts in dollars or units. This information is almost always numeric. Then, you identify the criteria with which you want to view the data and the granularity of the data. These criteria form the dimensions. The most common dimensions are time, accounts, geography, products, and department.



These steps are not covered in this documentation. For more information, see your Essbase, PowerPlay, or PS/ROLAP documentation.

Designing the metadata.

With the PeopleSoft OLAP solution, you use PeopleSoft metadata—in the form of trees and/or queries—to define rollup structure, member attributes, and data. In some cases, the necessary trees or queries might already exist.



For more information, see Designing Cube Metadata.

Using Cube Manager.

In these steps you use the trees and queries created in the previous stage to define cube dimensions, analysis models, and templates. These steps all involve the Cube Manager pages and are covered in a later chapter.



For more information, see Designing Cubes and Building Cubes.

CHAPTER 2

Designing Cube Metadata

There are several kinds of metadata that define an OLAP (online analytical processing) cube: dimensional metadata (hierarchy and members), member attribute metadata (consolidation method, sign flip, and label), as well as the cube metadata (dimensions and measures). You use PeopleTools object trees and queries to describe all of this metadata to Cube Manager.



For more information about trees, see [Implementing Dynamic Tree Controls](#).

Trees

Metadata that exists in PeopleSoft trees can be particularly useful when designing cube dimensions. The main reason to use an existing tree is to leverage the rules 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 a cube definition generates the automatic evolution of your data used for data analysis.

Cube Manager leverages the information already stored in your PeopleSoft trees as an outline upon which to build each dimension. Using the Cube Manager's Dimensions - Rollup Inputs Page, 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.



What we call levels are called generations in Essbase terminology.

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, but not exactly, what you want the dimension to look like, you might want to make a copy of the tree and modify the copy.

If the hierarchy you want to use is a subset of an existing tree, you don't have to create a whole new tree. Cube Manager enables you to use a subset of an existing tree by specifying a starting

node (the Top Node on the Dimensions - Rollup Inputs Page), as well as the number of levels below the top node of the tree to include in the hierarchy. More than one tree belonging to the same business unit may be used to define a single hierarchy.

In addition, if a tree doesn't provide quite the proper structure you need for a dimension, you can add members, attributes, and levels by using one or more queries to provide the additional metadata.

Cube Manager can also handle unbalanced trees; an unbalanced tree is where all of the details are not represented at the same "level" in the tree. Cube Manager has the ability to balance out an unbalanced tree so that it pushes down all details to the lowest level and creates dummy nodes at the highest level. This action in turn makes an unbalanced tree become balanced.

Because you cannot have duplicate member names in a dimension, unless those members are shared, you can't have duplicate tree node names in your tree metadata. Cube Manager treats uppercase and lowercase characters as distinct, so the names "ABC," "Abc," and "abc" are all considered unique member names. However, Essbase offers an option to change all member names to uppercase. If this option is enabled, you create problems in Cube Manager with members that are identical except for their letter casing.



Cube Manager permits duplicate node names, if the duplication can't be avoided. See Rollup Options Page for details.

Cognos does not allow duplicate values at any level within the same hierarchy. Cognos wants the member values to be unique in the same hierarchy. Because of this reason duplicate values in the same hierarchy are dropped while generating the MDL. Alternate hierarchies can have detail values (leaves) which appear in the other hierarchies. However, the non-detail values (non-leaf or node) cannot be the same as in any other hierarchy. This will cause an error to be returned by Cognos when the cube is being created.

In Essbase a dimension could have multiple rollups. The resulting total for the 1st rollup is calculated differently from the resulting totals for the rollups after the 1st rollup. For example, there is a dimension with two rollups. Two different trees are used for these two rollups. The first tree is set up as A-> B-> C while the second tree is A-> D-> C. Let's assume A, B, C, D have the fact data values: 2, 5, 10, 20 respectively. The total for the 1st rollup A, B, C will be 17 since the total = 2 + 5 + 10. The total for the 2nd rollup A, D, C will be 10 since each parent gets its total from its children. Since C has the fact value 10, the total for C is 10. Since 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.

Queries

There are five kinds of queries you can create to use with Cube Manager, all of which must be defined as User (Ad Hoc) queries, as opposed to role queries or database agent queries:

- Dimension queries define cube dimension structure, and you can use them instead of, or in addition to, trees. You can create two different types of dimension queries: narrow and wide.



For more information about when to use each type, see Dimension Queries.

Dimension Queries are useful in situations where hierarchical information already exists in your database, but you don't have an existing tree corresponding to this information and would rather not build one.

- Attribute Queries define optional attributes for each member.
- Security queries define optional row-level security for members of a dimension defined in PS/ROLAP (relational OLAP).
- Level Attribute Queries define metadata about levels only to PS/ROLAP.
- Data Source Queries define the data that populates your cube. Each result row of a data query must map to your dimensions and measures as defined in the cube.

Because you cannot have duplicate member names in a dimension, unless those members are shared (Essbase is the only platform that handles shared members), you can't have duplicate query column names in your query metadata. Cube Manager treats uppercase and lowercase characters as distinct, so the names "ABC," "Abc," and "abc" are all considered unique member names. However, Essbase offers an option to change all member names to uppercase. If this option is enabled, you create problems with members that are identical except for their letter casing.



Cube Manager does permit duplicate node names, if the duplication can't be avoided. See Rollup Options Page for details.



For more information about PeopleSoft Queries, see PeopleSoft Query.

Dimension Queries

Dimension queries enable you to define the dimension structure using query results instead of, or in addition to, a tree. However, keep in mind that you are using queries to create a tree-like structure. For narrow query definitions, each dimension query maps child members (member) at a particular level to parent members (parent) at the next higher level. For wide query definitions, only one query is needed to build the dimension. Optionally, these queries might contain extra members or level attributes, as well as the relationship information.



For more information about Attribute Queries, see Attribute Queries.

Hierarchical information can be conveyed in one of two ways: by parent/child relationship (a narrow query) or by level specification (a wide query).

Narrow Query Definition

When using multiple queries to define dimensional structure, the first query you specify defines the first two levels of the dimension. To add lower levels, you must write one additional query for each additional level.

Suppose you want to build a department dimension that contains the following departments within an organization:



The levels of the organization might be described as follows:

Level 1 (top of dimension)	Level 2	Level 3
ALL DEPARTMENTS	DEVELOPMENT	100
		200
	SALES	300
		400
		500

To create the dimension, you must write two queries to provide the information above: one to define the child members at level 2 and one to define the child members at level 3.

Query 1

Parent	Member (Child)
ALL DEPARTMENTS	DEVELOPMENT
ALL DEPARTMENTS	SALES



During the build, Cube Manager knows that child members with an unspecified parent become level 2 members, directly under the top of the dimension. You can create an empty column in a query by adding a blank ("") expression. Be sure to enter some meaningful text for the Heading Text and Unique Field Name—such as "Top of Dimension"—so you can easily identify the blank column when mapping query columns to dimension levels.



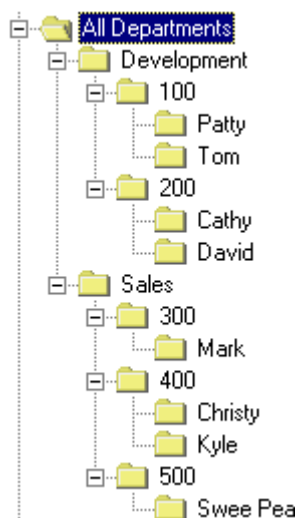
For more information, see PeopleSoft Query.

Query 2

This query must return one column for the second-level members (DEVELOPMENT and SALES) and one column for the third-level members (100, 200, 300, 400, 500). The query results look like this:

<i>Parent</i>	<i>Member (Child)</i>
DEVELOPMENT	100
DEVELOPMENT	200
SALES	300
SALES	400
SALES	500

If you want to add additional levels, you write one query for each additional level. Building on the previous example, let's assume that you want to create an employee dimension instead of a department dimension. You want to include a fourth level showing the employees in each department:



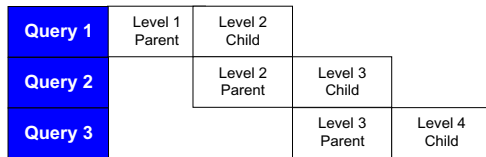
Now the levels of the organization might be described as follows:

Level 1	Level 2	Level 3	Level 4
ALL DEPARTMENTS	DEVELOPMENT	100	Patty
			Tom
		200	Cathy
			David
	SALES	300	Mark
		400	Christy
			Kyle
		500	Sweet Pea

In addition to the two queries you create, you need to create a third query with the following results to add the fourth level to the dimension:

Parent	Child
100	Patty
100	Tom
200	Cathy
200	David
300	Mark
400	Christy
400	Kyle
500	Sweet Pea

The diagram below illustrates how you supply three queries to create a dimension with four levels:



Four level dimension created using three queries

Your dimension queries should always produce at least two columns in the result set: one for the parent and one for the child (or member). Except for the top-level query, a parent column for a given level must always correspond to the child column of the previous level.

Optionally, attributes might be included in the query. All attributes specified in a narrow query are associated with the field specified as the member. Attributes for the topmost member must be added using an attribute query.

Wide Query Definition

When defining your dimension with a wide query, you need to follow the same logic you use with narrow queries.

Using the same example for narrow queries above, here is a graphical look at the dimension we want to build with a wide query:

Level 1	Level 2	Level 3	Level 4
ALL DEPARTMENTS	DEVELOPMENT	100	Patty
			Tom
		200	Cathy
			David
	SALES	300	Mark
		400	Christy
			Kyle
		500	Sweet Pea

Instead of writing multiple queries to build this dimension, you need write only one that contains this information. The query result set might look like this:

Level 1	Level 2	Level 3	Level 4
ALL DEPARTMENTS	DEVELOPMENT	100	Patty
ALL DEPARTMENTS	DEVELOPMENT	100	Tom
ALL DEPARTMENTS	DEVELOPMENT	200	Cathy
ALL DEPARTMENTS	DEVELOPMENT	200	David
ALL DEPARTMENTS	SALES	300	Mark
ALL DEPARTMENTS	SALES	400	Christy
ALL DEPARTMENTS	SALES	400	Kyle
ALL DEPARTMENTS	SALES	500	Sweet Pea

As with narrow queries, the query can return attributes that can be associated with members. However, in the case of wide queries, the attributes can be assigned to any member.

Attribute Queries

Attribute queries can be used to set optional attributes on the members within a dimension. Attributes can be set on either node members or detail members of a dimension.

An attribute query should return at least two result columns: one that identifies the members whose attributes you want to define and one for each type of attribute to be set.

Member	Attribute 1	Attribute 2 (Optional)	Attribute 3 (Optional)
1000	XXX	XXX	
1100	XXX	XXX	XXX
1200	XXX		XXX
1300	XXX	XXX	

Most attribute types are platform-specific; they are used by one of the target platforms, but not all. Each attribute type uses its own set of conditions to determine whether to apply the attribute. In some cases, for the attribute to be applied, the attribute column must contain a specific value. In other cases, the attribute column must simply not be blank. An attribute query can return a subset of the members, in which case, the specified attributes are set only on those members returned.

If an attribute query is on a table that uses a Set ID, you must select only the Set ID in which you're interested, because there is only one member ID field to tie the query results to the dimension.



In the example above, one query is used to set three attributes. If you prefer, you could create three separate queries in which you return the member and one attribute.

General Attributes

There are numerous types of attributes you can define using queries and Cube Manager. Valid attribute types are discussed below. Refer to your Essbase/PowerPlay/OLAP Services/PS/ROLAP documentation.

Flip Sign

This attribute refers to the reversal of + and – signs for the member, which is sometimes necessary for accounting purposes. Valid values are blank (don't flip sign) or non-blank (flip sign). In a star schema, an attribute column is populated with a value of – 1 when a member is to be sign flipped. In PowerPlay and Essbase, the data that is populated in a flip-signed cell will have the opposite sign of the source query.

Label

This attribute provides a description for a member. In an Essbase database, this equates to the default alias. In a PowerPlay PowerCube, it equates to the label. For star schemas, Microsoft OLAP Services and PS/ROLAP, the label is a description of the member.



For more information **about** prefix/suffix options, see Dimension Options Page.

User-Defined

This attribute applies to Essbase only. Essbase enables you to specify user-defined attributes for members. You can then use calculation (calc) scripts to search for and manipulate members having particular user-defined attribute values.

Cube Manager supports user-defined attributes. See Attribute Names - Attribute Name page. To design your own user-defined attributes, select Design, Attribute Names from the Cube Manager menu.

Essbase Attributes



For more information **about** any of the following attributes, see your Essbase documentation.

Currency Category

Currency category applies to Account-type dimensions. This attribute specifies a member that requires currency conversion to a specific category type. In the attribute query field, you supply the type of conversion required. This could be a value, normally in dollars.

Currency Conversion Type

Currency conversion applies to account dimensions. This attribute equates to the Currency Conversion radio buttons on the Account tab of the Attributes dialog page for an account dimension in the Essbase Application Manager.

If the Currency Category attribute is set to a non-blank value on a member, the Currency Conversion Type attribute is automatically applied with that value. If the Currency Category attribute is set to blank, the Currency Conversion Type attribute is automatically set to inherit-use ancestor. If the Currency Category attribute is not applied at all, the Currency Conversion Type attribute is automatically applied with a value of no conversion ("").

If you set this attribute manually, valid values are blank (no conversion) or non-blank. If the query returns a non-blank value, that value is used as the Currency Category.

Currency Name

Currency name applies to country dimensions. The value of this attribute defines what type of currency the country or market region uses. This value is used in a currency cube to identify the type of currency.

Data Storage

This value tells Essbase what type of storage to allocate for the member. Valid values are 0/blank (store data), 1 (never share), 2 (label only), 3 (shared member), 4 (dynamic calculation and store), and 5 (dynamic calculation, no store). Cube Manager 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 attribute. Valid values are *Blank* (set) and *non-Blank* (don't set).

Time Balance

This attribute 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.

Unary Operator

Essbase refers to this as the consolidation attribute. This attribute 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.

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

PowerPlay Attributes

Long Description

This equates to any valid description in PowerPlay, meaning that it can contain unlimited text.

Short Label

This equates to the short name. Valid values are any valid PowerPlay short name.

PS/ROLAP Attributes

Sequence in Year

This time dimension attribute is a numeric value that tells the PS/ROLAP engine the member's sequence in the year.

Sequence Number

This numeric value uniquely identifies the exact position in the time dimension.

Current Period

The value of this time dimension attribute is used in time calculations. If the value is *Y*, the member is part of the current period.

Level Attribute Queries

Level attribute queries enable you to specify specialized metadata for the levels of a rollup for the Sterling PS/ROLAP engine. The attributes identify characteristics of the level itself.

Resolution Type

The value of this time dimension attribute identifies a specific time grouping as found in this table.

<i>Value</i>	<i>Description</i>
0	Undefined
1	Year
2	Quarter
3	Month
4	Week
5	Fortnight
6	Semi-annual
7	Day
8	Bi-month
9	Bi-quarter
10	Bi-week
11	Semi-month
12	Semi-quarter
13	Semi-week

Prior Resolution Count

This numeric time dimension attribute helps to correlate the different time resolutions for the time functions in the ROLAP engine. The value tells the ROLAP engine how many of one resolution exist within another resolution. Cube Manager uses default values, if this level attribute is not specified.

Data Source Queries

Data source queries define the data you bring into the cube. Writing a data source query is straightforward. The query simply needs to return one column for each dimension and one column for each measure. Let's assume that you want to build a data source query for a cube containing Amounts dimensioned by Account, Department, and Period. The output of your query has four columns, as shown below.

Dimensions			Measures
Account	Dept	Period	Amount
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX

You have the option of using several queries as the data source for a single cube, thus defining multiple measures. Every data source query used must include an output column for every dimension used and for at least one measure. However, it's not required that you provide an output column for every measure in every data source query.

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

Results of Query #1

Account (Dimension)	Dept (Dimension)	Period (Dimension)	Budget Amount (Measure 1)
1000	DEV	Q4 1997	4000
1100	SALES	Q4 1997	6000

Results of Query #2

Account	Dept	Period	Actual Amount (Measure 2)
1000	DEV	Q4 1997	3000
1100	SALES	Q4 1997	5000

Resulting Cube using Query #1 and Query #2 as Data Sources

<i>Dimensions</i>			<i>Measures</i>	
<i>Account</i>	<i>Dept</i>	<i>Period</i>	<i>Budget Amount</i>	<i>Actualize Amount</i>
1000	DEV	Q4 1997	4000	3000
1100	SALES	Q4 1997	6000	5000

CHAPTER 3

Designing Cubes

The preceding chapters discussed the concepts and general steps involved in designing cube metadata. This chapter describes specifically how to use PeopleTools Cube Manager to design a cube.

Remember that before beginning to use Cube Manager, it is a good idea to define the specific goals and results you expect from online data analysis using the guidelines discussed in previous chapters. After defining your goals, you should design the PeopleSoft trees and queries that are appropriate for creating both the structure and the data of the cube you plan to build. See Designing Cube Metadata.

Designing a cube with Cube Manager can be done from the top (starting with the cube) or from the bottom (starting with the dimension). The following chapter focuses on creating a cube from the dimension and working up to the cube instance definition.

Overview of Cube Components

Many different kinds of components make up the cubes you build with the Cube Manager.

Dimensions	The most basic component of a cube is a dimension. When you create a dimension, you specify the PeopleSoft metadata to be used to create the dimension's rollup structure. Cube Manager dimensions are platform-independent, and can be reused again and again to build different cubes. For more information, see Dimensions and Members.
Attributes	Dimension members may have various attributes associated with them. For example, if you have a product dimension, you might want to add color as an attribute of the actual product.
Level Attributes	Levels within the PS/ROLAP (relational online analytical processing) time dimension can have associated attributes. An example of time dimensions having attributes is the number of accounting periods in the year level of the time dimension.

Cube Definitions	<p>A Cube Definition defines the basic structure of a cube. You select the dimensions and measures that make up the cube and the data source queries that populate the members and cube cells with data. Like dimensions, Cube Definitions are platform-independent and can be reused.</p>
Cube Instance Definitions	<p>Cube Instance Definitions are platform-specific. In a Cube Instance Definition, you specify the platform for which the cube will be built, and choose the Cube Definition on which the cube will be based. You then set various other options, specific to the chosen platform.</p>
Filters	<p>Filters are used on the cube instance definition to limit the contents of the cube being built. This enables a single cube definition to be used when building different data marts. For example, you might wish to define the rules for building a generic profitability cube, but you might want to give each region in your company a cube with its own information. Filters provide a mechanism for specifying the subset of the possible cube data to be contained in a specific cube.</p>
Platform Options	<p>Each of the OLAP target databases has an associated set of options with which Cube Manager controls how the cube is built. These options can provide Cube Manager with any required security information or how to allocate database space for dimensions in the cube.</p>
Bookmarks	<p>Bookmarks identify a set of facts in the cube. They are used for determining whether facts are being loaded for the first time or whether they are being re-loaded. Cube Manager uses bookmarks to ensure that when facts are being reloaded into a cube, they are not counted twice.</p>
Run Control IDs	<p>When you've defined your dimensions, Cube Definition, and Cube Instance Definition, you're ready to start the build process. Process Scheduler runs this process and—like all Process Scheduler processes—you create a run control ID that defines the specifics of the process:</p> <ol style="list-style-type: none">1. Specify which Cube Instance Definition you want to use to build the cube.2. Choose whether the build process is to create a new cube or update an existing one.3. Specify a post-build script to run on the cube.4. Define any query bind variables that you want to use to limit the data source queries during the build.

Cube Instance

A Cube Instance is the output of the Cube Manager. It is a physical cube that is built by the Cube Builder process in PowerPlay, Essbase, or PS/ROLAP.

Designing a Cube

The whole process of building a cube starts with designing Cube Metadata: building, testing, and refining the trees and/or queries that feed data from your PeopleSoft application to any cubes you create.

After you establish your goals and create the necessary trees and queries upon which the resulting cube will be built, you use PeopleTools Cube Manager to begin designing a cube. The first step is to define the dimensions of the cube using the Dimensions - Dimension Page. At this stage, you're beginning to build the structure of each dimension that comprises your cube. You use the Dimensions - Rollup Inputs Page to determine how the source trees and queries are defined for each rollup. You're not mapping data into the cube at this point.

The second step is to define the cube using the Cube Definitions - Cube Definition Page. Then specify the source of the data using the Cube Definitions - Cube Inputs page.

Dimensions - Dimension Page

Usage	Define the structure of each dimension.
Object Name	DIMENSION
Navigation	Design, Dimensions, Dimension
Prerequisites	You must first create the trees and queries on which to build your cube.
Access Requirements	Enter a Dimension ID.

Dimension **Rollup Inputs**

Dimension ID: DEPARTMENT [Dimension Options...](#)

Description: DEPARTMENT *Dim Type: Standard

Rollup Name: DEPARTMENT 1 [Rollup Options...](#) + -

Rollup Levels		View All First 1-3 of 3 Last	
	*Level Name		
1	Division	+	-
2	Group	+	-
3	Department	+	-

Save Return to Search Add Update/Display

[Dimension](#) | [Rollup Inputs](#)

Dimensions - Dimension page

Dimension ID

The dimension ID was entered in the search page. The dimension ID uniquely identifies the dimension of the target OLAP database.

You cannot rename or delete a Dimension ID once it has been saved.

Description

The **Description** defaults to the **Dimension**, but you can provide another description. This is used as the description of the dimension in the resulting cube.

Dimension Options

See Dimension Options Page.

Dimension Type

Choose one of the following dimension types:

Standard: Select for every kind of dimension that does not contain account, time, currency, or country data. You can have many standard dimensions in each cube.

Account: Select if the dimension is based on account information. You can only have one account dimension for each cube.

Time: Select if the dimension is time-based. You can only have one time dimension for each cube. PS/ROLAP requires a time dimension in each cube. See Setting Time Dimensions.

Currency Partition: This is only valid for Essbase. Select if the dimension is based on currency. You can have only one currency dimension for each cube.

Type: Not supported in PeopleTools 8.1.

Country: This is only valid for Essbase. Select if the dimension is based on countries. You can only have one country dimension in each cube.

For more information **about** how to take advantage of non-standard dimension types, see your Essbase or PS/ROLAP documentation.

Rollup Name

The default Rollup name is based on the **Dimension ID**. You can change the Rollup name, but the name cannot be identical to the **Dimension ID**.

The **Rollup Name** becomes the top node of the dimension if more than one rollup is specified in the dimension. If you have multiple rollups, rolling up to a single top node, this name only is used for the alternate rollups. A node is created for each Rollup as a child of the dimension's top node.

Rollup Options

See Rollup Options Page.

Level Name

This field is required. If you want to explicitly define level names in the rollup, specify them in the **Rollup Levels** group box. Level names defined in this manner override the level names that might be returned from the tree or any names that Cube Manager creates as defaults. You are required to specify levels to be used in mapping wide queries or for defining aggregates.

When you enter multiple level names, a level number is assigned to the declared level name in the order in which the level name is entered. This number also populates the Dim Input Field Page.



Click on the **Add** button to add a row for an additional level name.



Click on the **Remove** button to remove a row, deleting that level name.

The structure of a dimension is based on a PeopleSoft tree you build, but dimensions can also be based on queries. You must create one dimension definition for each dimension of your cube.

Because dimensions are independent of cubes, they can be used in any number of cube definitions. Where possible (such as in star schemas), Cube Manager creates dimensions that can be shared across conforming dimensions (unless the underlying data relies on different business units). This method ensures consistent results across different data marts.

To design a dimension:

1. Select Design, Dimensions, Dimension.

Choose whether to Add a new dimension or to Update/Display an existing one.

2. Enter a Dimension ID and click the OK button.

You need to provide a unique name for each dimension that you define. The dimension ID uniquely identifies the dimension of the target OLAP database. After you enter a Dimension ID, the Dimensions page group appears, with the Dimension page selected.



Once you create a Dimension ID and save it, you cannot rename it or delete it. You can change the description, but you cannot remove the ID from the system.

3. Enter a Description and specify the Dim Type (dimension type).
4. Set additional parameters for the dimension by clicking the Dimension Options button.

The Dimension Options pop-up page appears.

5. Define the Rollups for the dimension.
6. Define the level name.
7. Define additional rollup parameters by clicking the Rollup Options button.

The Rollup Options pop-up page appears.

8. Define the sources of the rollup structure and attributes by clicking the Rollup Inputs tab.

The Rollup Inputs page appears. The Rollups scroll area on the Dimension page determines which rollup is displayed on the Rollup Inputs page.

Setting Time Dimensions

For a PowerPlay time dimension, you don't need to set a Tree or PSQuery as an input source, just identify the dimension as Time, define the levels, set the corresponding Date Formats and Date Functions in the Dimensions - Dimension Page. You are also able to select the Earliest Date and Latest Date for the time dimension using the Cognos PowerPlay Options.



For more information about Date Function and Date Format, see your Cognos PowerPlay Transformer documentation.

If you are creating a time dimension specifically for another platform, you should proceed to build the dimension as you would any other dimension, ignoring the Date Function and Format columns in the Dimensions - Dimension Page. A dimension that has both input source as well as the PowerPlay options can be used on any platform. Cube Manager will simply ignore the unnecessary information for the specific platform and use only that is required on that platform.

To define PowerPlay time dimension:

1. In the Dimensions - Dimension Page, specify the dimension type to be Time.

Dimension **Rollup Inputs**

Dimension ID: TIME [Dimension Options...](#)

Description: TIME 'Dim Type: Time

Rollup Name: TIME 1 [Rollup Options...](#) + -

Rollup Levels			
	Level Name	Date Function For PowerPlay	Date Format For PowerPlay
1	Year	Year	YYYY
2	Quarter	Quarter	YYYY"Q" Q
3	Month	Month	YYYY/MMMM
4	Day	Day	YYYY/MM/DD

Save Add Update/Display

[Dimension](#) | [Rollup Inputs](#)

Dimension - Dimension page with Dim Type set to Time

This will also set the DimType in the .mdl file to Date when the cube is finally built, allowing the cube to use PowerPlay date wizard.



When building a PowerPlay time dimensions, you are not required to complete the fields in Dimensions - Rollup Inputs Page or Rollup Options Page Prefix/Suffix information for Node and Details. If you want to build a time definition for a platform independent cube you will still want to specify them. Cube Manager will ignore the input source while building a PowerPlay cube even if it is specified.

2. Identify each level name with a DateFunction and Date Format.

This information will only be used when creating a PowerPlay Cube, the cube build process will ignore this information if the cube is intended for another platform.



Do not mix lunar calendar and standard calendar in the Date Function column when setting Levels. If you want to be able to use both calendars, you should create two different time dimensions, one for lunar calendar and one for the standard calendar.

3. Select the Earliest and Latest date options in Cognos PowerPlay Options page.

These settings are optional, the default is 1901-01-01 for the Earliest Date and 2100-12-31 for the Latest Date. If you select Current Date option it is controlled by settings in Cognos. If you select As of Date.

4. Set up a fact query to return lowest level of detail.

Dimension Options Page

Usage	Name blank members of a hierarchy so that you can see them in your cube.
Object Name	DIM_OPTIONS
Navigation	<ul style="list-style-type: none"> • Click the Dimension Options button in the Dimensions - Dimension Page. • Click the Dimension Options button in the Dimensions - Rollup Inputs Page.

Dimension Options page

Blank Member

Default name is based on the dimension name. Enter a name for each dimension that might have data that is not already associated with a named hierarchy.

Label Prefix/Suffix

This is a required field. The default value is *None*.

Both Essbase and PowerPlay require unique member names and labels. You can apply a unique prefix or suffix to each member label to fulfill this requirement. Select either *Prefix* or *Suffix* from the **Label Prefix/Suffix** drop-down list. To bypass this option, select *None*.

Not all data in the cube is associated with every dimension. For example, some data might be related to an account, but not to a product. In that case, the product dimension has blank members for that data. This means that there is an additional implied valid value of the product dimension that might not be on the product tree—blank (or No Product”). You must give a name to this blank member and make sure that it is on your dimension hierarchy, because every valid member of a dimension must have a unique, non-blank name.

You can either add the unique, non-blank name to your tree or query and then specify it in the Dimension Options page, or just enter a name and let the Cube Builder add it to your dimension for you. Then, when the Cube Manager is building the cube, it creates a new node directly off the top node of the dimension with the Blank Member name you specify. All blank members are grouped under this node name for this dimension. However, the blank member’s name does not accumulate cube data into a single-dimension member.

To define dimension options:

1. Click the Dimension Options button on the Dimension page.

The Dimension Options pop-up page appears.

2. Specify the name to give Blank Members in this dimension.

The Blank Member defaults to a name based on the dimension name you create, but you may change it, if necessary. However, keep in mind that different dimensions within in a single cube must have unique names for blank members. Thus, if a cube has a Department dimension and a Product dimension, and both of these dimensions have blank members, the Blank Member names must be different, such as [No DEPARTMENT] and [No PRODUCT].

Of course, some dimensions, such as TIME, may not have blank members. If you're sure that a dimension falls into this category, you can delete the default Blank Member value and leave the field blank. But, if you do this and a blank member is found in the dimension, it is excluded from the dimension.



Your Blank Member name can match a node/detail value elsewhere in the cube structure. This enables you to insert blank members at a level other than directly below the current dimension.

3. If desired, specify a prefix or suffix for member labels.
4. Click the OK button.

Rollup Options Page

Usage	Name blank members of a hierarchy so that you can see them in your cube.
Object Name	DIM_ROLLUP_OPTIONS
Navigation	<ul style="list-style-type: none">• Click the Rollup Options button in the Dimensions - Dimension Page.• Click the Rollup Options button in the Dimensions - Rollup Inputs Page.

Rollup Options

Node Prefix/Suffix

*Type:

Style:

Text:

Detail Prefix/Suffix

*Type:

Style:

Text:

OK Cancel

Rollup Options page

Type

In the **Node Prefix/Suffix** group box, you set prefix/suffix preferences at the node level. The prefix or suffix you specify is applied to all members in the rollup, except the lowest level members. This provides a convenient mechanism for ensuring all node members have unique names between rollups.

In the **Detail Prefix/Suffix** group box, you set prefix/suffix preferences at the Detail level. As with Nodes, choose *Prefix*, *Suffix*, or *None*. This value is only valid in the first rollup of the dimension. Secondary rollup options can only set the same selection from the **Node Prefix/Suffix** group box as the first rollup option.

Style

From the **Style** drop-down list you can choose to give your prefix or suffix no style, a custom style you enter, or a Dimension name or Level name. If you choose Dimension or Level, the text is either the name of the Dimension in the Description field of the Dimension page, or the Level Name you specify in the Rollup Levels section.

Text

Enter the **Text** to be displayed and used for the prefix or suffix, if applicable. The text will only be displayed if the prefix or suffix is specified as level.

To define rollup options:

1. Click the Rollup Options button on the Dimension page.
The Rollup Options page appears.
2. If desired, choose a prefix or suffix to add to your node (non-detail) members.
3. Choose Prefix, Suffix, or None as the Type; choose a Style; and enter the Text to be used for the prefix or suffix.
4. If desired, choose a prefix or suffix, style, and text to add to your Detail members.

- Click the OK button to save your changes, or the Cancel button to exit the page without saving your Rollup Options changes.

Dimensions - Rollup Inputs Page

Usage	Define the source of the metadata for each rollup.
Object Name	DIM_ROLLUP_INPUTS
Navigation	Design, Dimensions, Dimension, Rollup Inputs
Prerequisites	Define high-level dimensional metadata using the Dimensions - Dimension page.
Access Requirements	Enter a Dimension ID and define high-level dimension information in the Dimension page.

Dimension Rollup Inputs

Dimension ID: DEPARTMENT [Dimension Options...](#)

Description: DEPARTMENT Dim Type: Standard

Rollup View All First 1 of 1 Last

Rollup Name: DEPARTMENT 1 [Rollup Options...](#) + -

Inputs View All First 1 of 1 Last

	Src Type	Input Name	Top Node	Levels From Top	Field Map	Mapped		
1	Query	OLAP_DEPT_DIM_QRY			Field Map	<input checked="" type="checkbox"/>	+	-

Save Return to Search Add Update/Display

[Dimension](#) | [Rollup Inputs](#)

Dimensions - Rollup Inputs page

Now that the high-level dimensional metadata has been defined (identifying the dimension name, defining the number of different rollups, defining prefixing and suffixing options, and so on), the next step is to define the source of each rollup's metadata on the Rollup Inputs page.



Rollup-level names that are entered on this page only appear in an Essbase cube if the cube contains a dimension with a multiple rollup. Rollup-level names are always used in PowerPlay.

Src Type (source type)

This field is required. The default is blank. Select from one of the following options:

Calendar: This option is not yet implemented, but will enable you to use PeopleSoft Calendar.

Level Attr (level attribute): If you select Level Attribute, you can choose a query from which PS/ROLAP's time dimension attributes are defined.

Query: If you select **Query**, you are specifying how the hierarchical relationships, as well as member attributes, are defined.

Security: If you select **Security**, you can choose from a list of security queries as possible inputs that enable row-level security for PS/ROLAP.

Tree: If you select **Tree**, you are specifying how the hierarchical relationships, as well as member attributes, are defined.

Input Name

Enter the name of the **Tree** or **Query** to be used to define the dimension



Click the **Lookup Input Name** button to search for existing input names.

Node

Pre-populated with the top node from the tree based on the input name selected.



Click the **Lookup Top Node** button to search for existing top node.

Levels From Top

Specify the level number from the top of the tree (top node is 0) that corresponds to the lowest level of the tree to be included in the rollup.



Click the **Lookup Levels From Top** button to search for existing levels from the top of the node.

Field Map

Available only when the source type is a query. See Dim Input Field Page.

Mapped

This check box is selected when the input name is mapped; if the field is not mapped, the box is clear. Map the field using the **Field Map** button.



Click the **Add** button to add a row to input an additional input name.



Click the **Remove** button to remove a row, deleting the input name.

To add Rollup Inputs:

- 1. Design your high-level dimension information in the Dimension page.
- 2. Click the Rollup Inputs page or select Design, Dimensions, Rollup Inputs.

The Dimensions - Rollup Inputs Page appears.

The Rollup Inputs page is where you define cube inputs, such as queries or trees, and their associated field mappings.

- 3. In the Inputs group box, select a source input (Src Type) of Query, Security, or Tree-level attribute.
- 4. In the Input Name field, click the Lookup button to display the valid values, and choose from the available options.
- 5. If you added a tree to your dimension as a rollup input, specify a Node by clicking the Lookup button and specifying a valid value.
- 6. If you added a tree to your dimension rollup as input, select Levels from Top by clicking the Lookup button and selecting from the list.
- 7. If the source type is Query, Security, or Level Attribute, the next step is to define how each field in the result set is used in the rollup.

Dim Input Field Page

Usage	Used to map the fields to a role in the rollup.
Object Name	DIM_INPUT_FLD
Navigation	Click the Field Map button in Dimensions - Rollup Inputs Page.

Dim Input Field

Dimension ID: DEPARTMENT

Descr: DEPARTMENT 1

Input Name: OLAP_DEPT_DIM_QRY

OK

Cancel

Input Field Map

View All

First

1-3 of 3

Last

	*Query Fld Name	*Role	Level Name	Seq	Attribute	
1	Company	Q	Parent			+ -
2	Department ID	Q	Member			+ -
3	Department ID	Q	Attribute		Flip Sign	+ -

Dim Input Field (dimension input field) page

Query Fld Name (query field name) Default based on query name mapping columns from query and defining the new role.

Role Choose from the following options:

Attribute: If you select Attribute, you must also select the **Attribute Name**.

Class ID: (Not implemented yet) Based on row-level security for CA cubes only.

Member: If all sources are members, then select the level name.

Parent: If a parent exists, then no level names or numbers can be selected. The parent is the top of the dimension, and the member is the child of the parent.

Level Name	Click in the field to display a list of valid level names, and select one of the names that were specified in the Dimensions - Dimension Page. This field is only available if all roles are set to Member .
Level Num (level number)	This number is automatically assigned to the level name when the level name is declared in the Dimensions - Dimension Page.
Attribute Name	The available values displayed in the Attribute Name column can be added to or changed in Attribute Names - Attribute Name page. This field is only available of the Role is set to Attribute .

Now that the high-level rollup information is created, you need to map each of the fields in the query-based rollup inputs to a role.

To add Dim Input Field (dimension input field) maps:

1. Click the Field Map button to define the fields you want to map from the input to the cube.
The Dim Input Field page appears with the source query fields already completed.
2. In the Input Field Map section, click the drop-down buttons at the Source Field, Role, Level Name, Level Num, and Attribute Name columns to specify fields, roles, levels, and attributes in the cube dimension.
3. Click the OK button to save your changes, or the Cancel button to exit the page without saving changes.

Designing Cube Definitions

In the first step you created the cube outline in Designing a Cube . The second step is to define the cube using the Cube Definitions - Cube Definition Page. Then specify the source of the data using the Cube Definitions - Cube Inputs page.

Cube Definitions - Cube Definition Page

Usage	Use this page group to map data sources to any dimension you've defined using the Dimension page, and to select which dimensions are used in the cube.
Object Name	CUBE_DEF Component
Navigation	Design, Cube Definition

Cube Definition **Cube Inputs**

Cube Def ID: EMPLOYMENT

Description:

Long Description:

Components					View All	First	1 of 4	Last
	Role	Dimension/Measure/Attribute		Dim Type	Status			
1	Dimension	DEPARTMENT		Standard	Mapped			
2	Dimension	EMPL_STATUS		Standard	Mapped			
3	Measure	HEADCOUNT			Mapped			
4	Dimension	JOBCODE		Standard	Mapped			

Save Return to Search Add Update/Display

[Cube Definition](#) | [Cube Inputs](#)

Design - Cube Definition page

Description

Enter the **Description** you want to associate with the cube. These values aren't transferred to the resulting cube definition; they are used for prompting within Cube Manager only.

Long Description

Enter the **Long Description** you want to associate with the cube. These values aren't transferred to the resulting cube definition; they are used for prompting within Cube Manager only.

Role


This field is required. Select from the following options:

Cube Attribute: Select if the component is an attribute. A cube attribute is a field that can be included in the fact table that can be used for external processing (not by the ROLAP engine). Our bookmark field on the fact table could be an example of a cube attribute. The bookmark cannot be used in analysis, but cube manager uses it to manage facts. For example, you can define fields that can exist on the fact table for processes that may read that table for subsequently.

Dimension: Select if the component is a dimension.

Measure: Select this option if the component is a measure or fact.

Dimension/Measure/Attribute

Use the  Lookup button to search for a valid Dimension, Measure, or Attribute.

 **Open button**

Available only if Dimension is selected in the **Dimension/Measure/Attribute** field. Displays the Cube Input Field page.

Dim Type

This field is read-only. Information is populated for dimensions and obtained from **Dim Type** on the Dimensions - Dimension Page.

Status

This field is read-only. The mapping information is obtained from Dim Input Field Page.



Click the **Add** button to add a row to input an additional component.



Click the **Remove** button to remove a row, deleting the component.

After creating data source queries and defining the dimensions, you must define a cube definition by mapping fields from the data source queries to the dimensions you've defined. For each dimension, you must specify which field in the data source query is the source field for the dimension.

To design a cube definition:

1. Design your cube dimension.



For more information, see Designing a Cube.

2. Select Design, Cube Definition, Cube Def.

Choose whether to Add a new cube definition or to Update/Display an existing one.

3. Enter a Cube Definition ID and click the OK button.

You need to provide a unique name for each cube definition you define. After you enter a Cube Def ID, the Cube Definitions page group appears.

4. Enter a Description and a Long Description for the cube definition.
5. Define the Components of the cube definition.

This is a critical step in which you define the dimensions, measures, and attributes that Cube Manager uses to pull data into the cube you are defining. The Role column shows the component type (Dimension, Measure, or Attribute), and the Dimension/Measure/Attribute column shows the name of the component. Once you define a component, its dimension and rollup types appear, as well as the mapped status.

6. Specify a dimension or measure to be used in the cube.

Because you can reuse the same dimension among various cubes, you are given an opportunity to rename dimensions for use with this particular cube definition. By default, the label you enter here corresponds with the name of the dimension you've specified and is used as the dimension's alias/label in the resulting cube. You might want to put the word "ALL" in front of the dimension name—ALL PRODUCTS, ALL REGIONS, and so on.

The number beside the Role determines in what order the dimensions and measures appear to the user in the OLAP tool. When adding dimensions and measures, this number increments automatically.

7. Add additional dimensions/measures.

For each additional dimension or measure, repeat steps 5 and 6.

8. Click the Save button on the toolbar to save your cube definition and all of the components you have added.

It is important to save your new cube definition at this stage, so that you can effectively add cube inputs.

Cube Definitions - Cube Inputs page

Usage	Use this page to add cube inputs to your cube definition.
Object Name	CUBE_DEF_INPUTS
Navigation	Design, Cube Definitions, Cube Inputs

Cube Definition | **Cube Inputs**

Cube Def ID: EMPLOYMENT

Description: Headcount of each department.

Long Description: This is an analysis model built to analyze headcount of different job codes and employment status for different departments.

Inputs		
Input Name	Field Map	Mapped
1 OLAP_EMP_DATA_QRY	Field Map	<input checked="" type="checkbox"/> + -

Save Return to Search Add Update/Display

[Cube Definition](#) | [Cube Inputs](#)

Design - Cube Inputs page

Long Description Text up to 30 characters that describes what you are defining.

Input Name Enter the name of the source of the fact data.


Field Map Click to view how each field in the fact source data is mapped to a dimension.

The cube definition delineates which dimensions and values are to be grouped together in a cube. The next step in the process is to actually create the cube definition.

To design cube inputs:

1. Click the Cube Inputs page on the page group or select Design, Cube Definitions, Cube Inputs from the Cube Manager menu.

The Cube Inputs page appears.

2. Add inputs by selecting from the list of valid values. If you need to add another Input Name, press  to insert a row.

Any number of input sources may be specified. However, all of them must be mapped completely in order to save the cube definition.

Cube Input Field

Usage	Map the fields of the source query to the appropriate cube components, in order to associate the data from the source query to a dimension. The Cube Inputs are the source values for the facts to be created in the
-------	--

	OLAP database.
Object Name	CUBE_INPUT_FLD
Navigation	Click the Field Map button in Cube Definitions - Cube Definition Page.

Cube Input Field

Cube Def ID: EMPLOYMENT

Description: Headcount of each department.

Input Name: OLAP_EMP_DATA_QRY

Input Field Map View All First 1-4 of 4 Last

	*Query Fld Name	Dimension/Measure/Attribute	Role
1	Department ID	DEPARTMENT	Dimension
2	HeadCount	HEADCOUNT	Measure
3	Job Code	JOB CODE	Dimension
4	Status	EMPL_STATUS	Dimension

OK Cancel

Cube Input Field page


Source Field

This field is populated from the Cube Definitions - Cube Inputs page.

Dimension/Measure/Attribute

Use this field to specify which dimension measure or attribute to which the source field will be mapped. You can check to see whether a field has been mapped to a dimension, measure, or attribute by looking in the **Role** column.

To design cube input field maps:

1. Click the Field Map button to map specific fields in the query.
The Cube Input Field page appears.
2. Press  to add rows, and select from the fields in Valid Values, as necessary.
3. In the Dimension/Measure/Attribute column, specify which dimension measure or attribute to which the corresponding source field will be mapped.
4. Click the OK button to save your changes or the Cancel button to exit without saving.
5. After you click the OK button, the Mapped check box on the Cube Inputs page is checked automatically. This enables you to come back in the future and easily see whether fields have been mapped for this cube input.

Designing Attribute Names

You can specify member attributes yourself by using the Attribute Name page. This feature enables you to specify user-defined attributes for members. Users can set these attributes in Essbase, Start Schema, and PS/ROLAP platforms, too. You can then use calculation (calc) scripts to search for and manipulate members having particular user-defined attribute values. For example, they have been extensively used for the Peoplesoft Workbenches.



For more information about Attributes and general attributes, see Attribute Queries.

Attribute Names - Attribute Name page

Usage	Specify user-defined attributes for members
Object Name	OLAP_ATTRIB_NAM
Navigation	Design, Attribute Names

Attribute Name
View All First 1-9 of 18 Last

Attribute Number	Attribute Name	Created By	
10	Label	System	+ -
11	Short Label	System	+ -
12	Alias	System	+ -
13	Long Description	System	+ -
14	Flip Sign	System	+ -
15	Unary	System	+ -
16	Time Balance	System	+ -
17	Expense Item	System	+ -
18	Shared	System	+ -

Save

Design - Attribute Name page

Attribute Number

If you are dealing with large numbers of attributes, this feature helps you group the attributes into more manageable categories. For example, you might want to list Human Resource attributes in the 100-300 range, while Finance attributes could be in the 400-600 range. This number is not used anywhere else.

Attribute Name

Enter a descriptive name for the attribute.

Attribute Number

If you are dealing with large numbers of attributes, this feature helps you group the attributes into more manageable categories. For example, you might want to list Human Resource attributes in the 100-300 range, while Finance attributes could be in the 400-600 range. This number is not used anywhere else.

Attribute Name

Enter a descriptive name for the attribute.



Click the **Add** button to add a row to input an additional attribute name.



Click the **Remove** button to remove a row, deleting the attribute.

CHAPTER 4

Building Cubes

Cube Manager provides a powerful template feature that lets you set up Cube Instance Definitions for use and reuse when building and updating your cubes. Having a Cube Instance Definition simplifies your work in managing updates to the cube structure, data loads and reloads, and setting platform-specific options when you create the cube. Handling such actions without the benefit of Cube Manager—and the Cube Instance Definition feature in particular—would require tedious and error-prone steps to build or update a cube.

Each Cube Instance Definition is associated with a single physical cube. You typically have one Cube Instance Definition for each cube, but you might also find it useful to have more than one.

Everything up to this point in creating a cube is done in a more or less platform-generic way that does not require you to specify the platform (PowerPlay, Essbase, Star Schema, or PS/ROLAP) that you ultimately use. Thus, if you decide to switch platforms, the majority of your cube design remains unchanged. However, when building the Cube Instance Definition, you must finally designate the specific platform for which you create the cube.

Building Cube Instances

The definition of a cube instance creates the link between the platform-independent definition of the cube and the actual physical storage mechanism of the cube. In essence, you tell Cube Manager where to create the cube that you have defined.

Cube Instances - Cube Inst Def page

Usage	This page is where you define cube settings (such as platform), and select dimensions, measures, and attributes to be included as cube instance definition components.
Object Name	ANALYSIS_DB
Navigation	Build, Cube Instances, Cube Inst Def

Cube Inst Def | **Aggregates**

Cube Instance ID: HEADCOUNT

Description: Cube template

Long Description: Headcount cube based on the department security tree

Platform: Cognos PowerPlay [Platform Options](#)

Based On: ☒ Cube Definition ☐ Cube Instance
EMPLOYED

Included Components First 1-4 of 4 Last

Dimension ID	Role	Dim Type	Status	*Sparsity
4 HEAD COUNT	Measure		Mapped	Default
1 DEPTTREE	Dimension	Standard	Mapped	Dense
2 JOBCODE	Dimension	Standard	Mapped	Dense
3 EMPL_STATUS	Dimension	Standard	Mapped	Sparse

Save Return to Search Add Update/Display

[Cube Inst Def | Aggregates](#)

Cube Instances - Cube Inst Def (cube instance definition) : Dimension tab

Cube Instance ID

This is a generic name to identify the Cube Instance.

Description

This description is used for prompting within Cube Manager only, and is not passed along to the resulting cube.

Long Description

This description is used for prompting within Cube Manager only, and is not passed along to the resulting cube.

Platform

Choose one of the following:

Hyperion Essbase

Cognos PowerPlay

PS/ROLAP

Generic Star Schema

Platform Options

See Platform Options for more details.

Based On

Cube Definition: If you select a **Cube Definition** source, the value refers to the **Cube Definition ID** you specified when designing your cube definition.

Cube Instance: If you select a **Cube Instance** source, the value refers to a physical star schema that was created based on a particular cube definition. Once again, the value is the **Cube Definition ID** used to create the actual star schema. This can be used to create smaller data marts based on a previously created star schema.



Click the **Open** button to open the selected **Cube Definition**.

Dimension ID

Name of the Dimension or Measure to be included in the cube.

Role

Defines the role of the dimension, measure, or attribute.

Dim Type

The dimension type is specified on the Dimensions - Dimension Page.

Status

Lets you check that your dimension has been mapped. If your dimension has not been mapped, you must map it before the dimension can be used in a cube.

Sparsity

This is specific to Essbase. Refer to the Essbase option page. (See the Essbase Advanced Options section for more details.)

Cube Instances - Cube Inst Def page : Filter tab

Usage	This tab is used to add filters and create new labels for your cube dimensions.
Object Name	ANALYSIS_DB
Navigation	Build, Cube Instances, Cube Inst Def
Prerequisites	Dimension must be added in the Dimension tab, before you can create a filter for it in the Filter tab.

Cube Inst Def | **Aggregates**

Cube Instance ID: HEADCOUNT
Description: Cube template
Long Description: Headcount cube based on the department security tree

Platform: Cognos PowerPlay [Platform Options](#)

Based On: ☒ Cube Definition ☐ Cube Instance
 EMPLOYED

Included Components First 1-4 of 4 Last

Dimension ID	Updt Filter	Filter	Label		
HEAD COUNT	Updt Filter	<input type="checkbox"/>	Head Count	+	-
DEPTTREE	Updt Filter	<input type="checkbox"/>	Department	+	-
JOBCODE	Updt Filter	<input type="checkbox"/>	Job Code	+	-
EMPL_STATUS	Updt Filter	<input type="checkbox"/>	Status	+	-

[Save](#) [Return to Search](#) [Add](#) [Update/Display](#)

[Cube Inst Def | Aggregates](#)

Cube Instances - Cube Inst Def (cube instance definition) : Filter tab

Updt Filter (update filter)

Click this button to create the Cube Filter Criteria .

Filter

This check box indicates if a filter has been created for this dimension

Label

This is how the dimension label is displayed in the results.



Click the **Add** button to add a row to input an additional level name.



Click the **Remove** button to remove a row, deleting the level name.

Cube Filter Criteria

Usage	Use this page to create the cube filter for use on the Filters tab in Cube Instances - Cube Inst Def page : Filter tab.
Object Name	CUBE_FILTER
Navigation	Select Build, Cube Instances, Cube Inst Def, then click the Updt Filter button in the Filter tab on the Cube Instance Def page.

Cube Filter Criteria

Cube Inst ID: EMPLOYMENT
Description: Department Headcount
Dimension ID: DEPARTMENT

Member List		View All	First	1 of 1	Last
	Member Value				
1	<input type="text"/>				

OK Cancel

Cube Filter Criteria page

Member Value

Enter the value for the criteria. If your source dimension has more information than you wish to report, you can create a filter to eliminate information you don't want. For example, if you have data from the U.S., UK, and Japan, but only want to view the U.S. and UK data, create a Japan filter to eliminate this data from being created in the cube dimension.

To build a cube instance definition:

1. First design your cube definition.

2. Select Build, Cube Instances, Cube Inst Def.

Choose whether to Add a new Cube Instance Definition or to Update/Display an existing one.

3. Enter a Cube Instance ID and click the OK button.

You need to provide a unique name for each Cube Instance Definition you define. After you enter a Cube Instance ID, the Cube Instances page group appears.

4. Select the Cube Inst Def page to define the cube platform and included components.

The Cube Inst Def page appears.

5. Enter a Description and Long Description.

6. Specify the Platform.

Once you choose the appropriate platform type, you can then click the Platform Options button to specify platform-specific build options.

7. Specify the criteria for this cube instance definition: either a cube definition or an existing star-schema-based cube instance.



For more information, see Designing Cube Definitions.

8. Remove any Dimensions or Measures that are not to be created in this Cube Instance.

By default, Cube Manager adds all of the dimensions and measures to the Cube Instance Definition. However, you might not need all of the dimensions in this definition. You can remove any unwanted dimensions or measures by selecting the element in the grid and clicking the Remove button.

9. Click the Platform Options button to display platform-specific options based on which platform you selected from the Type drop-down list.

What page you see depends on which platform you chose.

Platform Options

There are four platform options:

- Cognos PowerPlay Options
- Generic Star Schema Options
- PS/ROLAP Options
- Hyperion Essbase Options

Hyperion Essbase Options

Usage	Enables you to define cube settings specific to Essbase.
Object Name	ANALYSIS_DB_ESS
Navigation	Cube Manager, Build, Cube Instances, Cube Inst Def Click the Platform Options button.
Prerequisites	Hyperion Essbase must be selected in the Platform Type field on the Cube Instances - Cube Inst Def page.

Essbase Advanced Options

Cube Instance ID: HEADCOUNT

Server: localhost User Name: admin

Application: Employ Password: *****

Database: Count

User Access: Design

Meta-Data Update Action
Replace All

Data Update Action
Delete Upper Level D

Rate Cube Action
No Action

Advanced Options

☒ Run Default Calc
☐ Auto Config Sparsity

View All First 1-2 of 3 Last

*Language Code

1	(Invalid Value)	+	-
2	French	+	-

Rate Cube Mapping

Rate Cube Name: HEADC_MC

Query Name:

Time:

Account:

Country:

Type:

Rate:

Default Currency Name: US\$

Default Currency Category:

OK Cancel

Hyperion Essbase Advanced Options page

Server Enter the **Server** name.

Application In Essbase, an application is a group of one or more databases. Enter the application name.

Database In Essbase, a database is, essentially, an OLAP (online analytical process) cube. Enter the **Database** name.

User Access You can set the **User Access** to *Calculate*, *Design*, *Read-Only*, *Read-Write*, or *None*. This setting sets the default user security on the resulting cube, if explicit security has not been granted in Essbase. It applies to everyone but the system administrator (to whom the **User Name** and **Password** options apply), so you typically set at least *Read-Only* access.

Choose from one of the following options:

Meta-Data Update Action The **Meta-Data Update Action** enables you to specify the update action that should occur in Essbase when a cube structure has been redesigned in Cube Manager. The options are as follows:

Replace All: This option is the default. Performing a meta-data update with the update action set to ***Replace All*** completely refreshes the cube's structure. Any members defined in PeopleSoft that don't exist in the cube are added to the cube. Any members in the cube that are no longer defined in PeopleSoft are deleted from the cube, and any associated data is lost. Members moved in PeopleSoft, from one node to another, for example, are moved in the cube with any associated data preserved.

Incremental Update: Performing a metadata update with the update action set to ***Incremental Update*** merges the metadata defined in your cube definition with the existing meta-data of the cube. No members are deleted.

Data Load Action

The **Data Load Action** enables you to specify the update action that should occur in Essbase when a cube's data has been changed within a PeopleSoft application. You have three choices:

Delete Upper Level Data: This option deletes all aggregate data.

Delete Non-Input Data: This option deletes any data that hasn't been loaded into a cube by the build process—for example, data you've entered manually in Essbase or data that results from a calculation in Essbase.

Incremental Update: This action preserves existing data and enables new data to be added to the existing data for both multiple hierarchy and single hierarchy dimensions.

Rate Cube Action

If the cube definition contains the appropriate dimensions (account, time, and country) you may create a Rate cube. This setting enables you to specify the following Rate cube creation options:

Create All: This action regenerates the currency cube definition from the cube defined in Cube Manager. Then, it populates the resulting currency rate cube with the data defined in the **Rate Cube Mapping** group box, if any mapping exists.

Rate Update Only: This action replaces the existing data in the Currency Rate cube without regenerating the outline.

No Action: Does not perform any action on the Currency Rate cube.

Run Default Calc (run default calculation)

If you select the **Run Default Calc** option, then the default calculation script in Essbase is run upon creation of the cube. This means that Essbase aggregates the tree in the given rollups, but doesn't do anything else. You might instead decide to create a custom calculation script, in which case you might want to turn off this option.


Auto Config Sparsity (automatically configure sparsity)

If you clear the Auto Config Sparsity check box, you can manually set the sparsity levels on the Cube Instance Definition page. Otherwise, Essbase automatically sets sparsity levels. After you've cleared this check box, you can set the sparsity of each dimension in the template to either *Sparse* or *Dense*.

Note. You may experience errors due to lack of memory if you have set every dimension to Dense.

Language Code

You can choose to build an Essbase cube in multiple languages. When the cube is then viewed in Essbase, it isn't displayed in the user's preferred language. If the cube is not available in the user's language, the default language (developer's language) is used.

To add one additional language, select a language from the drop-down list. To add additional languages, click the  **Add** button to add more rows and select a language for each.

 **Delete** button to remove a selected language.

Rate Cube Name

This section enables you to supply a Rate Cube mapping. Cube Manager generates the actual Currency Rate Cube by default.

Specify the **Cube Name** to be created by Cube Manager and mapped to the defined cube.

Query Name

Click the Lookup button to select the query that has your currency rate definition.

Time

Click the Lookup button to select the field name for time from the Time Dimension.

Account

Click the Lookup button to select the field name for currency category from the Account Dimension.

Country

Click the Lookup button to select the field name for currency name from the Country Dimension.

Type	Click the Lookup button to select the field name for currency type.
Rate	Click the Lookup button to select the field name for currency rate multiplier.
Default Currency Name	Enter the default currency name. If you do not specify a currency name for a member in your Country Dimension, the default currency name will apply to the member.
Default Currency Category	Enter the default currency category. If you do not specify a currency category for a member in your Account Dimension, the default currency category will apply to the member.

The following procedure shows both the optional and required steps that enable Cube Manager to build and tune an Essbase target cube.

To define Essbase cube instance definition options:

1. Select Hyperion Essbase from the Type drop-down list in the Platform group box and click the Platform Options button.

The Essbase Advanced Options page appears.

2. Specify the Essbase Server name, Application, and Database. (Required)
3. Enter the Essbase username and password. (Required)
4. Select the default user access to the cube. (Required)
5. Choose a Meta-Data Update Action. (Required)
6. Choose a Data Load Action. (Required)



Both Meta-Data Update Action and Data Load Action refer to the metadata and data action options on the Build Cube page, and only apply to updating existing cubes.



For more information, see Designing Cubes.

7. Choose a Rate Cube Action. (Applies to Rate Cubes only.)
8. Choose whether to run the default calculation script after building the cube.
9. Choose whether you want to override Essbase's automatic sparsity configuration.



For more information about Density and Sparsity settings, see your Essbase documentation.



Dimensions marked as sparse might take up significantly more space in the resulting cube than had they been marked as dense. If you mark every dimension as sparse, you could end up with an enormous cube that is not practical or even possible to store on even the most powerful servers. That's why Time and Account dimensions are defaulted to Dense. It's also possible to change these settings in Essbase after the creation of your cube. Whether you change the sparsity settings from Cube Manager or from Essbase, take care when making your adjustments and refer to the Essbase documentation for more information about Dense versus Sparse dimensions.



For more information, see your Essbase documentation.

10. Choose any additional languages for which you want to build the cube from the Language Code drop-down list.



The cube is automatically built in your own language. You do not have to select your own language here.

11. Define the Rate Cube Name.

12. Enter the Query Name to be used as a data source for the Currency Rate cube.

13. Select the Time, Account and Country dimensions to be used for the currency rate cube.

14. Select the field names for the currency type and currency rate multiplier.

15. Enter the Default Currency Name & Default Currency Category.

16. Click the OK button.

Multi-Currency Rate Cubes

Essbase supports multi-currency cube. Essbase multi-currency cube contains two databases, the regular main database and the currency rate database. The currency rate database is created out of the main database. In order to generate the currency rate database from the main database, the main database **MUST** have the appropriate dimensions: Accounts Dimension, Time Dimension, and Country Dimension.

- **Accounts Dimension** contains items that you want to measure, such as profit and inventory and makes Essbase built-in accounting functionality available.

- **Time Dimension** defines the time period for which you report and update data.
- **Country Dimension** contains data about where business activities take place.

You can also specify **Currency Partition Dimension** (Cube Manager also calls it Currency Dimension.) in the main regular database for Essbase multi-currency cube. The purpose of **Currency Partition Dimension** is for separating local currency members from a base currency defined in the application. If the base currency for analysis is US dollars, the local currency members contain values that are based on the currency type of the region. Currency Partition Dimension is used only in the main regular database and is used for currency conversion using Essbase script or other tools.

To Create a Multi-Currency Cube in Essbase:

The following steps outline the things you specify for the **Accounts Dimension** and the **Country Dimension** specifically for Essbase multi-currency cube.

1. Specify the **Currency Category** in your Accounts Dimension, for example P&L (for profit and loss).

To do that, you need create an attribute query and specify the query as an input source in the Rollup Inputs tab. Then map the fields in the query as you do for attribute query.

Dim Input Field

Dimension ID: P&L

Description: P&L 1

Input Name: CHY_TR

OK

Cancel

Input Field Map

View All

First

1-2 of 2

Last

	*Query Fld Name	*Role	Level Name	Level Num	Attribute Name	
1	ID	Member				+ -
2	COUNT	Attribute			Currency Category	+ -

Dimension Input Field - showing currency category for Count attribute.

2. Specify **Currency Name** in your Country Dimension, for example US\$.

To do that, you need create an attribute query and specify the query as an input source in the Rollup Inputs tab. Then map the fields in the query as you do for attribute query.

Cognos PowerPlay Options

Usage	Enables you to define cube settings specific to PowerPlay.
Object Name	ANALYSIS_DB_PPL
Navigation	Cube Manager, Build, Cube Instances, Cube Inst Def Click the Platform Options button.
Prerequisites	Cognos PowerPlay must be selected in the platform type field.

Access Requirements	Enterprise Server and Cube Manager must both be installed on the same machine, so that Cube Manager can build and add cubes to the Enterprise Server.
---------------------	---

PowerPlay Options

Cube Instance ID: HEADCOUNT

PowerCube Path/File (.MDC): C:\USER\HEADCOUNT.MDC

Model Path/File (.MDL): C:\USER\HEADCOUNT.MDL

Data Working Directory: C:\TEMP

Language Override: Always Include ☐

Earliest Date

Use As Of Date ☐

Use Current Date ☐

Use Specific Date ☐

Latest Date

Use As Of Date ☐

Use Current Date ☐

Use Specific Date ☐

Enterprise Server Options

Enterprise Server ☐

Service Port:

Logon Password:

OK

Cancel

Cognos PowerPlay Options page

PowerCube Path/File (.mdc)

PowerPlay requires both a PowerCube file and a model file. The PowerCube file has an .MDC filename extension and equates to a cube. Specify the path and filename in the **PowerCube Path/File (.mdc)** field.

Model Path/File (.MDL)

Specify the path and filename for the **Model Path/File (.MDL)** field. The model file has an .MDL filename extension and represents the cube structure.

Data Working Directory

Specify the location where the .DAT file is written.

Language Override

If you are implementing the PowerCube in a non-English language, you might need to select from the available languages in the **Language Override** drop-down list.

Always Include

Always Include retains the category in the model and includes it in cubes, even if it, or any of its descendants, fails to appear in the data source. The category is also included if any of its descendants are included and if the category has not been excluded, summarized, or cloaked in a dimension view. If necessary, the Transformer includes ancestors of the category, regardless of their Inclusion settings.

Note. **Always Include** is the default Inclusion setting when you create a time dimension. If you exclude categories, special categories that use relative time concepts (such as Last Month) don't work properly.

PEOPLESFT PROPRIETARY AND CONFIDENTIAL

BUILDING CUBES 4-13

Enterprise Server

Select the **Enterprise Server** option if you want your resulting cube to be registered on the PowerPlay Enterprise Server automatically by Cube Manager.

If you do not select the Enterprise Server option, Cube Manager creates the cube using the Transformer just as it would if you were using PowerPlay version 5.2.

Earliest Date

These fields are optional and are used when designing and creating Time dimensions for PowerPlay. If not specified these will default to the specific time of 1901-01-01.

Latest Date

These fields are optional and are used when designing and creating Time dimensions for PowerPlay. If not specified these will default to the specific time of 2100-12-31.

The following steps show both the optional and required options that enable Cube Manager to build a Cognos PowerPlay cube.



Cube Manager and the Enterprise Server must be set up to run on the same machine.



For more information about how the different versions of PowerPlay and Cube Manager work together, see PowerPlay.

To define PowerPlay cube instance definition options:

1. Select Cognos PowerPlay from the Type drop-down list in the Platform group box and click the Platform Options button.

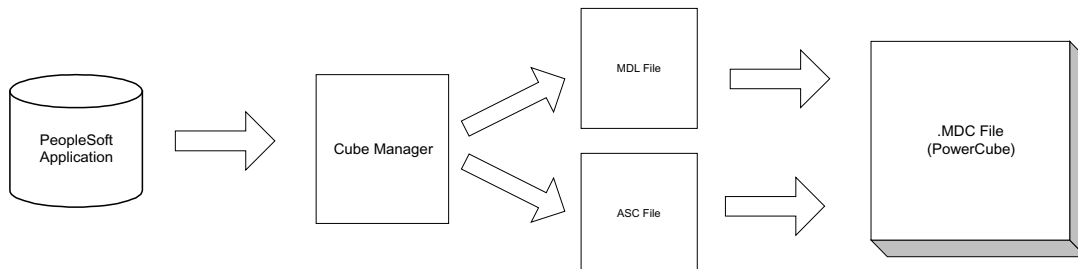
The PowerPlay Options page appears.



PowerPlay refers to a multidimensional database as a PowerCube. Thus, we use this term in place of cube when referring specifically to the resulting cube on PowerPlay.

2. Specify the file names to be created when the cube is built.
3. Enter the Data Working Directory.

When building a PowerCube, Cube Manager sends output from each query used in the PowerCube to a flat data file before sending this data on to the PowerCube (.mdc) file. This data file has a .DAT filename, and the Data Working Directory you specify is where this file is written.



The MDL file and .DAT files merge to create the PowerCube (.MDC file)



The data file created in the data working directory is not automatically deleted or overwritten when Cube Manager builds a PowerCube. You must delete these files manually.

4. If necessary, select a Language Override.
5. Click the OK button if you are not using the Enterprise Server. Otherwise, continue to step 6.
6. Select Enterprise Server if you want to use the Enterprise Server functionality.

The Cube Instance Definitions page appears.

The Service Port ID defaults to 8010. This should be the correct port for most machines. PowerPlay Enterprise Server is set to communicate on port number 8010 for both Windows and Web connections.



Cube Manager assumes that the Enterprise Server is running locally. Cube manager retrieves the local machine name from the system to connect to the Enterprise Server.

Cube manager uses Ppadmtool.exe to talk to the locally running Enterprise Server to add or remove cubes. Cube manager first tries to remove the cube from the Enterprise Server. If the cube already exists, it removes the cube from the Enterprise Server. This prevents other users from trying to access the cube through the Enterprise Server while Cube Manager is actually creating it. After removing the cube from the Enterprise Server, Cube Manager uses the Transformer to generate the .mdc file.

If the cube is not listed in Enterprise Server at the moment, Cube Manager uses the Transformer to generate the .mdc file.

After Cube Manager creates the .mdc file, Cube Manager uses the Ppadmtool.exe to add the .mdc file to the Enterprise Server. Once the cube has been added to the Enterprise Server, any user connected to it from a local or remote machine can view the newly added cube.

If you do not select the Enterprise Server option on the PowerPlay Options page, Cube Manager creates the cube using the Transformer, just as it would if you were using PowerPlay version 5.2.



PowerPlay does not allow duplicate details in the same hierarchy, but it does allow duplicate details across hierarchies for the same dimension.

Generic Star Schema Options

Usage	These options help to identify the field and record objects generated by Cube Manager.
Object Name	ANALYSIS_DB_SS
Navigation	Cube Manager, Build, Cube Instances, Cube Inst Def Click the Platform Options button.
Prerequisites	Generic Star Schema must be selected in the Platform Type field.

Generic Star Schema Platform Options page

PeopleTools Object Name Prefix

The prefix is applied to each record and field definition created by Cube Manager. (The prefix allows you some manner of control over the record and field names.) The default is DW_

To define Star Schema cube instance definition options:

1. Select Generic Star Schema from the Type drop-down list, and click the Platform Options button.

The Star Schema Options page appears.

2. Enter the prefix you want to use with the generated objects.

PS/ROLAP Options

Usage	These options help to identify the star schema metadata to the PS/ROLAP engine. Also, as with all star schema target databases, you can specify the field and record object prefixes.
Object Name	ANALYSIS_DB_IAS

Navigation	Cube Manager, Build, Cube Instances, Cube Inst Def Click the Platform Options button.
Prerequisites	PS/ROLAP must be selected in the Platform Type field.

PS/ROLAP Options page

Catalog	This value specifies the catalog in which the PS/ROLAP objects are defined. All objects defined in a single catalog can be used in a ROLAP report.
Datamart	A datamart is a basic grouping of related OLAP objects. The data mart grouping helps break objects into a smaller, more manageable set.
Server	This is the host machine that has the PS/ROLAP services and files.
Port	This is the port for the PS/ROLAP services.
User Name	These must be valid UNIX user IDs for PS/ROLAP.
Logon Password	These must be valid UNIX user IDs for PS/ROLAP.
Prefix	This is the prefix that is appended to the ROLAP tables in the database IE.

To define PS/ROLAP cube instance definition options:

1. Select Sterling PS/ROLAP from the Type drop-down list and click the Platform Options button.

The PS/ROLAP Options page appears.

2. Enter the Catalog and Datamart in which to create the metadata.

These values help to identify the groupings of metadata to the PS/ROLAP engine.

3. Enter the PS/ROLAP username and password.
4. Enter the prefix you want for the star schema objects.

- Click the OK button.

Cube Instances - Aggregates page

Usage	This page has not been implemented for Peopletools 8.12.
Object Name	CUBE_AGGREGATE
Navigation	Build, Cube Instances, Aggregates

Cube Inst ID: HEADCOUNT
 Description: Cube template

Aggregate ID: 1

*Dimension ID	*Rollup	Rollup Name	Level Name	Levels From Top
1				

Save Return to Search Add Update/Display

Cube Instances - Aggregates page

Building Cubes

When you're ready to build a cube, you need a place to specify the runtime parameters to be passed during the creation or update of the data cube using Process Scheduler. The Build Cube page in Cube Manager provides an example of how to pass this information from a PeopleSoft page to the resulting cube using a Process Scheduler run control. But most likely, you're building cubes using pages that were either supplied with your PeopleSoft applications or that you've built yourself.

You have the option to customize the way in which a cube is built by creating your own run control pages and process definitions using the Cube Builder process type. Certain PeopleSoft products deliver their own pages for building cubes that leverage product-specific metadata.



If you are building a Hyperion Essbase cube, and it contains Time, Account, and Country dimension types, Cube Manager automatically builds a currency cube, if one is defined. The calculation for the currency cube always defaults to *Division*.

Build Cubes - Build Cubes page

Usage	The Build Cubes page is where you define the cube build specifications and bind variables.
Object Name	RUN_MAKE_ANL_DB
Navigation	Build, Build Cubes, Build Cube

Build Cubes - Build Cube page

Cube Instance ID

Use the Lookup button to search for valid Cube Instance definitions.



To view the cube instance definition, click the Open Cube Instance Definition button. The Cube Instances - Cube Inst Def page displays with the instance definition you specify.

Meta-Data

The metadata options are as follows:

Create: Create the cube. If the cube already exists, then Cube Manager recreates it, overwriting any dimensions and data that previously existed.

Update: This option in PS/ROLAP and Generic Star updates the metadata. In Essbase, this option is linked to the Meta-Data Update Action option on the Essbase Cube Instance Definition page. It updates the structure of the cube according to the setting for the Meta-Data Update Action. For PowerPlay, this has the same effect as **Create**.

None: Don't make any changes to the structure of the cube or its individual dimensions (you may want to update the data only).

Data

You can allow the user to define the data action to take place when updating the cube:

Create: Completely reload the data, overwriting any existing data.

Update: Update the existing data in the cube. For Essbase, this option is linked to the Data Load Action on the Essbase Cube Instance Definition page.

None: Don't make any changes to the data in the cube. (You may want to update the structure only.)

Aggregation

Select from the following options:

Create: Completely reload the aggregates, overwriting any existing data.

None: Don't make any changes to the aggregates in the cube. (You may want to update the structure only.)

Business Unit

The **Business Unit** enables you to get the appropriate set IDs for tree metadata. Leave the **Business Unit** blank if your trees do not use set IDs.

As of Date

The **As of Date** enables you to get the appropriate effective dates for tree metadata. Type in the date, or use the calendar button to select a date.

Bookmark

BookMarks identify a set of facts in the Generic Star Schema cube. They are used for determining whether facts are being loaded for the first time or whether they are being re-loaded. The first time you create a build cube definition, the bookmark is set to **Initial**. You can also add a build script.

Script

The **Script** field refers to platform-specific post-build scripts, such as the MDL file for PowerPlay or any command line for Windows. This is a powerful feature that enables you to extend the capabilities of Cube Manager. For example, you could use this feature to specify an Esscmd script that sets up security. Then, rather than having to configure security manually every time a cube is built, you could write a script once and specify it in the **Script** field to execute it for each build.

Report Manager

This link shows you the report list in **Report Manager**. For more information about **Report Manager**, see Using Report Manager.

Process Monitor

This link shows you the Process list in Process Monitor. For more information about Process Monitor, see Using Process Monitor.

Run

When you click the Run button, the Process Scheduler Request page appears.

To build a cube:

1. First build the Cube Instance Definition.



For more information, see Building Cube Instances.

2. Select Build, Build Cube.

Choose whether to Add a new cube definition or to Update/Display an existing one.

3. Enter a Run Control ID and click the OK button.

After you enter a Run Control ID, the Build Cubes component appears.



For more information about creating custom run control pages, see the Process Scheduler documentation.

4. In the Cube Specification group box, select a Cube Instance ID.

5. Define the run control information in the Run Time Parameters group box.



If the default calculation script is defined to run in the Essbase Cube Instance Definition, the default calculation script runs first, then the command line is invoked to run a subsequent script.

6. Define the Meta-Data Action that should occur during the build.
7. Define the Data Action that you want to occur during the build.
8. Define the Aggregate Action that you want to occur during the build.
9. Set your Process Scheduler parameters and save the page.



When working in a 3-tier environment, you *must* build your cubes on a Process Scheduler NT server, not on the client.

Build Cubes - Bind Variables page

Usage	Use this page to define bind variables that limit the data that is used to build a cube by prompting users for information when they start the build process.
Object Name	CUBE_BUILD_BIND
Navigation	Build, Build Cube, Bind Variables

Build Cube Bind Variables

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

Cube Instance ID: EMPLOYMENT Platform: PowerPlay
Description: Department Headcount

Bind Variables View All First 1 of 1 Last

Description	Bind Value
1	

[Save](#) [Return to Search](#) [Add](#) [Update/Display](#)

[Build Cube](#) | [Bind Variables](#)

Build Cube- Bind Variables page

Bind Variables Description Enter a short description.

Bind Value Enter the value for the variable.

If you want to use dynamic queries, you have to create a query that joins the run control with the desired record, and joins the Operator ID and the Run Control ID.

Any Cube Manager query (dimension query, data source query, or attribute query) can be made dynamic by joining in runtime parameters. The parameter value must be stored in the database on a Cube Builder run control record, or on a record that can be joined to by the run control record. Then, in PeopleSoft Query, join in the run control record and set prompts for the operator ID and the run control ID. These prompts are filled in at runtime with the operator ID and run control ID of the run control used to launch the Cube Builder process. The **Description** and **Bind Value** on the sample Build Cube run control page provide one way to do this.

To add bind variables to a run control page:

1. Identify the run control record used to launch the Cube Builder process.

In the case of the sample Build Cube page delivered in the PTDMO database, this is MAKE_ANL_DB_REQ.

2. Identify the fields you want to use for parameters for runtime queries.

For example these may be From and To values for time periods.

3. Ensure that the fields identified in step 2 are available on the run control record, or that they are on a record that can be joined with the run control record.



For more information about designing records, see Creating Record Definitions.

4. Ensure that these same fields are available on a page, so that users can enter values for them.

If these fields exist on the run control record, then they most likely exist on the run control page, as well.



For more information about designing pages, see Creating Field Definitions.

5. In PeopleSoft Query, join the records that contain the fields mentioned above to the query that you want to make dynamic.

For example, assume that you want to bind in values for the From and To periods and that these fields exist on the run control record. In the data source query, you join the From and To periods on the run control record to the From and To fields in the main data record (the primary record in the query that contains the data you want to return).



For more information, see PeopleSoft Query.

6. In PeopleSoft Query, set up prompts for the operator ID (first) and the run control ID (second).

Instead of prompting for these values at runtime, Cube Manager automatically supplies the bind values that the user entered on the run control page. Be sure to set up the prompt for the operator ID followed by the prompt for the run control ID.

7. Open the Bind Variables page by selecting the Bind Variables page in the Build Cubes component, or select Build, Build Cube, Bind Variables and enter a Cube Instance ID.

The Bind Variables page appears.

8. Add variables and save your changes.

Enter a short text description and give the variable's value.

Example of Bind Variables Used in PeopleSoft General Ledger

General Ledger is just one of the PeopleSoft applications that provides a custom run control page. If you need to create a custom run control page, you might find it useful to follow the example that was set in this application.

The run control page in PeopleSoft General Ledger is named Build Ledger Cubes. It provides a way for users to enter bind values for the Ledger, Currency Code, From Year, To Year, From Period, and To Period. Other than these bind variables, the page is essentially the same as the sample Build Cube page delivered in the PeopleTools Demo database (PTDMO).

Usage	Use the Build Ledger Cubes page to run the background process that creates ledger cubes.
Object Name	GL_OLAP_LED_REQ
Navigation	Process Financial Information, Review Financial Information, Process, Build Ledger Cubes
Access Requirements	Enter a User ID and a Run Control ID.

Build Ledger Cubes

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

Process Request Parameters

*Cube Instance ID: [Open Cube Instance Definition](#)

*Business Unit: *As of Date:

Post-Build Script:

*Ledger: *Currency:

From Year: To Year:

From Period: To Period:

*Meta-Data Action:

*Data Action:

[Save](#) [Return to Search](#) [Next in List](#) [Previous in List](#) [Add](#) [Update/Display](#)

Process - Build Ledger Cubes page

Cube Instance ID	Enter the name of the cube template or select it from the menu.
Open Cube Instance Definition	Click this link to change any settings on the Cube Instances - Cube Inst Def page.
Business Unit	Select the setID for the tree you are using.
As of Date	Select the effective Date.
Post-Build Script	This could be an Essbase, PowerPlay, or PS/ROLAP script that runs on the cube after the system builds and populates it.
Ledger	Select the ledger you want to use when populating the cube.
Currency Code	Select the appropriate code, according to the currency type, to be displayed.
From Year	Enter the starting time range for the ledger data you want to appear in the cube.
To Year	Enter the ending time range for the ledger data you want to appear in the cube.
From Period	Enter the starting period range for the ledger data you want to appear in the cube.
To Period	Enter the ending period range for the ledger data you want to appear in the cube.
Meta-Data Action	Select from the following choices:

Create: If the cube already exists, then the Cube Manage recreates it, overwriting any dimensions and data that previously existed.

Update: This option is linked to the Meta-Data Update Action option on the Essbase Cube Instance Definition page. It updates the structure of the cube according to settings for Meta-Data Update Action. For PowerPlay, this has the same effect as **Create**.

None - Don't make any changes to the structure of the cube or its individual dimensions. (You may want to update the data only.)

Data Action

Select from the following options:

Create: Completely reload the data, overwriting any existing data.

Update: Update the existing data in the cube. For Essbase, this option is linked to the Data Load Action on the Essbase Cube Instance Definition page. This option does not apply to PowerPlay.

None: Not applicable in General Ledger.



For more information about how General Ledger uses a custom Cube Manager page to build cubes, see Defining the Cube.




The General Ledger run control page makes it easy for a user to enter bind values by limiting the choices available. The user knows exactly the fields for which he or she is supplying values, because those controls are supplied on the page: Ledger, Currency Code, From Year, To Year, From Period, and To Period. This approach, while beneficial for the user, requires more time and care on the part of the designer.

Keep in mind that it's possible to create a run control page that is essentially identical to the sample Build Cube page. That is, where the field for which the user supplies the bind value(s) is determined at runtime by the user. In the Description text box, the user enters the name of the field that requires a bind value, and then enters that value in the Bind Value text box. Additional Descriptions and Bind Values can be inserted as required.

Setting Up Process Scheduler

Part of the cube build process is setting up the Process Scheduler Manager page.



For more information about Process Scheduler Manager, see Understanding Process Types in the Process Scheduler Manager documentation.

Process Definition page on Process Definitions page

Usage	You use the Process Scheduler Manager to define a cube build process request to run on the system.
Object Name	PRCSDEFN
Navigation	Process Scheduler, Use, Process Definition

Process Definition

Process Definition Options

Override Options

Destination

Process Type:

Cube Builder

Name:

PSBLDHC

*Description:

Build Cube

Long Description:

*Priority:

Medium

☒ API Aware

☒ Log client request

☐ SQR Runtime

Process Scheduler - Use - Process Definition page

- Description

Enter descriptive text of up to 30 characters.
- Long Description

This field is optional. Enables you to enter additional detailed description.
- Priority

Priority defines the relative priority used by the Process Scheduler Server Agent to determine which process to initiate first, if multiple processes are queued to run on a given server.
- API Aware

Select this option, if this is an API-aware process. If this option is selected for any process that is not API-aware, Process Scheduler includes it in the concurrent task count. This can result in improper server load balancing. Selecting this option does not mean your process become API-aware. You still have to add API code to your process.

Log client request

Select this option, if you want to provide an audit trail for any processes requested to run on a client workstation. This causes the system to log the request to the Process Request table each time an operator runs the process on their client workstation. **Log client request** is selected by default for all API-aware processes, and logging is always performed for all server-based requests.

SQR Runtime

If you select this option, the system appends *SQT* to the process name in the parameter list on the Process Definitions Options page, and uses the SQT Working Directory specified in the Process Type Definition page.



For more information about Process Scheduler, see Defining Processes.

Process Scheduler Request Page

Usage	The Process Request page is the final page in the cube build process.
Object Name	PRCSRQSTD LG
Navigation	Cube Manager, Build, Build Cubes Click the Run button.

Process Scheduler Request

User ID: PTDMO Run Control ID: EMPLOYMENT

Server Name: Run Date: 07/28/2000

Recurrence: Run Time: 1:23:44PM

Time Zone: [Reset to Current Date/Time](#)

Select	Description	Process Name	Process Type	*Type	*Format
<input checked="" type="checkbox"/>	Build Cube	PSBLDHC	Cube Builder	(None)	NONE


[OK](#) [Cancel](#)

Process Scheduler Request page

Server Name

The name of the server on which you want the process to run. Select from the following choices:

	<i>PSNT:</i> a Microsoft NT server.
	<i>PSOS390:</i> an OS 390 server.
	<i>PSUNIX:</i> a UNIX server.
Recurrence	Select either <i>Daily Purge</i> or <i>M-F at 5pm.</i> This tells Process Scheduler the recurring time intervals for a process request to run.
Time Zone	Select the time zone in which your process runs. For instance, you could be in Eastern Standard Time (EST) and schedule a process to run in PST (Pacific Standard Time).
Run Date	The date you want the process to run.
Run Time	The time at which you want the process to run.
Reset to Current Date/Time	Sets the run Run Date and Run Time to the present date and time.
Select	If this check box is selected, the process is included in the build process.
Description	This helps to uniquely identify a process. You should be familiar enough with the processes that you run as part of your daily tasks to identify them by this description.
Process Name	The name of the process as it appears in the definition.
Process Type	The type of process, such as COBOL or Crystal.
Type	The type of output you want to generate for this job.
	<i>File:</i> This enables you to write the output to a file that appears in the Output Destination.
	<i>Printer:</i> This value resolves to the default printer defined for a workstation or a server. You can enter a custom printer location, if you have the appropriate security access.
	<i>Email:</i> If you want a report to be sent to a particular email list, you can enter the appropriate email address in the Output Destination edit box. This option is available for SQR, nVision, and Crystal.
	<i>Web:</i> Sends all output of the process to the report repository, including log and trace files. The format of the report is specified by the format list.
Format	Just as you have a few options for Output Type, there are even more options regarding your Output Format. There are a variety of possible output types, depending on what Process Type you have selected. The default output format for Crystal, SQR, and nVision is <i>HTML.</i>

Server Name	The name of the server on which you want the process to run. Select from the following choices: <i>PSNT:</i> a Microsoft NT server. <i>PSOS390:</i> an OS 390 server. <i>PSUNIX:</i> a UNIX server.
Recurrence	Select either <i>Daily Purge</i> or <i>M-F at 5pm.</i> This tells Process Scheduler the recurring time intervals for a process request to run.
Time Zone	Select the time zone in which your process runs. For instance, you could be in Eastern Standard Time (EST) and schedule a process to run in PST (Pacific Standard Time).
Run Date	The date you want the process to run.
Run Time	The time at which you want the process to run.
 Distribute to	This is the list of people who have the correct security to receive the output of this process.

For more information, see Process Scheduler Manager, “Process Scheduler Report Distribution”.



For more information about Process Scheduler, see Process Scheduler for the End User.

To set up and run the build cube process in Process Scheduler Manager:

1. Open Process Scheduler Manager and choose Add or Update/Display.
2. Create a Process Definition for your new cube build process.
3. Open Cube Manager and choose Build, Build Cube, click the Schedule button.
The Process Request page appears.
4. Specify Server name, Recurrence, Time Zone, Run Date, and Run Time.
5. Specify the cube to be built by clicking the Select check box.
6. Click the OK button to run the process.

CHAPTER 5

Securing Cubes

This chapter will help you secure the various components of the cubes you can create with Cube Manager and guide you through troubleshooting the cube build process itself. You can check your cubes for various performance-related problems, including proper data loading and access parameters.

Setting Cube Manager Security

Use the centralized Maintain Security application in PeopleTools to set permissions for Cube Manager.

Component Permissions for Cube Manager

Usage	Use this page to control the operations a user is allowed to perform within Cube Manager
Object Name	ACL_COMPONENT2
Navigation	Maintain Security, Use, Permission Lists Click the Pages tab.

Component Permissions

CUBE_MANAGER

View All

First

1-5 of 5

Last

Edit Pages	Authorized?	Bar Label	Component
Edit Pages	<input checked="" type="checkbox"/>	Design	Dimensions
Edit Pages	<input checked="" type="checkbox"/>	Design	Cube Definitions
Edit Pages	<input checked="" type="checkbox"/>	Design	Attribute Names
Edit Pages	<input checked="" type="checkbox"/>	Build	Cube Instances
Edit Pages	<input checked="" type="checkbox"/>	Build	Build Cubes

Select All

Deselect All

OK

Cancel

Cube Manager - Component Permissions page

Edit Pages	An Edit Pages button appears for every Component under the current menu.
Authorized?	This column shows which Components are accessible to the current Permission List.
Bar Label	This shows the menu bar label that you click to access a component.
Component	This shows the name of each component beneath the authorized menu.
Select All	If you want to grant access to most or all items, click Select All . All items are then selected (highlighted). This button quickly grants access to all menu items. When you want to grant access to <i>most</i> of the menu items, after selecting all the items, you can clear those you don't want.
Deselect All	All items are no longer highlighted.



For more information about securing components using Permission Lists, see Working with Permission Lists.

Page Permissions for Cube Manager

Usage	This is where you set the actions a user can complete on a page.
Object Name	ACL_PAGES2
Navigation	Maintain Security, Use, Permission Lists Click the Pages tab, then click Edit Pages.

Page Permissions

CUBE_MANAGER / Design / Dimensions

Page	Authorized?	Display Only
Dimension	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Rollup Inputs	<input checked="" type="checkbox"/>	<input type="checkbox"/>

View All First 1-2 of 2 Last

Actions

☒ Add

☒ Update/Display

☐ Update/Display All

☐ Correction

☐ Data Entry

Select All

Deselect All

OK Cancel

Cube Manager Page Permissions page

Authorized?

Select this option to enable a user to access the page. After doing so, you need to decide the degree to which a user is authorized on a page by selecting **Display Only** or one or more of the available options in the **Actions** group box.

Display Only

Select this option to enable the user to view the information provided by the page, but not to alter any of the data. To enable write access to a page, the **Display Only** check box must be clear.

Actions

If you want the user to be able to alter the data presented by the page, you can select from the options that appear in the **Actions** group box, such as **Add**, **Update/Display**, and **Correction**. The options that are available depend upon the options selected when the page was initially developed in Application Designer.

Select All

Grant access to all pages and all the actions for each page.

Deselect All

All items are longer highlighted.



For more information about securing pages using Permission Lists, see Working with Permission Lists.

Troubleshooting Cube Manager

This section examines some of the tools available for troubleshooting problems. It provides an example of how to narrow the focus of the problem.

PeopleSoft Log	<p>The PeopleSoft Log details the entire cube build process. This should be the first place you check when an error occurs.</p> <p>These log files reside on the appserver that runs the cube process. Cube Manager puts them under the application server temp directory. In the temp directory, it creates a directory called "PS" and then under that is a directory for whatever you called your runcontrol during your cube build process. For example C:\temp\ps\testruncontrol (where c:\ is the root of the appserver).</p> <p>The log files for each dimension are here, and the files are prefixed with "D-". Log files for your facts are prefixed with "F-" and a cube definition log is prefixed with "CD" and a fact error log.</p>
Process Monitor	<p>The Process Monitor is where you see the status of the cube build process. It is usually the first place that indicates that an error in the process has occurred. It tells you that you need to start the troubleshooting process, but it doesn't really give you a lot of detailed information.</p>
Essbase Logs	<p>The Essbase Server Log (essbase.log located in the Essbase directory) is a text file that can be viewed from Explorer, but it can also be viewed from within Application Manager. This log details all activity on the essbase server. For example, if you launch the cube build process and it fails right away, the Request Parameters tab on the process monitor might say "could not connect to Essbase." Check the Essbase Server log to see if an error occurred in the connection, or if there is another error that occurred.</p> <p>Essbase Application Log (located Essbase:\App\'application name'.log) is also a text file that can be viewed from Explorer and from Application Manager. It details all the activity on a specific application, such as data load and data extraction, what user is accessing the cube, and so forth. If your cube build is successful but you don't see any data when you look at the outline, you should check the application log to see if there were any errors in the data load process.</p> <p>Essbase Error Code List. In some cases, a very general Essbase error message appears on the PeopleSoft log. The Essbase Error Code List provides a list of Essbase error codes with descriptive text of the error messages to give you a little more information.</p>
Hyperion	<p>The Hyperion website can also provide valuable troubleshooting information: www.hyperion.com.</p>
Cognos	<p>The Cognos website can also provide valuable information: www.cognos.com.</p>
PS/ROLAP	<p>.iamedatada located in the Strategy server bin directory contains the repository path and other crucial variables.</p>

For more information about troubleshooting Strategy PS/ROLAP, see Using Error Logs in your DecisionMaster Administrator's Guide.

Troubleshooting Examples

When you are troubleshooting, you typically have to check general logs first to gradually pinpoint the error. The following example provides you with some troubleshooting strategies.

To troubleshoot a failed cube PS/ROLAP platform:

1. First check the cube instance log for obvious errors.
2. Try running the cube again with the Generic Star Schema platform instead.
 - If this works as a generic, then you need to verify that your PS/ROLAP (CA Strategy) platform options are correct in the cube instance.
 - Verify that your “.iametadata” configuration file in the Strategy server “bin” directory has the correct repository path and that other variables look correct.
 - Verify that Strategy is up and running in UNIX when you try to run on the PS/ROLAP (CA Strategy) platform. If the server is down, then you may see errors like “can’t connect.”
 - Look through the cube instance log file for items like "can’t find a table" or something similar.
3. If you also fail to generate a successful cube on a generic star schema platform, then look at your cube instance log file.

This file usually makes the problem dimension or fact the last entry in this file. Sometimes it’s hard to determine which one it is, but you can also check each dimension log file for the statement like “Process Complete” at the bottom of the log file.

- Cube Manager first creates the cube instance log file, then initiates a process for each dimension. These dimensions run simultaneously until each one finishes, and then it starts processing the fact data. So if you error out and don’t see any fact data, then you know it has to be a dimension problem.
- You can verify that you have found the problem dimension by pulling it out of the cube build and re-running.
- Once you have determined which dimension it is, then open up the dimension and see how it is built. If it is based on a query, then run the query to see if you get results.
- If it is based on a tree, then try to open the tree.

CHAPTER 6

Glossary

A note on OLAP (online analytical processing) terminology: Although the term “cube” might suggest a three-dimensional structure, PeopleSoft uses the term to mean an analytical database structure capable of having numerous dimensions, not necessarily just three. A dimension usually corresponds to a single type of field—such as time, account, or department.

Data Mart A database containing data summarized from one or more transactional systems, optimized to support the business analysis needs of a particular department, such as Sales, Finance, or Marketing. Users analyze the data using OLAP tools and ad hoc query/reporting tools. The data mart can be a relational database or a multidimensional database, and its data can come directly from transactional systems or from a data warehouse.

An organization usually has a data mart for each department that performs a lot of analysis. Data marts can lighten the load on your corporate data warehouse by satisfying the needs of the most frequent users.

Data Warehouse A large database containing data summarized from one or more transactional systems optimized to support the analysis needs of the enterprise. An ideal data warehouse contains all the data necessary to make business decisions. Users analyze the data in the warehouse using OLAP tools and ad hoc query/reporting tools.

An increasing number of organizations have virtual data warehouses, where the data warehouse is not one physical database, but a collection of specialized (and distributed) data marts.

Dimension A single element of a business model, such as product, department, or location. Cube Manager uses the term Conforming Dimension.

Drill Down The ability to go down to the next level of detail in a set of data. For instance, if you’re looking at an expense figure for a division, you can drill down to the expenses for each department in the division.

DSS Decision Support System. A DSS is a workstation-based analysis and reporting system, typically aimed at analysts and line managers. OLAP tools provide a powerful DSS.

EIS	Executive Information System. An EIS is a workstation-based analysis and reporting system for executives. An EIS provides a higher-level view of the data than a DSS, and typically requires less knowledge about the underlying transactional systems. OLAP tools provide a powerful EIS.
Cube	See Multidimensional Database (MDDB).
HOLAP	Hybrid Online Analytical Processing. Microsoft's PLATO product is a good example of a HOLAP tool. HOLAP tools use a relational star schema to store base fact information, and a separate multi-dimensional structure to store the related aggregated information.
Measure	A measure represents the amounts brought into a cube—the numerical data. For example, a cube might contain dimensions for Account, Department, and Period, plus a Measure/Fact for Budget Amount. Essbase allows only one measure for each cube, unless you create a measures dimension to hold more than one measure. For example, you might create a dimension called Measures, and in that dimension you might have two members at the same level, Actual and Budget.
Member	A member is the OLAP equivalent of a node or detail value on a PeopleSoft tree. A member is a single item within a dimension, such as a single product name, department ID, or part number. Member names must be unique, even across dimensions. Cube Manager uses the term Dimension Field Mapping to identify members, dimension parents, and label mappings.
Metadata	Information about data. Metadata is the information a database or application stores to describe your business data. At its simplest, metadata defines the structure of a data field—its data type and size, for example. Metadata can also describe more complex data relationships, such as the rollup structure for a chart of accounts. Reporting and analysis tools should be able to use this metadata to let users access data just as they would from within the application, without having to understand how it is stored.
Multidimensional Analysis	A type of analysis that enables you to look at data from many different perspectives. You identify the dimensions, or attributes, of the data, and then combine the dimensions in various ways. For example, you might identify five dimensions of your sales data: sales, region, channel, product line, and time. Once you've identified the dimensions, you can slice and dice the data based on combinations of these dimensions, such as sales in the Western region for the last quarter.
Multidimensional Database	A database that stores data for multidimensional analysis in a proprietary multidimensional format. Users access MDDBs exclusively for reporting and analysis, never transaction processing, so they are optimized for retrieval speed.

OLAP	Online Analytical Processing. OLAP is the multidimensional analysis of application data, performed interactively. The acronym contrasts with OLTP (Online Transaction Processing), which is what most production business application systems do.
OLTP	Online Transaction Processing. OLTP refers to the applications that perform the business transactions that keep your company running, like processing invoices or enrolling employees in benefits programs.
ROLAP	Relational Online Analytical Processing. ROLAP adds to OLAP the ability to use two or more separate databases as data sources for defining cube dimensions. You can attach to several different databases and use information from each source to build a multidimensional cube.
Slice and Dice	Another term for multidimensional analysis. When your data has three (or more) dimensions, you can think of it as being arranged in a cube, with each side representing a dimension. When you analyze the data, you slice off part of the cube or dice it to get to an individual cell.
Transactional System	A business application for performing the business transactions that keep your company running. Transactional applications, and the databases that support them, are optimized for quick transaction processing. Because they are constantly changing and are not optimized for data retrieval, transactional system databases are not usually the best source of data for analysis.

Index

A

- access controls 5-2
- aggregation 4-18
- attribute names 3-20
 - designing 3-20
- attribute queries 2-8
- attributes 3-1

B

- bind variables 4-22
 - example 4-24
- bookmarks 3-2

C

- CD-ROM
 - ordering ii
- cells 1-3
- Cognos
 - PowerPlay 4-12. *See also* platform options
 - website 5-4
- components of a cube 3-1
- consolidation in OLAP cubes 1-5
- cube
 - building 4-1, 4-18
 - components 3-1
 - designing 3-1
- cube definitions 3-2, 3-15
 - designing 3-14
- cube input field 3-18
- cube inputs 3-17
- cube instance definitions 3-2, 4-1
 - filter 4-3
- currency category 2-10
- currency conversion type 2-10
- currency name 2-10
- current period 2-12

D

- data mart 6-1
- data source queries 2-13
- data storage 2-10
- data warehouse 6-1
- dimension 6-1

- dimension input field page 3-13
- dimension queries 2-3
- dimensions 1-3, 3-1
 - designing 3-3
 - options 3-8
- drill down 6-1
- DSS 6-1

E

- EIS 6-2
- error logs 5-4
- Essbase 1-13
 - and query attributes 2-9
 - cube instance definitions 4-10
 - platform options 4-6
- Essbase attributes
 - currency category 2-10
 - currency conversion type 2-10
 - currency name 2-10
 - data storage 2-10
 - expense item 2-10
 - time balance 2-10
 - unary operator 2-11
- Essbase Logs 5-4
- expense item 2-10

F

- filter
 - cube criteria 4-4
- filter, data 3-2
- flip sign 2-9

G

- general attributes
 - flip sign 2-9
 - label 2-9
 - user-defined 2-9
- generations *See* levels
- generic star schema 4-16. *See also* platform options

H

- hierarchies in cubes 1-4
- HOLAP 6-2
- HOLAP, definition of 1-2

hybrid OLAP 1-2
 Hyperion
 Essbase 4-6. *See also* platform options
 website 5-4

J

join 4-22. *See also* bind variables

L

label attribute 2-9
 level attribute queries 2-12
 prior resolution count 2-12
 resolution type 2-12
 level attributes 3-1
 level-attribute queries 1-9
 levels 2-1
 Log Client Request 4-28
 log files 5-4
 long description 2-11

M

MDC files 4-13
 MDL files 4-13
 measure 6-2
 measures 1-3
 member 6-2
 members 1-3
 metadata 1-8, 6-2
 in OLAP cubes 1-8
 MOLAP, definition of 1-2
 multidimensional analysis 6-2
 multidimensional database 6-2
 multiple dimension cubes {XEcubes
 multiple dimension 1-4

O

OLAP 6-3
 OLAP cubes 1-3
 building (overview) 1-13
 cells 1-3
 consolidation 1-5
 designing (overview) 1-13
 dimensions 1-3
 hierarchy 1-4
 measures 1-3
 members 1-3
 metadata 1-8
 metadata, designing 2-1
 rollups 1-4
 OLAP tools 1-9

Essbase 1-13
 PowerPlay 1-10
 PS/ROLAP 1-13

OLTP 6-3
 online transaction processing 1-1

P

PeopleBooks
 CD-ROM, ordering ii
 printed, ordering ii
 PeopleSoft log 5-4
 PeopleSoft metadata 1-8
 permission lists 5-1
 platform options 4-6
 PowerPlay 1-10
 and query attributes 2-11
 cube instance definitions 4-14
 MDC files 4-13
 MDL files 4-13
 PowerPlay attributes
 long description 2-11
 short label 2-11
 prior resolution count 2-12
 Process Monitor 5-4
 Process Scheduler 4-26
 and Cube Manager 1-9
 setting up process definitions 4-27
 setting up requests 4-28
 PS/ROLAP 1-13, 4-16. *See also* platform options
 and query attributes 2-11
 PS/ROLAP attributes
 current period 2-12
 sequence in year 2-11
 sequence number 2-11

Q

queries
 and OLAP cube metadata 1-8
 attribute 2-8
 attribute definition 2-3
 data source 2-13
 data source definition 2-3
 dimension 2-2, 2-3
 Essbase attributes 2-9
 general attributes 2-9
 level attribute 1-9
 level-attribute definition 2-3
 narrow definition 2-4
 PowerPlay attributes 2-11
 PS/ROLAP attributes 2-11
 security 1-8
 security definition 2-3
 using as OLAP cube metadata 2-2
 wide definition 2-7

R

- resolution type 2-12
- ROLAP 6-3
- ROLAP, definition of 1-2
- rollup inputs 3-11
- rollup options 3-9
- rollups 1-4
- run control ID 3-2

S

- securing cube components 5-1
- security 5-1
- security queries 1-8
- sequence
 - in year 2-11
 - number 2-11
- short label 2-11
- slice and dice 6-3
- snowflake schemas 1-2

- sparsity settings 3-2
- SQR Runtime 4-28
- star schemas 1-2
- Sterling Eureka
 - StrategyPS *See* PS/ROLAP. *See* PS/ROLAP

T

- time balance 2-10
- transactional system 6-3
- Trees
 - and OLAP cube metadata 1-8
 - using as OLAP cube metadata 2-1
- troubleshooting 5-4
- troubleshooting examples 5-5

U

- unary operator 2-11
- user defined attributes 2-9

