Oracle9*i* OLAP Services

Concepts and Administration Guide

Release 1 (9.0.1)

June 2001

A88755-01

Oracle9i OLAP Services Concepts and Administration Guide, Release 1 (9.0.1)

A88755-01

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle9*i*, JDeveloper, and Net8 are trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

6 Tuning

A Upgrading from Express Server

Index

Send Us Your Comments

Oracle9i OLAP Services Concepts and Administration Guide, Release 1 (9.0.1)

A88755-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- FAX - 781-684-5880. Attn: Oracle OLAP Services
- Postal service:
  Oracle Corporation
  Oracle OLAP Services Documentation
  200 Fifth Avenue
  Waltham, MA 02451-8720
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

What this manual is about

The *Oracle9i OLAP Services Concepts and Administration Guide* describes how to use Oracle OLAP Services (hereinafter referred to as “OLAP Services”) for business analysis. It introduces the concepts underlying analytical applications and multidimensional querying, and the tools used for application development and system administration.

This guide provides some information that is specific to Unix platforms. If you are using OLAP Services on a Windows system, refer to the *Oracle9i OLAP Services Concepts and Administration Guide for Windows*. It is located on the Oracle9*i* Windows installation CD.

Intended audience

This manual is intended for application developers and database administrators who are responsible for providing analytical services.

Before you begin

Before you can use OLAP Services, you must install Oracle9*i* Enterprise Edition.

Structure of this document

The *Oracle9i OLAP Services Concepts and Administration Guide* is structured as follows:

- Chapter 1 provides an introduction to online analytical processing and presents an overview of OLAP Services.
- Chapter 2 describes the programming environment for developing OLAP applications.
- Chapter 3 introduces the tools for administering OLAP Services.
- Chapter 4 describes how to set up your data warehouse and create metadata to support OLAP applications.
- Chapter 5 explains localization issues and explains how to choose the appropriate settings for the Oracle Globalization Support system.
- Chapter 6 identifies various ways of improving the performance of OLAP Services.
- Appendix A summarizes the differences between Express Server and OLAP Services. If you are upgrading from Express Server, you should read this first.

Related Documentation

This manual is part of a set of documentation, which also includes the following books:

- *Oracle9i OLAP Services Developer's Guide to the Oracle OLAP API*. Introduces Java programmers to the Oracle OLAP API, the application programming interface for Oracle OLAP Services. Through OLAP Services, the OLAP API provides access to data stored in an Oracle database. The OLAP API's capabilities for querying, manipulating, and presenting data are particularly suited to applications that perform online analytical processing.
- *Oracle9i OLAP Services OLAP API Reference*. Provides online reference documentation for the Oracle OLAP API, the Java application programming interface for Oracle OLAP Services.
- *Oracle9i OLAP Services Developer's Guide to the OLAP DML*. Explains how application developers can perform complex data analysis tasks (such as

forecasts, models, allocations, and some types of non-additive aggregation) by using the OLAP DML.

- *Oracle9i Data Warehousing Guide*. Discusses the database structures, concepts, and issues involved in creating a data warehouse to support OLAP solutions.

Conventions

Text conventions

You will find the following text conventions in this document.

Convention	Usage
Boldface text	Indicates menu items, command buttons, options, field names, and hyperlinks. Bold text is also used for notes and other secondary information in tables or procedures (for example, **Tip**).
`Fixed-width` text	Indicates folder names, file names, operating system commands, and URLs. Also indicates examples and anything that you must type exactly as it appears. **For example:** If you are asked to type `create user`, you would type all the characters exactly as shown in the fixed-width font.
Italic text	Indicates variables, including variable text. Italic type is also used for emphasis, for new terms, and for titles of documents.
UPPERCASE text	Indicates a command, an object, or an acronyms.

Mouse usage

Always use the left mouse button (“click”) unless you are specifically instructed to use the right mouse button (“right-click”).

The term “left mouse button” refers to the dominant button. If you have reconfigured your mouse to reverse the functions of the left and right buttons, then you will need to use the reverse button when you follow the procedures in this manual.

About graphical user interfaces

Graphical user interfaces, such as Oracle Enterprise Manager, typically provide many alternative ways of accomplishing the same task: right- and left-mouse clicks, keystroke combinations, icons, and menus. The procedures in this guide identify only one or two methods. For alternative methods, refer to Help.

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup that will assist the disabled community in accessing our documentation.. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program web site at:

`http://www.oracle.com/accessibility/`

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

1

Learning the Basics

Chapter summary

This chapter explains the basics of using Oracle OLAP Services and related client software for analytical applications. By reading this chapter, you will get an overview of its features.

List of topics

This chapter includes the following topics:

- Why OLAP?
- The Oracle OLAP Solution
- Building Analytical Applications
- Managing OLAP Services

Why OLAP?

Relational databases store the world's data

Relational databases have dominated database technology by providing the online transactional processing (OLTP) that is essential for businesses to keep track of their affairs. Designed for efficient selection, storage, and retrieval of data, relational databases are ideal for housing gigabytes of detailed data.

The success of relational databases is apparent in their use to store information about an increasingly wide scope of activities. As a result, they contain a wealth of data that can yield critical information about a business. This information can provide a competitive edge in an increasingly competitive marketplace.

Analytical processing answers business questions

The challenge is in deriving answers to business questions from the available data, so that decision makers at all levels can respond quickly to changes in the business climate. While a standard transactional query might ask, “When did order 84305 ship?” an analytical query might ask, “How do sales in the Southwestern region for this month compare with plan? or with sales a year ago?”

The first question involves simple data selection and retrieval. However, the second question involves inter-row calculations, time series analysis, and access to aggregated historical and current data. This is online analytical processing — OLAP.

The data processing required to answer analytical questions is fundamentally different from the data processing required to answer transactional questions. The following table highlights the major differences.

Characteristic	Transactional Query	Analytical Query
Typical operation	Update	Analyze
Age of data	Current	Historical
Level of data	Detail	Aggregate
Data required per query	Minimal	Extensive
Querying pattern	Individual queries	Iterative queries

Types of analytical reporting applications

Applications that support business analyses fall into these major groups:

- Standard and ad-hoc reporting
- Analytical reporting
- Predictive analysis

Oracle® provides the technology for all of these types of applications. OLAP Services and its development tools are particularly suited to analytical reporting and predictive analysis applications.

Analytic applications can support many facets of a business and offer high returns on the investment. Here are just a few examples of analytical applications, as reported by IDC:

- Accounting. Forecasting, budgeting, cost and profitability analysis, and consolidation
- Human Resources. Skills consolidation, labor scheduling and optimization
- Distribution. Scheduling and optimization
- Sales Force Automation. Cross-selling and territory analysis
- Marketing. Churn and market-based analyses
- Retailing. Site location and demographic analysis
- Manufacturing. Demand planning and forecasting
- Health Care. Outcomes analysis
- Financial Services. Risk assessment and management

Moreover, the forecasting and modeling features enable you to develop demand planning applications.

This guide will introduce you to the tools for developing these types of applications.

The Oracle OLAP Solution

Oracle9*i* supports business intelligence

Analytical queries require an online analytical processing (OLAP) solution. Oracle provides comprehensive support for OLAP:

- The Oracle relational database management system (RDBMS) is an efficient and secure way to store vast amounts of data. Within the Oracle database, you can create a data warehouse that provides data in a form suitable for business analysis.
- OLAP Services provides a Java OLAP API, a calculation engine, and an analytic workspace. These facilities enable you to build analytical applications that support complex statistical, mathematical, and financial calculations along with predictive analysis functions such as forecasting, modeling, consolidations, allocations, and scenario management. Because the OLAP API is all Java, OLAP

Services supports deployment of analytical applications to large, geographically distributed user communities on the Internet.

- The Oracle BI Beans complements OLAP Services by providing pre-built OLAP-aware application building blocks. The BI Beans can be used within Oracle JDeveloper or other Java development environments to build analytical applications, which can be deployed as either Java or HTML (“thin”) clients.

This guide explains the relationships among these components from the perspective of both the system administrator and the application developer.

Complementary functionality

The Oracle RDBMS and OLAP Services provide complementary functionality to support the most versatile and high performance applications. The RDBMS provides detail data, summary management, and one-dimensional calculations using the SQL-99 OLAP extensions. OLAP Services expands these capabilities to provide forecasting, modeling, what-if scenarios, and multidimensional calculations.

Building Analytical Applications

Java programming language

Java is the language of the Internet, and also the language of OLAP applications. Using Java, an application developer can write a standalone application or an *applet*, which is a program that can be included in an HTML page and executed in a browser. As an object-oriented, platform-independent, network-based, and secure language, Java is fast superseding C++ and Visual Basic as the language of choice for application developers:

- Object oriented. Java allows application developers to focus on the data and methods of manipulating that data, rather than on abstract procedures; the programmer defines the desired object rather than the steps needed to create that object. Almost everything in Java is defined as an object.
- Platform independent. The Java compiler creates byte code that is interpreted at runtime by the Java Virtual Machine (JVM). As the result, the same software can run on all Windows, Unix, and Macintosh platforms where the JVM has been installed. All major browsers have the JVM built in.
- Network based. Java was designed to work over a network, which allows Java programs to handle remote resources as easily as local resources.

- Secure. Java code is either trusted or untrusted, and access to system resources is determined by this characteristic. Local code is trusted to have full access to system resources, but downloaded remote code (that is, an applet) is not trusted.

  The Java "sandbox" security model provides a very restricted environment for untrusted code. For example, untrusted Java code cannot read to or write from files on the local file system, run programs, load libraries, define native method calls, or make network connections except to the originating host computer. A security manager determines the system resources that an applet can access. However, a signed applet, which identifies itself as being from a trusted source, has full access to system resources the same as local code.

JavaBeans for business intelligence

JavaBeans are the building blocks of application development. They are reusable pieces of Java code that can be assembled quickly into an application.

The BI Beans provide the basic building blocks for an OLAP application: Connecting to a database; authenticating user credentials; selecting and fetching data; and displaying the data in a variety of tabular and graphical formats. Using the BI Beans, you can create applications with a common "look and feel," enabling users to gain expertise quickly in the new product.

The Oracle OLAP API

The Oracle OLAP API is a Java application programming interface to OLAP Services. It is a querying language that selects and manipulates data in a data warehouse for display in a Java client. The analytical BI Beans are built using this API; you can extend (or even replace) the functionality provided by the BI Beans by using Java classes.

The Oracle OLAP DML

The Oracle OLAP DML is a data manipulation language that extends the analytic support of the OLAP API to include forecasting, modeling, and what-if scenarios. It operates on data that is stored (permanently or temporarily) in multidimensional objects in the analytic workspace.

Managing OLAP Services

Definition: OLAP Services

In its broadest definition, OLAP Services consists of the following components: the Oracle Java OLAP API, one or more OLAP services that run as child processes of a database instance, a metadata repository in each database instance, and tools within Oracle Enterprise Manager for creating OLAP metadata and managing OLAP services. However, OLAP Services is frequently used in this guide to refer to one or more components (such as an OLAP service child process) where distinctions among software components are not necessary.

Integrated tools

All of the tools that you use to manage OLAP services are integrated into Oracle Enterprise Manager. You do not need to use a separate set of tools.

Instance management

OLAP Services Instance Manager is the primary tool for managing OLAP services. You can use Instance Manager to do these system administration tasks:

- Start and stop existing services, and create additional services.
- Monitor user sessions.
- Add and remove optional features so that you can use the features you want using the fewest resources.
- Fine-tune its operating parameters so that OLAP services perform optimally in your unique data processing environment.

For information about managing OLAP services, read “Introducing OLAP Services Instance Manager” on page 3-3.

Metadata definition

The OLAP component of Oracle Enterprise Manager generates metadata for use by applications. This metadata is used to identify the facts available for analysis, and to create and populate multidimensional objects. These objects are needed to process analytical queries using data stored permanently in an Oracle data warehouse.

For information about configuring your data warehouse for use with OLAP Services, read “Using Oracle Enterprise Manager” on page 4-10.

OLAP metadata

Metadata is typically defined as “data about data.” OLAP applications can query the metadata repository to find out what data is available for them to analyze and display. While the data is stored in tables in an Oracle data warehouse, the metadata identifies the data in terms of the multidimensional objects they will become in OLAP Services: dimensions, measures, and attributes. Moreover, the metadata defines the dimension hierarchies required to aggregate the data into meaningful levels of detail.

For more information about creating metadata, read “Creating Metadata” on page 4-10.

Language support

Globalization Technology provides the Oracle standard for internationalizing and localizing Oracle products. Its use throughout Oracle9*i* allows text data in native languages to be passed between the RDBMS and OLAP Services without data loss or performance degradation.

Additional information about Globalization Technology is provided in “About Oracle Globalization Technology” on page 5-2.

Security methods

The metadata maps OLAP Services security directly to the security of the RDBMS. Users can access through OLAP Services only the data that they are authorized to access in the database.

Users can provide proof of identity in the following ways:

- A database user ID and password that will be validated against either:
  - The local user repository in the associated RDBMS.
  - An Oracle Internet Directory, which is a Lightweight Directory Access Protocol (LDAP) Version 3 service.
- A client-side X.509 certificate that is sent over a Secure Sockets Layer (SSL) connection for single sign-on for enterprise users. The certificate will be validated against an Oracle Internet Directory within the Public Key Infrastructure internet security strategy.

For more information about these security methods, refer to your Oracle9*i* networking and security documentation. For information about defining users, refer to “Granting Access Rights to Users” on page 3-27.

2

Developing OLAP Applications

Chapter summary

This chapter presents the rich development environment and the powerful tools that you can use to create OLAP applications.

List of topics

This chapter includes the following topics:

- The Java Solution
- Introducing the BI Beans
- Understanding the OLAP API
- Data Storage Alternatives

The Java Solution

Introduction

To develop an OLAP application, you use the Java programming language. Java enables you to write applications that are platform-independent and easily deployed over the Internet.

Java is the preferred programming language for an ever-increasing number of professional software developers. For those who have been programming in C or C++, the move to Java is easy because it provides a familiar environment while avoiding many of the shortcomings of the C language. Developed by Sun Microsystems, Java is a secure, object-oriented, portable, and multithreaded language.

The OLAP API is a Java-based application programming interface that provides access to multidimensional data for analytical business applications. The OLAP API fetches data stored in a data warehouse into the OLAP multidimensional data cache for manipulation by its analytical engine. Java classes in the OLAP API provide all of the functions required of an OLAP application: Connection to an OLAP instance; authentication of user credentials; access to data in the RDBMS controlled by the permissions granted to those credentials; and selection and manipulation of that data for business analysis.

The BI Beans simplify application development by providing these functions as JavaBeans. Moreover, the BI Beans include JavaBeans for presenting the data in graphs, crosstabs, and tables.

Ease of deployment

With the rise in Internet technology, more and more businesses are recognizing the savings they can accrue just by changing the way they deploy their applications.

Traditional *thick client* applications implement many of their functions on the user’s computer, thus requiring a large proportion of installed code. However, the days are gone when a team of technicians are required to install and maintain applications software on hundreds or thousands of individual desktop computers for a large user base. Instead, Java thick-client applications download the needed software to client computers automatically at run-time.

Alternatively, system administrators can deploy *thin client* applications that do not download any Java to client computers. These applications run on servers that users world wide can access using Java clients such as their Web browsers.

Regardless of whether you choose a thick-client or a thin-client configuration, Java applications provide an immediate solution to the problems inherent in supporting large user communities, which typically are equipped with a variety of incompatible hardware and software platforms.

Oracle JDeveloper

Oracle JDeveloper provides an integrated development environment (IDE) for developing Java applications. Although third-party Java IDEs can also be used effectively, only JDeveloper achieves full integration with the Oracle database and BI Beans wizards. The following are a few JDeveloper features:

- Remote graphical debugger with break points, watches, and an inspector.
- Multiple document interface (MDI)

- *Codecoach* feature that helps you to optimize your code
- Generation of 100% Pure Java applications, applets, servlets, Javabeans, and so forth with no proprietary code or markers
- Oracle database browser

For more information about the Java programming language, browse the Sun Microsystems Java Web site at `java.sun.com`. For information about JDeveloper, search the Oracle Web site at `www.oracle.com`.

Metadata

The OLAP API and the BI Beans use metadata to provide the information they need about multidimensional objects defined in an Oracle data warehouse, such as measures and dimensions. For information about metadata and other requirements, refer to Chapter 4.

Runtime repository

A runtime repository in the Oracle database allows users to save their personal analyses and to share their discoveries with other users.

Diagram: Thick-client application

The components of an OLAP thick-client application are grouped into three tiers, which can be on separate platforms or the same platform:

- Java client tier. A Java application can run either in a browser or directly in the Java Runtime Environment (JRE). The BI Beans that are dedicated to presenting the data and metadata also run on this tier.
- Application server tier. The “brains” of the application run on this tier, which includes the OLAP API and the OLAP BI Beans that are built using the OLAP API.
- Data server tier. The Oracle RDBMS and OLAP service form the data server tier, where the data is stored, selected, and manipulated. An OLAP API component also runs on the data server tier.

The following diagram shows these relationships in a thick-client configuration.

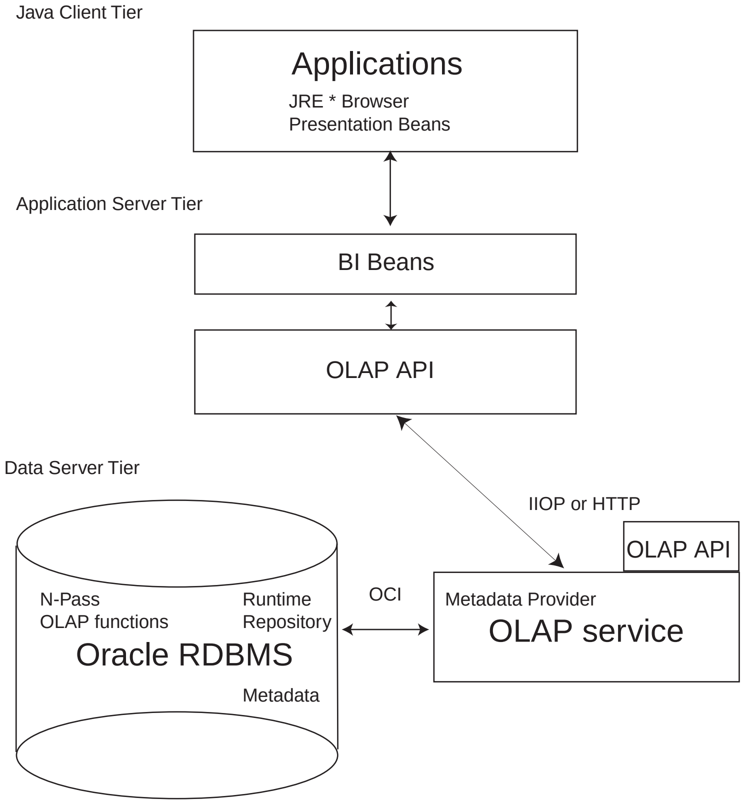
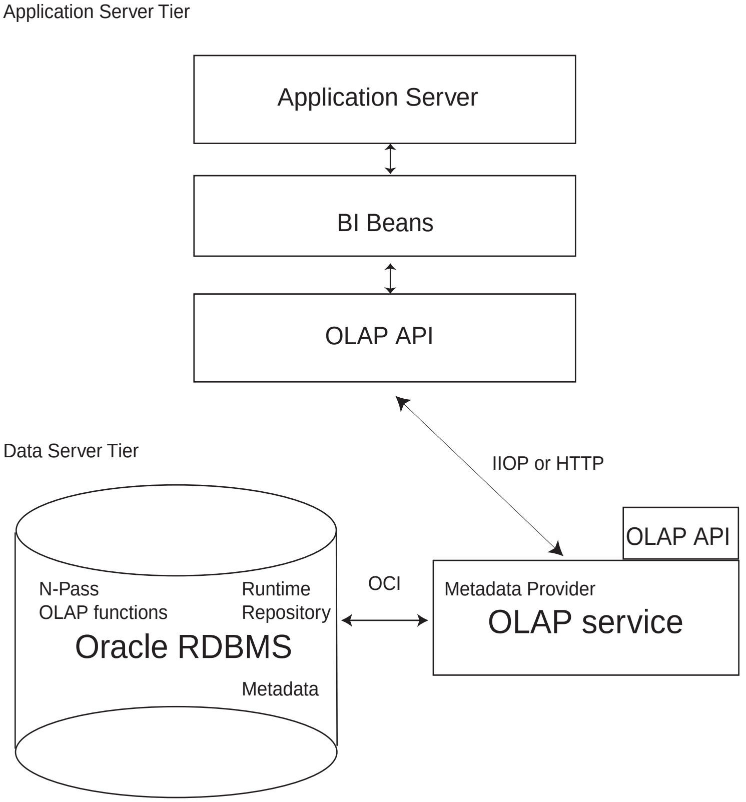


Diagram: Thin-client application

The components of an OLAP thin-client application are grouped into two tiers, which can be on separate platforms or the same platform:

- Application server tier. The “brains” of the application run on this tier, which includes a Web server, the OLAP API and the OLAP BI Beans, both presentation and analytical.
- Data server tier. The Oracle RDBMS and OLAP service form the data server tier, where the data is stored, selected, and manipulated. An OLAP API component also runs on the data server tier.

The following diagram shows these relationships in a thin-client configuration.



Introducing the BI Beans

Basic functionality

The BI Beans provide reusable components that are the basic building blocks for OLAP decision support applications. Using the BI Beans, developers can rapidly develop and deploy new applications, because these large functional units have already been developed and tested — not only for their robustness, but also for their ease of use. And because the BI Beans provide a common look and feel to OLAP applications, the learning curve for end users is greatly reduced.

Two groups of BI Beans are currently available:

- Presentation Beans display the data in a rich variety of formats so that trends and variations can easily be detected. Among the Presentation BI Beans currently available are Graph, Table, and Crosstabs.

  The Presentation Beans can be implemented as a thick client or a thin client. Thick clients best support users who do immersed analyses, that is, use the system for extensive periods of time with a lot of interaction. For example, users who create reports benefit from a thick client. Thin clients best support remote users who use a low bandwidth connection and have basic analytical needs. Thin clients can be embedded in a portal or other Web site for these users.
- OLAP BI Beans acquire and manipulate the data. The OLAP BI Beans use the OLAP API to connect to a data source, define a query, manipulate the resultant data set, and return the results to the Presentation BI Beans for display.

Navigation

The Presentation BI Beans support navigation techniques such as drilling, pivoting, and paging.

- *Drilling* displays lower-level values that contribute to a higher-level aggregate, such as the cities that contribute to a state total.
- *Pivoting* rotates the data cube so that the dimension members that appeared along the X-axis of a graph now appear along the Y-axis, or the dimension members that labeled columns in a crosstab now label rows instead. For example, if products label the rows and regions label the columns, then you can pivot the data cube so that products label the columns and regions label the rows.
- *Paging* handles additional dimensions by showing each member in a separate graph, crosstab, or table rather than nesting them in the columns or rows. For example, you might want to see each time period in a separate graph rather than all time periods on the same graph.

Formatting

The Presentation BI Beans allow you to change the appearance of a particular display. In addition, the values of the data itself can affect the format.

- Number formatting. Numerical displays can be modified by changing their scale, number of decimal digits and leading zeros, currency symbol, negative notation, and so forth. Currency symbols and scaling factors can be displayed in the column or row headers rather than in the cells.

- Stoplight formatting. The formatting of the cell background color, border, font, and so forth can be data driven so that outstanding or problematic results stand out visually from the other data values.
- Ranking. In ranking reports, the numerical rank of each dimension value, based on the value of the measure, is displayed.

Graphs

The Graph bean presents data in a large selection of two- and three-dimensional business chart types, such as bar, area, line, pie, ring, scatter, bubble, pyramid, and stock market. Many of the 2D graphs can be displayed as clustered, stacked, dual-Y, percentage, horizontal, vertical, or 3D effect.

Bar, line, and area graphs can be combined so that individual rows in the data cube can be specified as one of these graph types. You can also assign marker shape and type, data line type, color, and width, and fill colors on a row-by-row basis.

The graph image can be copied to the system clipboard and exported in GIF and other image formats.

Users can zoom in and out of selected areas of a graph. They can also scroll across the axes.

Crosstabs

The Crosstab bean presents data in a two-dimensional grid similar to a spreadsheet. Multiple dimensions can be nested along the rows or columns, and additional dimensions can appear as separate pages. Among the available customizations are: Font style, size, color and underlining; individual cell background colors; border formats; and text alignment.

Users can navigate through the data using either a mouse or the keyboard. They can insert rows and columns to display totals, and edit cells for what-if analysis.

Tables

The Table bean presents data in record format like a relational table or view. In contrast to the crosstab, the table display handles measures individually rather than as members of a measure dimension. Thus, each measure can be manipulated individually.

OLAP BI Beans

The OLAP BI Beans use the OLAP API to provide the basic services needed by an application. They enable clients to identify a database, present credentials for accessing that database, and make a connection. The application can then access the metadata and identify the available data. Users can select the measures they want to see and the specific slice of data that is of interest to them. That data can then be modified and manipulated.

Wizards

The BI Beans offer wizards that can be used both by application developers in creating an initial environment and by end users in customizing applications to suit their particular needs. The wizards lead you step-by-step so that you provide all of the information needed by an application. The following are some of the tasks that can be done using wizards.

- Building a query. Fact tables and materialized views often contain much more data than users are interested in viewing. Fetching vast quantities of data can also degrade performance unnecessarily. In addition to selecting measures, you can limit the amount of data fetched in a query by selecting dimension members from a list or using a set of conditions. A selection can be saved and used again just by picking its name from a list.

  The BI Beans take advantage of all of the new OLAP functions in the database, including ranking, lag, lead, and windowing. End users can create powerful queries that ask sophisticated analytical questions, without knowing SQL at all.

- Generating custom measures. You can define new “custom” measures whose values are calculated from data stored within the database. For example, a user might create a custom measure that shows the percent of change in sales from a year ago. The data in the custom measure would be calculated using the lag method on data in the Sales measure. Because a DBA cannot anticipate and create all of the calculations required by all users, the BI Beans enable users to create their own.

Related information

For more information about the BI Beans, go to the Oracle Web site at `www.oracle.com`.

Understanding the OLAP API

Object oriented

OLAP applications typically have object-oriented user interfaces where users manipulate objects that represent organized groupings of their data. Thus, there is a natural relationship between an object-oriented user interface and an object-oriented API such as the Oracle OLAP API. The OLAP API exploits this natural relationship by providing objects that match the end-user behavior that an application needs.

Object-oriented languages such as Java manipulate data by applying methods on objects. This approach enables the objects to maintain a current state and support incremental modifications to that state. This approach provides excellent support for common OLAP actions such as drill and rotate.

For example, a central activity for users of OLAP applications is refining queries. A user has a question in mind and devises a query to answer that question. In most cases, the initial results of the query prompt the user to want to dig deeper for a solution, perhaps by drilling to see more detailed data or by rotating the report to highlight correlations in the data. The OLAP API is able to use the result of one query as the input to the next query.

Intelligent caching

Analytical queries are by nature iterative. An analyst formulates a query, sees the results, and then formulates other queries based on those results. Since the likelihood is very high in business analysis of needing the same data to answer subsequent queries, the OLAP API caches the metadata so that it is available throughout the session without fetching it again. Moreover, the OLAP API defines the result set of a query geometrically. Using multidimensional cursors, the OLAP API can randomly access disparate regions of the result set. This allows an application to retrieve just the data currently of interest instead of all of the data in the result set. For example, you might scroll to the end of a page without having to fetch all of the data on the page.

To acquire data from a data warehouse, the OLAP API generates SQL statements. Data fetches use many of the newest innovations in Oracle9*i*, including concatenated rollup, scrollable cursors, and query rewrite.

CORBA distributed processing

Common Object Request Broker Architecture (CORBA) is a standards-based network communications architecture that provides a way for applications located on two or more servers to communicate so that they appear to run as a single unit. CORBA runs over a network protocol such as TCP/IP.

In a CORBA environment, every computer has an Object Request Broker (ORB). The application code addresses its requests to the ORB rather than to a particular server or database. The ORB responds to the request with an object registered in its namespace. From the standpoint of the application, the request is the same regardless of whether the registered object is stored on the same computer, on a network server, or somewhere on the Internet.

Objects such as OLAP API-enabled servers and databases are registered in the CORBA namespace.

Calculation capabilities

The OLAP API generates SQL commands to select and manipulate data stored in the database. These SQL commands can include the “N-pass” functions, such as RANK, PERCENTILE, TOPN, BOTTOMN, LAG, LEAD, SUM, AVG, MIN, MAX, COUNT, and STDDEV.

The OLAP API provides expanded calculation capabilities beyond those that can be handled efficiently in other OLAP solutions, such as:

- Totals broken out by multiple attributes
- Suppression of NA and zero rows, columns, and pages
- Row and column calculations
- Union dimensions
- Measures as dimensions
- Inter-row calculations such as the following book-to-bill ratio:

  ```
  Balance(Account "BOOKED", Period "PRIOR")/ Balance(Account
  "BILLED", Period "LAST")
  ```

- Asymmetric queries

Additional calculations, such as modeling, forecasting, and what-if scenarios, can be performed on data in the analytic workspace.

Code example: Selecting values

This OLAP API code fragment demonstrates the selection of dimension values based on the data values of a measure. The Sales measure has four dimensions. The Geography, Channel, and Time dimensions are limited to one member each, then Product members are selected with Sales values greater than 20,000,000.

```
Source geographySel = geography.selectValue("BOSTON");
Source channelSel = channel.selectValue("TOTALCHANNEL");
Source timeSel = time.selectValue("1996");
Source prodSel = product.select(salesSel.gt(20000000));
Source result = sales.join(geographySel).
      join(channelSel).join(timeSel).join(prodSel);
```

Related information

For information about developing applications using the OLAP API, refer to the *Oracle9i OLAP Services Developer's Guide to the Oracle OLAP API*.

For information about programming in the DML, refer to *Oracle9i OLAP Services Developer's Guide to the OLAP DML*.

Data Storage Alternatives

Database or workspace?

The types of analyses performed by your application determine the best choice of a data repository. You must examine the benefits of each storage method in light of your application and decide which one most closely matches your requirements. You can choose to store the data for your business analysis applications from these alternatives:

- Entirely in the relational database. Data is stored entirely in a data warehouse and made available to Java applications by OLAP metadata as described in Chapter 4. During user sessions, data is selected and manipulated in the relational database. This method is typically called Relational OLAP or ROLAP.
- Entirely in the multidimensional analytic workspace. As a routine maintenance task, data is loaded into dimensions and variables in the workspace from one or more sources (including the relational database and flat files) and saved for use by all sessions. During user sessions, data is selected and manipulated in the analytic workspace. This method is typically called Multidimensional OLAP or MOLAP.

- Distributed between the relational database and the analytic workspace. The implementation of this model can, of course, vary widely since it encompasses any scheme that draws on the other two methods. A distributed solution may be desirable when an application requires the advanced calculation capabilities and speed of a MOLAP solution combined with the efficient storage of a relational database. This method is typically called Hybrid OLAP or HOLAP.

Related information

If you are interested in a MOLAP or HOLAP solution, then refer to *Oracle9i OLAP Services Developer's Guide to the OLAP DML* for additional information.

Query and reporting applications

The relational database is the preferred data repository for most query and reporting applications that require read-only access to the data. For these applications, the relational database offers scalability in supporting very large data sets efficiently and manageability with a single set of administrative tools.

Predictive analyses applications

Analytic workspaces should be used as a persistent data store for applications that support predictive analysis functions, such as models and forecasts, and what-if scenarios. Other design choices, such as the types of hierarchies and use of non-additive aggregation methods, may make the analytic workspace the preferred data repository.

Aggregate data

In addition to *where* you store the data is the decision of *what* to store. Aggregate data can either be generated and stored as a data maintenance step, or aggregated whenever it is needed in response to a query. If the data is already available, then response time is quicker. However, you will need more storage space to accommodate the additional data.

In deciding what to store, you should consider the merits of several factors: how frequently the aggregate data is needed; how much time is required to generate the data; whether the response time for generating the data falls within acceptable limits; how much disk space is required to store the data; and whether you have the option of acquiring additional storage media.

3

Administering an OLAP Service

Chapter summary

In this chapter, you will learn how to reconfigure, start, and stop an OLAP service. Typically, you will use OLAP Services Instance Manager, which is a graphical user interface, to perform these tasks. However, a command-line utility provides an alternative means of doing them. The OLAP service and both of these administrative tools operate in a service environment. Before you can start them, you must first start the service environment.

Access to OLAP Services Instance Manager is through Oracle Enterprise Manager. You will use Oracle Enterprise Manager to perform other administrative tasks, such as defining database user names and passwords for the OLAP service itself, other administrators, and application users.

List of topics

This chapter includes the following topics:

- Managing the Service Environment
- Introducing OLAP Services Instance Manager
- Starting OLAP Services
- Stopping or Pausing OLAP Services
- Acquiring System Administration Privileges
- Changing the Configuration Settings
- Unlocking Database Identities
- Managing Sessions
- Viewing Status Messages

- Creating New Services
- Command-line Administration
- Granting Access Rights to Users
- Configuring Instance Manager
- Configuring OLAP Services Agent

Managing the Service Environment

Components of the service environment

OLAP Services runs within a service environment. Before you can run any of the administrative tools for OLAP Services, you must initialize its service environment.

This service environment consists of the following components:

- OLAP Services Agent consists of several processes that run continuously in the background to provide the service environment within which OLAP Services runs. You can configure OLAP Services Agent, as described in “Configuring OLAP Services Agent” on page 3-28.
- A local repository stores configuration information about all local instances of OLAP Services, as described in “Changing the Configuration Settings” on page 3-11. Implemented as a file named `olap.key`.
- The `olap` script starts and stops OLAP Services Agent and OLAP Services.

Procedure: Starting and stopping the service environment

To initialize the service environment, follow these steps:

1. Log in as the installation user and go to the `bin` subdirectory of `$ORACLE_HOME`.
2. To initialize the service environment, type `./olap start` .

   or

   To stop the service environment, type `./olap stop`.

Notes: To confirm that all associated processes are stopped, issue commands such as these:

```
ps -ef | grep xs
```

```
ps -ef | grep os
```

In the `olap` script, `start` is the default parameter and thus is optional.

Introducing OLAP Services Instance Manager

Overview of Instance Manager features

OLAP Services Instance Manager is the primary tool for managing an OLAP service. Integrated into Oracle Enterprise Manager, Instance Manager provides a familiar environment for performing administrative tasks. Among these tasks are the ability for you to:

- Start and stop an OLAP service
- Change the configuration settings that control the behavior of an OLAP service
- View system log files that record status messages from an OLAP service
- Create additional instances of OLAP Services

Complete information about all of the tasks that you can perform using Instance Manager can be found in its Help system. This chapter only highlights some of the most basic tasks.

Procedure: Opening Instance Manager

Follow these steps to open Instance Manager:

1. Start up Oracle Enterprise Manager Console.

   You see the main screen for Oracle Enterprise Manager.

2. Click the plus sign (+) to the left of **Databases**.

   You see a list of databases.

3. Click the plus sign (+) to the left of the database you want to manage.

   You see the Database Connect Information dialog box.

4. Type in your user name and password for the Oracle database.

The list of administrative options expands under that database.

Note: Your ability to administer OLAP Services depends on the privileges granted to your user ID. Refer to “Acquiring System Administration Privileges” on page 3-9 for a discussion of privileges.

5. Right-click **Olap**, then choose **OLAP Services Instance Manager**.

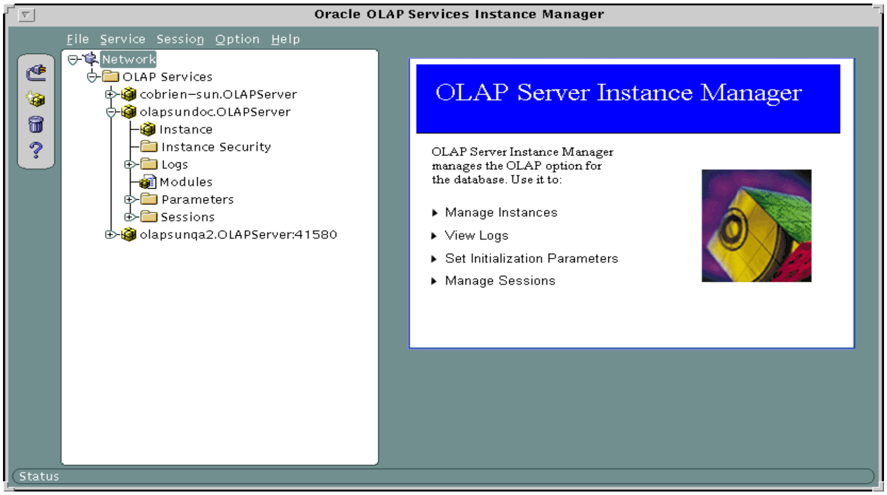
   or

   From the Object menu, choose **OLAP Services Instance Manager**.

   Instance Manager opens in its own window.

Figure: Main window

The following figure identifies the components of the Instance Manager main window with an expanded list of items in the Navigator. You can adjust the width of the Navigator by dragging its right edge. The page that is displayed on the right side of the window adjusts automatically to fit the available space.



Oracle OLAP Services Instance Manager
File Service Session Option Help
Network
OLAP Services
cobrien-sun.OLAPServer
olapsundoc.OLAPServer
Instance
Instance Security
Logs
Modules
Parameters
Sessions
olapsunqa2.OLAPServer:41580
OLAP Server Instance Manager
Status

OLAP Server Instance Manager manages the OLAP option for the database. Use it to:

- Manage Instances
- View Logs
- Set Initialization Parameters
- Manage Sessions

Common problems in starting Instance Manager

If you are unable to connect to an OLAP service from Instance Manager, then the service environment may not be operational. Stop (if necessary) and restart it, following the instructions in “Managing the Service Environment” on page 3-2.

Instance Manager writes messages to a log file. These messages should help you identify the problem. For information on this log file, refer to “Viewing Status Messages” on page 3-18.

Related information

For information about exiting, search Instance Manager Help for the following topics:

“Exiting OLAP Services Instance Manager”
“Logging Into and Out of a Service”

To learn about performing these tasks in a command-line or batch mode, refer to “Command-line Administration” on page 3-23. In OLAP Service Manager, get help on the following options: `-b`, `-p`, `-u`.

Starting OLAP Services

About service modes

OLAP Services operates either in manual or automatic mode. It is set initially to run in manual mode so that it only starts when you explicitly restart it. However, you can set OLAP Services to run in automatic mode, so that it restarts automatically whenever you restart your computer or the service environment.

Procedure: Starting an OLAP service

Follow these steps to start an OLAP service:

1. Open Instance Manager by taking the steps listed in “Procedure: Opening Instance Manager” on page 3-3.
2. Click the plus sign (+) to the left of the OLAP service that you want to start.

   A tree of options expands under the service name.
3. Choose **Instance.**

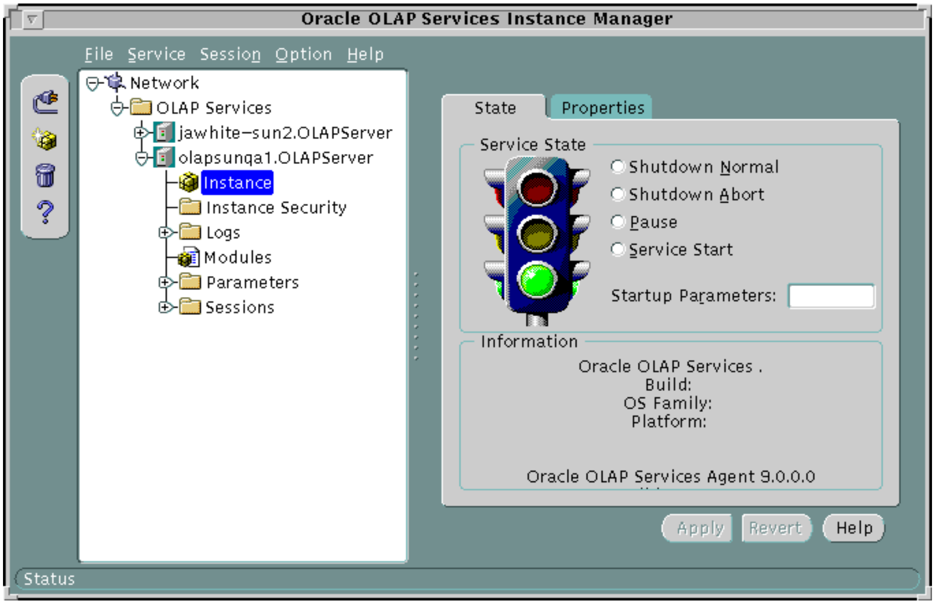
   You see the State page of the Instance sheet.

   Note: If you want to change the startup mode, then choose the Properties sheet.
4. Select **Service Start**, then choose **Apply.**

   Instance Manager will start the selected OLAP service.

Figure: Instance page

The following figure shows the State page of the Instance sheet:



Oracle OLAP Services .
Build:
OS Family:
Platform:

Oracle OLAP Services Agent 9.0.0.0

Problems starting an OLAP service

Upon installation, the environment variables and configuration settings used by OLAP Services during startup are set correctly. However, in the course of reconfiguring a service, installing other products, and so forth, you might make changes that will prevent OLAP Services from starting.

The following are some of the most common reasons why an OLAP service fails to start:

- The current primary Oracle home is not the Oracle9*i* home directory. You must change it back and restart the OLAP service environment. After starting OLAP Services, you can change the primary Oracle home as necessary.
- The value of TNS_ADMIN does not point to `$ORACLE_HOME/network/admin`.
- OLAP Services cannot authenticate itself through the RDBMS. The values of TNS_ALIAS, the proxy user name and password, or the DBA user and

password are incorrect. Test these values by using them in SQL*Worksheet or SQL*Plus to log into the RDBMS. If you cannot log in, then do the following:

  - Check for valid service names by using Net8 Assistant or by viewing the content of `tnsnames.ora` (located in the `network/admin` subdirectory of Oracle home).
  - Check that the proxy user and the DBA user identified in Instance Manager are defined in the database with the identical credentials. For more information, refer to “OLAP Services identities” on page 3-14.

- The OracleServiceName setting (located on the Security page under Parameters in Instance Manager) points to an invalid database alias. The value of OracleServiceName must match the value of TNS_ALIAS.
- The ServerDBPath setting (located on the I/O Management page under Parameters in Instance Manager) is invalid. Verify that the `olap/db` subdirectory is first in the list. Check the names of the other directories to make sure that they exist and are spelled correctly.
- The maximum number of connections have been made to the database. You might need to increase the connection limit.

OLAP Services sends messages to log files to help you diagnose any problems. You can access these log files through Instance Manager. For information about viewing these files, refer to “Viewing Status Messages” on page 3-18.

Problems connecting to a service

If OLAP API applications cannot connect to an instance of OLAP Services that you know is up and running, then the problem might be that an environment variable has been reset and no longer points to the needed directories. The correct paths are set by a file named `olap.prm` (located in the `olap` subdirectory of Oracle home), which runs during startup of the service environment. You can run this file manually or restart the service environment.

Some platforms require that the OSAgentAddr and OSAgentPort configuration parameters be set for the OLAP API to locate that OLAP service. Check the values of these parameters, and refer to Instance Manager Help for information about the appropriate setting for your platform.

Related information

For information about starting an OLAP service, search Instance Manager Help for the following topics:

"Customizing OLAP Service Initialization"
"Starting a Service"

To learn about performing these tasks in a command-line or batch mode, refer to "Command-line Administration" on page 3-23. In OLAP Service Manager, get help on the following commands: `listservices`, `start`, `status`, `setstartupmode`.

Stopping or Pausing OLAP Services

Warn users of service disruption

Pausing a service simply suspends any open sessions, but stopping a service does disconnect them. Users will appreciate your use of a system messaging facility to forewarn them of a system shutdown so that they can save their work.

Procedure: Stopping or pausing OLAP Services

Follow these steps to stop or pause an instance of OLAP Services:

1. Open Instance Manager by taking the steps listed in "Procedure: Opening Instance Manager" on page 3-3.
2. Click the plus sign (+) to the left of the OLAP service that you want to start.

   A tree of options expands under the service name.
3. Select **Instance.**

   You see the State page of the Instance sheet.
4. Select **Shutdown Normal**, **Shutdown Abort**, or **Pause**, then choose **Apply.**

   Instance Manager will change the state of the selected OLAP service.

   For a description of these options, choose **Help**.

Related information

For information about stopping an OLAP service, search Instance Manager Help for the following topics:

“Customizing OLAP Service Termination”
“Stopping a Service”

To learn about performing these tasks in a command-line or batch mode, refer to “Command-line Administration” on page 3-23. In OLAP Service Manager, get help on the following commands: `listservices`, `pause`, `stop`, `status`.

Acquiring System Administration Privileges

Three levels of privileges

Instance Manager recognizes three levels of access rights. All users must have a login ID with the parent Oracle database; those who cannot provide valid credentials have no access rights. The levels are:

- All RDBMS users. All users have the right to view some of the information that is available in Instance Manager, such as the configuration settings for an OLAP service.
- Instance Administration Privilege. Users can manage a particular OLAP service by changing configuration settings, terminating sessions, starting and stopping the service, and so forth.
- OLAP Service Administration Privilege. Users have the Instance Administration Privilege, plus the ability to control the service environment on the host machine. For example, a user with the OLAP Service Administration Privilege can create new OLAP services, remove existing services, and grant the OLAP Service Administration Privilege to other users.

The local repository stores the Instance Administration and OLAP Service Administration privilege information, along with all other configuration information. Passwords are encrypted.

Who can grant administration privileges?

The OLAP_DBA role has the highest level of privilege for administering OLAP Services. This role is managed by database security, in contrast to the administration privileges for OLAP Services, which are managed by Instance Manager.

The OLAP_DBA role is granted automatically to the system DBA role, and can be granted to other users through the Security node of Oracle Enterprise Manager. A user with the OLAP_DBA role in the database automatically has the OLAP Services Administration Privilege in Instance Manager.

Any user with the OLAP Services Administration Privilege can grant other users the OLAP Services Administration Privilege or the Instance Administration Privilege by using the Security page of Instance Manager.

The OLAP_DBA role is also required to create OLAP metadata, as described in "OLAP Services identities" on page 3-14.

Procedure: Granting administration privileges

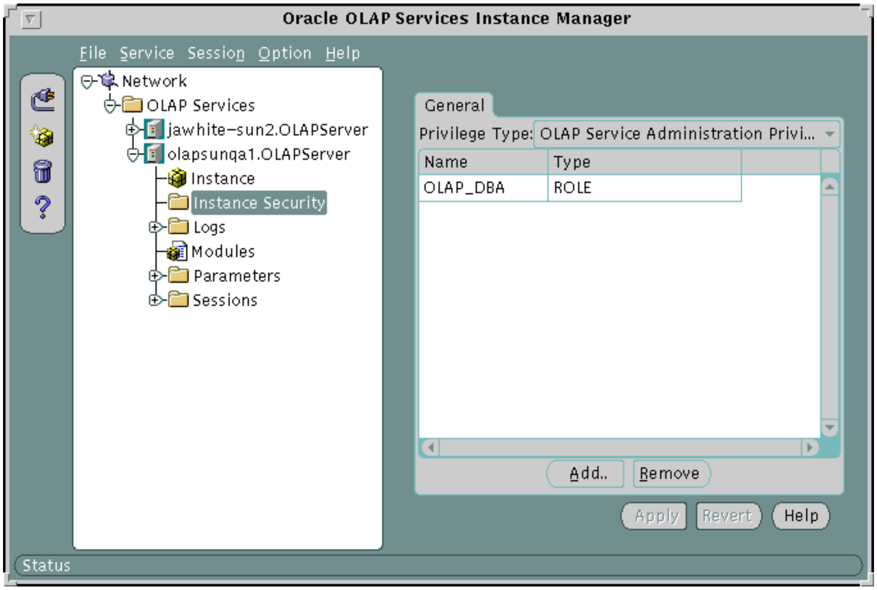
Follow these steps to grant Instance Administration or OLAP Services Administration privileges to a valid database user name or to a role. You will be able to select valid user names and roles from run-time lists fetched from the database into Instance Manager.

1. Open Instance Manager by taking the steps listed in "Procedure: Opening Instance Manager" on page 3-3. Log in with a user name that has the OLAP Services Administration Privilege, as described in "Who can grant administration privileges?" on page 3-9.
2. Select **Instance Security**.

   You see the General page for Instance Security.
3. Choose **Add** to display the Add Users/Roles to Privileges page.
4. Choose **Help** for further information about adding users to the list of those with administration privileges.

Figure: Instance Security sheet

In the following figure, the Instance Security sheet identifies the users who have been granted the OLAP Services Administration Privilege.



Privilege Type: OLAP Service Administration Privi...

Name	Type
OLAP_DBA	ROLE

Changing the Configuration Settings

What are configuration settings?

The configuration settings control and define the behavior of an OLAP service. All of the configuration parameters have default settings that are appropriate for most installations.

Which settings do I change?

Default values are provided for all required settings. You only need to change the settings that you have identified as inappropriate for your installation.

The following are some common reasons for changing the default settings:

- Change the password for the proxy and DBA users
- Change the character set that the service uses for internal text manipulations
- Improve performance
- Change the identifiers for additional instances of OLAP services
- Identify additional disk drives where the service can create temporary files

Procedure: Changing a configuration setting

Follow these steps to change a configuration setting:

1. Open Instance Manager by taking the steps listed in “Procedure: Opening Instance Manager” on page 3-3.
2. Click the plus icon next to **Parameters**.

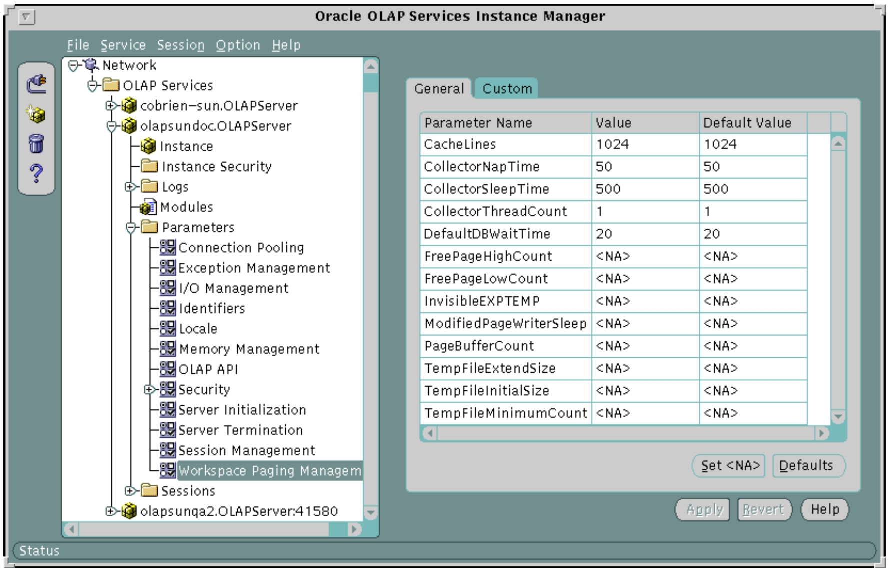
   You see the expanded list of categories.
3. Select a category.

   You see the sheet that lists the parameters grouped under that category.

   Tip: To find a specific parameter when you don’t know its category, search Help for that parameter by name.
4. Choose **Help** for further information.
5. Stop and restart the OLAP service so that the changes will take effect. Your changes have no effect on the currently running instance.

Figure: Parameter sheets

The following figure shows the expanded list of parameter sheets. The workspace paging parameters are currently displayed.



Parameter Name	Value	Default Value
CacheLines	1024	1024
CollectorNapTime	50	50
CollectorSleepTime	500	500
CollectorThreadCount	1	1
DefaultDBWaitTime	20	20
FreePageHighCount	<NA>	<NA>
FreePageLowCount	<NA>	<NA>
InvisibleEXPTEMP	<NA>	<NA>
ModifiedPageWriterSleep	<NA>	<NA>
PageBufferCount	<NA>	<NA>
TempFileExtendSize	<NA>	<NA>
TempFileInitialSize	<NA>	<NA>
TempFileMinimumCount	<NA>	<NA>

Related information

For further information about configuration settings, search Instance Manager Help for parameters by name, or for the following topics:

"Adding Configuration Parameters"
"Changing a Configuration Parameter"
"Removing Configuration Parameters"

Unlocking Database Identities

OLAP Services identities

To communicate with the database, OLAP Services must authenticate itself as a user. OLAP Services uses two identities, which are set up automatically by the installation process.

Database User Name	Default Password	Description	OLAP Services Configuration Parameters
OLAPSVR	INSTANCE	Proxy user	OlapProxyUID OlapProxyPwd
OLAPDBA	OLAPDBA	DBA user	UserID UserPassword

These user names and passwords are identified in two places: the database and the OLAP repository. When the OLAP service needs to authenticate itself with the database, it uses the credentials that it finds in the repository. If they match the appropriately configured credentials for a user in the database, then the OLAP service has the privileges it needs. It is therefore vital that the credentials in the OLAP repository match those in the database.

Important: This correspondence between an identity stored in the local repository and an identity defined in the database occurs automatically upon installation. However, you must unlock these identities in the database and change their passwords in both locations before OLAP Services can be used. Users will not be able to connect using the OLAP API until you have completed this step.

The proxy and DBA identities are defined in the database with different privileges and are used for different tasks by OLAP Services. They are not intended for use by client sessions.

Procedure: Changing the proxy and DBA passwords

Follow these steps to change the passwords associated with the proxy and DBA identities used by OLAP Services:

1. Open Oracle Enterprise Manager and connect to your database.
2. Expand the Security node in the navigator tree.

3. Under Users, select the OLAPSVR user to change the proxy password.

   or

   Select the OLAPDBA user to change the DBA password.

4. Change the password on the General page.
5. Choose **Apply**.
6. Open Instance Manager by right-clicking on the OLAP node.
7. To change the OLAPSVR password, select **Security** under the Parameters node. Then change the password on the General page.

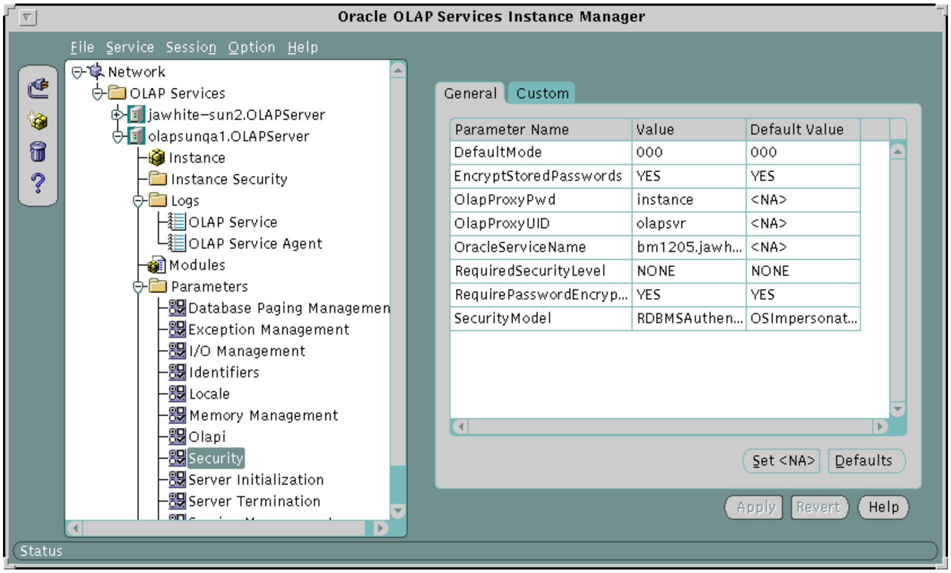
   or

   To change the OLAPDBA password, select **DBA** under the Security node. Then change the UserPassword on the General page.

8. Choose **Apply**.
9. Stop and restart the instance of OLAP Services, following the instructions in “Stopping or Pausing OLAP Services” on page 3-8 and “Starting OLAP Services” on page 3-5.

Figure: Proxy user parameters

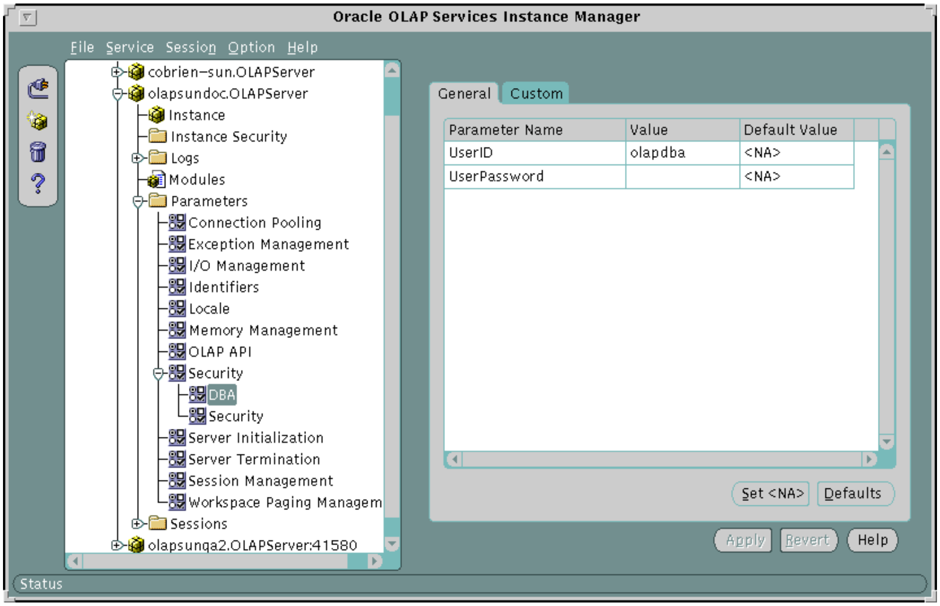
The following figure shows the OlapProxyUID and OlapProxyPwd settings on the Security sheet.



Parameter Name	Value	Default Value
DefaultMode	000	000
EncryptStoredPasswords	YES	YES
OlapProxyPwd	instance	<NA>
OlapProxyUID	olapsvr	<NA>
OracleServiceName	bm1205.jawh...	<NA>
RequiredSecurityLevel	NONE	NONE
RequirePasswordEncryp...	YES	YES
SecurityModel	RDBMSAuthen...	OSImpersonat...

Figure: DBA user parameters

The following figure shows the DBA user settings on the DBA sheet.



Parameter Name	Value	Default Value
UserID	olapdba	<NA>
UserPassword		<NA>

Managing Sessions

Who can terminate sessions?

Most user sessions end normally, when the user chooses to close the session. However, if a session hangs for any reason or runs a program that gets caught in an infinite loop, then you must force the session to end by using a tool such as Instance Manager. With administrative rights, you can end any user session. Users can also end their own sessions if they log into Instance Manager with the same credentials that they used to open their OLAP sessions.

Procedure: Terminating a session

To terminate a session, follow these steps.

1. Open Instance Manager by taking the steps listed in “Procedure: Opening Instance Manager” on page 3-3.
2. Click the plus icon next to **Sessions**.

   You see the list of active sessions.
3. Choose **Help** for additional information.

Related information

For more information, search for the following topics in Instance Manager Help:

“Terminating a Session”
“Viewing Session Information”

To learn about performing these tasks in a command-line or batch mode, refer to “Command-line Administration” on page 3-23. In OLAP Service Manager, get help on the following commands: `queryolapsessions`, `viewsessions`, `terminate`, `interrupt`.

Viewing Status Messages

About message logs

OLAP Services sends information to several log files to help you monitor operations and troubleshoot any problems.

You can view the following logs through Instance Manager:

- OLAP Service. Check this log if you are having trouble starting OLAP Services or if it stops unexpectedly. It contains a list of events with times and descriptions. The following I/O Management parameters affect this log:
  - EventLogPath. Identifies the path where `OLAPEvent.log` is written. Each OLAP service must have its own unique log path.
  - OverWriteEventLog. Controls whether `OLAPEvent.log` is cleared when a service restarts or whether it retains all of the information from startup to shutdown from previous executions of the service.

    - SeparateXCAEventLog. Controls whether XCA events are written to this log or a separate log file.

    The OLAP Service log is stored in a text file named `OLAPService.log`.

- OLAP Service Agent. This log records information that pertains to communications between the OLAP service and Instance Manager. It also contains information about the OLAP service environment, such as the startup and initialization of various daemons. This information is stored in a text file named `OLAPServicesAgent.log`

You can use a text editor or other utility to view the following additional logs:

- Instance Manager log. Check this log if you are having trouble running Instance Manager. Messages concerning startup and operation of Instance Manager are stored in a file named `osim.log` in the `log` directory of Oracle home.
- XCA log. Check this log if you are unable to connect to OLAP Services using OLAP Worksheet. Diagnostic information about XCA is stored in a file named `OES_XCA_Event.log` in the `olap/log` directory of Oracle home when the SeparateXCAEventLog parameter (located on the I/O Management sheet) is set to YES.

Procedure: Viewing a log in Instance Manager

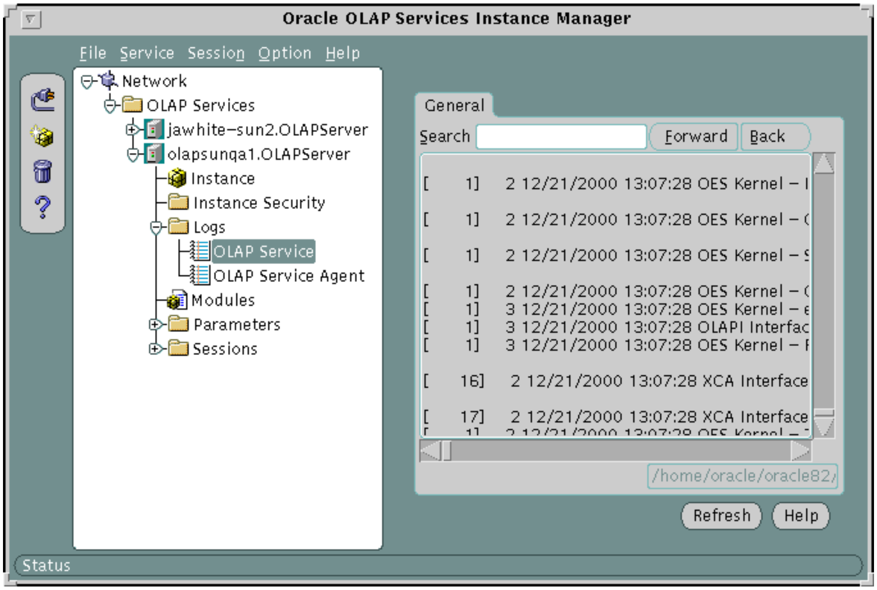
To view a log file through Instance Manager, follow these steps.

1. Open Instance Manager by taking the steps listed in “Procedure: Opening Instance Manager” on page 3-3.
2. Click the plus icon next to **Logs**.

   You see the list of log files.
3. Select the name of the log file you wish to view.

Figure: Logs sheet

In the following figure, the General page of the Logs sheet displays the OLAP Service log file.



```
[      1]    2 12/21/2000 13:07:28 OES Kernel – I

[      1]    2 12/21/2000 13:07:28 OES Kernel – (

[      1]    2 12/21/2000 13:07:28 OES Kernel – S

[      1]    2 12/21/2000 13:07:28 OES Kernel – (
[      1]    3 12/21/2000 13:07:28 OES Kernel – e
[      1]    3 12/21/2000 13:07:28 OLAPI Interfac
[      1]    3 12/21/2000 13:07:28 OES Kernel – F

[     16]    2 12/21/2000 13:07:28 XCA Interface

[     17]    2 12/21/2000 13:07:28 XCA Interface
[      1]    2 12/21/2000 13:07:28 OES Kernel – 
```

Related information

For more information about log files, search for these topics in Oracle Enterprise Manager Help:

“EventLogPath Setting”
“OverWriteEventLog Setting”
“Viewing Event Logs”

To learn about performing these tasks in a command-line or batch mode, refer to “Command-line Administration” on page 3-23. In OLAP Service Manager, get help on the following commands: `setlog`, `start`, `viewlog`.

Creating New Services

Why create additional services?

The installation process defines one service. You can modify the configuration of this service by changing the configuration settings. The configuration of a particular service is the same for every client session. Thus, if different clients need different configuration settings, then you need to provide them with different services.

Procedure: Creating a new service

Follow these steps to create a new service.

1. Open Instance Manager by taking the steps listed in “Procedure: Opening Instance Manager” on page 3-3.
2. Choose the Create icon.

   or

   From the Service menu, choose **Create Service Wizard.**

   You see Step 1 of the Create New Service wizard.
3. Provide the information requested.

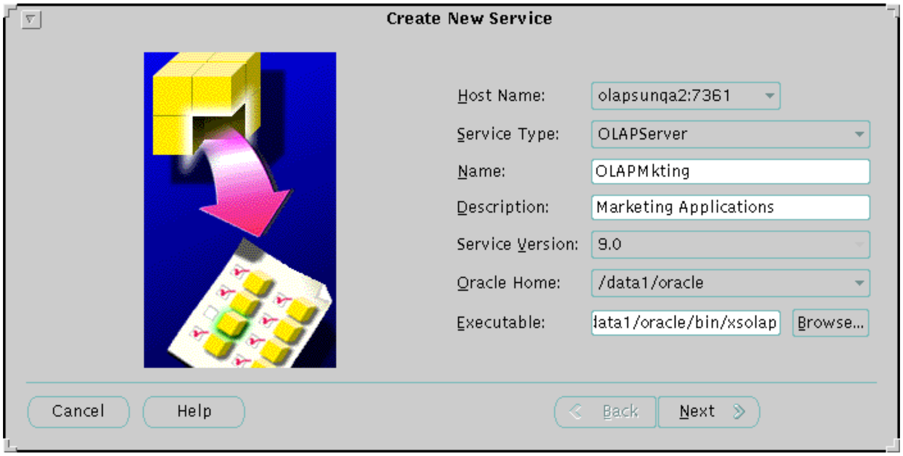
   Choose **Help** for a description of the required information.
4. Make whatever changes to the configuration settings that you wish.

   Although Instance Manager configures the new instance to require a minimum amount of additional configuration, you should check some settings for the new service:

   - ServerDBPath and other parameters on the I/O Management page identify file paths that should be set appropriately for the new service.
   - XCAPortNumber on the Identifiers page is set to a value that users of OLAP Worksheet will need to specify to make a connection.

Figure: Create New Service Wizard

The following figure shows the first page of the Create New Service wizard.



Create New Service

Host Name: olapsunqa2:7361

Service Type: OLAPServer

Name: OLAPMkting

Description: Marketing Applications

Service Version: 9.0

Oracle Home: /data1/oracle

Executable: lata1/oracle/bin/xsolap

Access to persistent analytic workspaces

If you are using persistent analytic workspaces in a MOLAP or HOLAP solution, then you must avoid conflicts among multiple instances of OLAP Services in accessing the same workspaces. Ideally, each instance is associated with its own applications so that no conflict arises. However, there are circumstances in which users of different instances may attempt to attach the same workspace.

In order for two separate instances of OLAP Services to have the same workspace open simultaneously, the workspace file must be flagged as read-only at the operating system level.

Particularly during development of an application, the code workspaces and certain system workspaces must be flagged read / write so that they can be updated. For this reason, they cannot be attached by multiple instances of OLAP Services simultaneously. Each instance must have its own copy of any workspaces that are not flagged as read-only at the operating system level, and the ServerDBPath variable for each instance should be modified to point only to its own workspaces.

Related information

For further information about creating new services, search Instance Manager Help for the following topic:

"Creating a Service"

Command-line Administration

About OLAP Service Manager

OLAP Service Manager is a command-line program that supports many of the same features that are available through Instance Manager. With OLAP Service Manager, you can do the following:

- View a list of OLAP services that are available on a given host
- Obtain the status of an OLAP service
- Start, stop, or pause an OLAP service
- View the event log for an OLAP service, and set the event log path
- View, interrupt, or stop individual sessions

You can run OLAP Service Manager interactively or as a batch command processor. By using it interactively, you can access its online Help facilities, which provide you with the information you need to formulate complete and syntactically correct commands. After you know the exact syntax of the commands you want to use, you can issue them in a batch environment. As an operating system command, OLAP Service Manager can be invoked from within scripts and from the Unix command line.

You can schedule and launch scripts that contain OLAP Service Manager commands using the job-scheduling facility within Oracle Enterprise Manager.

Setting the library path

OLAP Service Manager requires that the library path be set to include the `lib` subdirectory of Oracle home. You can run OLAP Service Manager using a script that sets the library path, or you can run it directly and assume responsibility for setting the library path yourself.

- `xscosvc` is the file name of OLAP Service Manager executable. If you run it directly, then you must first set the library path.
- `cosvc` is a script that sets the library path and then runs `xscosvc`. To run OLAP Service Manager in batch mode, you must use `cosvc` or create a similar script.

To set the library path for OLAP Service Manager, you set the appropriate environment variable to the `lib` subdirectory of Oracle home. The name of this variable varies among operating systems. To learn how to set it yourself, view the contents of `cosvc`.

Procedure: Running OLAP Service Manager

Follow these steps to run OLAP Service Manager interactively:

1. Access the command line.
2. Change to the `bin` subdirectory of Oracle home.
3. To see how to access the various types of Help, just type `cosvc`.

   The following text will be displayed.

   ```
   OLAP Service Manager, version: 9.0.0.0, build 72026
   Copyright (c) 1998-2000 Oracle Corporation. All Rights Reserved.

   Usage: xscosvc [options] [command1[; command2[; commandn]]]

   For additional help, refer to the commands below:
   xscosvc help options:    Display list of available options
   xscosvc help commands:   Display list of available commands
   xscosvc help <option>:   Display help on a particular option
   xscosvc help <command>:  Display help on a particular command
   ```

4. To run the utility interactively, use the `-s` and `-i` options.

   ```
   cosvc -s -i
   ```

   The utility displays a command prompt (–>).

5. Type in service-management commands. Enter one command per line, or delineate multiple commands with semicolons. Use the Help command to learn about the available commands and options.

As you can see from the Usage line in step 3, input to OLAP Service Manager consists of a series of options followed by one or more commands.

6. Type `quit` to return to the command line.

Common problems in starting OLAP Service Manager

If you are unable to run OLAP Service Manager, then you may be experiencing one of the following problems:

- Bad OLAP Agent object error. OLAP Services Agent, which is used by `xscosvc`, is not running on the target computer. Restart the service environment as described in “Procedure: Starting and stopping the service environment” on page 3-2.
- Missing file error. The value of the system library path environment variable is missing the `lib` subdirectory of Oracle home.

Syntax of the cosvc command

To manage OLAP services in batch mode, specify the full operation in a single command using `cosvc`. The input to OLAP Service Manager consists of a series of options followed by one or more commands.

cosvc [*option parameter*]... [*command parameter*]...

Example: Getting the status of an OLAP service

The following command queries the status of a local OLAP service named `OLAPServer`. The session is authenticated through the database with a user name of `scott` and a password of `tiger`. Depending on how you are running OLAP Service Manager, you may need to specify its full path name.

```
cosvc -u scott -p tiger status OLAPServer
```

The following information is sent to `stdout`.

```
OLAP Service Manager, version: 9.0.0.0, build: 72026.
Copyright (c) 1998-2000 Oracle Corporation.  All Rights Reserved.
Status of OLAPServer on olapsunprod:
    Pathname:       /home/oracle/bin/xsolap
    Description:    Oracle_OLAP_Server_9.0.0.0.0
    State:          running
    Startup mode:   automatic
    Event log path: /home/oracle/olap/OLAPServer/log
```

Redirecting output

OLAP Service Manager writes text output to `stdout` and `stderr`, which are typically defined as the screen. This implementation allows you to redirect output to a file, pipe it to another program, or capture and interpret output in a script.

If the output consists of multiple lines of text, then the end-of-line character will be represented in the native format. On Unix, the end-of-line character is `LF`.

All text output from OLAP Service Manager is in English US ASCII format. All text output returned from OLAP Services is in the character set used by that OLAP service.

Handling errors

OLAP Service Manager returns a status code that indicates whether or not it successfully completed execution. You can write a batch program that traps the status code and proceeds conditionally, based on the type of error, rather than simply terminating.

The error messages are sent to `stdout`, but you can redirect this output to a log file for later analysis.

Tip: Use the `-i` option if you want OLAP Service Manager to ignore errors rather than terminating.

The following table identifies the error codes.

Status Code	Status Message
0	Processing completed successfully
101	Specified service does not exist
102	Command timed out
103	Bad command or help request
104	Cannot get current host name
105	Connect to service environment failed
106	Generic “command failed”
107	Syntax error or unsupported command
108	No network present

Granting Access Rights to Users

About access rights

To open a session in OLAP Services, a user must present a user name and password that can be authenticated in the database. The privileges associated with that user name control the user's access to data. A user who cannot present valid credentials cannot open a session. As a database administrator, you must set up user names with appropriate credentials for all users of OLAP Services applications.

Minimum rights

Users of OLAP applications must have the following access rights to the database:

- CONNECT role
- QUERY REWRITE system privilege
- CONNECT THROUGH *OlapProxyUID* proxy
- SELECT privileges on the database objects containing the data to be analyzed

You can define user names and grant them these rights through the Security node of Oracle Enterprise Manager. *OlapProxyUID* is an OLAP Services configuration setting. Its default value is `OLAPSVR`.

Configuring Instance Manager

How to set Instance Manager properties

Instance Manager has several properties that affects the way it runs. You can create a text file named `osim.properties` in the `olap` subdirectory that lists the properties you wish to change. Using a text editor, enter only one setting on a line and use an equal sign (=) to separate the property name from its value.

Property descriptions

The following table describes the various properties that affect the way that Instance Manager runs.

Property	Value
GeneralTimeOut=	The number of milliseconds allowed for any general operation, including user interface operations, performed in Instance Manager before timing out. The default value is 40000 ms (40 seconds). Only numeric values are acceptable. Any non-numeric characters will invalidate the setting.
SocketTimeOut=	The number of milliseconds allowed for any socket operation performed in Instance Manager before timing out. The default value is 20000 ms (20 seconds). Only numeric values are acceptable. Any non-numeric characters will invalidate the setting.
DebugToLogFile=	A value of true or false, which controls whether debug statements are written to osim.log.
DebugLevel=	The following keywords control the level of detail of debug statements written to osim.log when DebugToLogFile=true: ▪ PROD. Production ▪ DBG. Debugging ▪ VDBG. Verbose debugging

Example: Instance Manager configuration file

The following is a sample osim.properties configuration file.

```
#Thu Nov 09 14:55:39 EST 2000
DebugToLogFile=false
SocketTimeOut=20000
DebugLevel=VDBG
GeneralTimeOut=40000
```

Configuring OLAP Services Agent

What is OLAP Services Agent?

OLAP Services Agent is the back end for the Instance Manager graphical user interface. It actually performs the tasks, such as starting and stopping an OLAP service or changing its configuration settings. It also returns information from the operating system, such as the success or failure of a startup attempt.

Startup parameters for OLAP Services Agent

You can define a system variable named OES_EA_STARTUP and provide settings that alter the information that OLAP Services Agent provides to Instance Manager. OLAP Services Agent runs on the same computer as OLAP Services, so be sure to define OES_EA_STARTUP on that computer. These settings affect the information provided to all instances of Instance Manager, regardless of where they are running or in what mode.

For information about defining and setting system variables, refer to the documentation for your operating system.

Setting	Description
`-LogOutput=`*`level`*	Controls the amount of information provided in the logs to the following levels: ■ `PROD`. Production ■ `DBG`. Debugging ■ `VDBG`. Verbose debugging

How to change the port number

OLAP Services Agent uses 7051 as its default port number. You can change this port assignment when another process is using this port.

To specify a different port number, edit `olap` and search for the following line:

```
$OH/bin/xsdaemon $@
```

Add the `-e` *`portno`* parameter so the line looks like the following example, in which 6666 is the new port number:

```
$OLAP_HOME/bin/xsdaemon -e6666 $@
```

4

Configuring Your Data Warehouse

Chapter summary

This chapter describes the structures in a data warehouse that are required by the OLAP metadata. It explains how to map these relational structures to multidimensional OLAP objects so that applications built using the OLAP API can access the data.

List of topics

This chapter includes the following topics:

- Your Data Warehouse
- Transforming Tables into Multidimensional Data Structures
- Database Requirements
- Creating Metadata
- Defining Dimensions
- Defining Cubes
- Viewing Cubes
- Defining Measure Folders
- Modifying the Metadata
- Creating Materialized Views

Your Data Warehouse

What is a data warehouse?

The term *data warehouse* is used to distinguish a relational database that is used for business analysis (OLAP) rather than transaction processing (OLTP). While an OLTP database contains current low-level data and is typically optimized for the selection and retrieval of records, a data warehouse typically contains aggregated historical data and is optimized for particular types of analyses, depending upon the client applications.

Creating a data warehouse

This guide assumes that you have generated a data warehouse using an Extraction Transformation Transport (ETT) tool such as Oracle Warehouse Builder. If you do not currently have a data warehouse and wish to build one, then refer to the *Oracle9i Data Warehousing Guide*. It provides in-depth discussions about all of the database objects and concepts related to data warehouse design.

This chapter highlights some of the most important concepts developed in the *Oracle9i Data Warehousing Guide*. It also provides information that is specific to a data warehouse that will be used with OLAP Services. However, this chapter does not provide sufficient information for you to build a data warehouse of your own, or even to fully understand the issues involved in creating and maintaining a data warehouse.

Contents of a data warehouse

The contents of your data warehouse depends on the requirements of your users. They should be able to tell you what type of data they want to view and at what levels of aggregation they want to be able to view it.

Your data warehouse will store these types of data:

- Historical data
- Derived data
- Metadata

Historical data

A data warehouse typically contains several years of historical data. The amount of data that you decide to make available depends on available disk space and the types of analysis that you want to support. This data can come from your transactional database archives or other sources.

The lowest level of data is stored in *fact tables* in your data warehouse. The lowest level in a data warehouse is typically at a much higher level than in the transactional database. The transactional data should be aggregated to a base level where patterns and trends can emerge and analysis is meaningful, before being stored in the data warehouse. For example, individual purchase orders might be aggregated by sales representative, zip code, or some other demographic feature.

Some applications might perform analyses that require data at lower levels than users typically view it. You will need to check with the application builder or the application's documentation for those types of data requirements.

Derived data

Derived data is generated from existing data using a mathematical operation or a data transformation. OLAP Services uses SQL ROLLUP to generate aggregate data in the data warehouse. *Dimension tables*, also called *lookup tables*, are used to store the dimension members for all levels in the hierarchy. A dimension object in the database stores additional information about the hierarchy so that it can be aggregated further within the data warehouse.

By using dimension hierarchies, applications allow users to recognize trends at one level of aggregation, drill down to lower levels to identify reasons for these trends, and roll up to higher levels to see what affect these trends have on a larger sector of the business. For example, in a time dimension, days might roll up into months, and months into years.

Oracle9*i* provides *materialized views* for storing precomputed data derived from fact tables. Materialized views significantly improve querying times because the aggregates are computed for everyone's use, as a database administration task when the data is refreshed, rather than recomputed repeatedly whenever the aggregates are needed.

Metadata

Metadata is data that describes the data and schema objects, and is used by applications to fetch and compute the data correctly. The metadata informs the OLAP API about the data that is available within the database so that it can define multidimensional objects in OLAP Services. When an application runs, it instantiates these objects and populates them with data fetched from the database.

You must use Oracle Enterprise Manager to create the metadata required by the OLAP API, as described in "Creating Metadata" on page 4-10.

About schemas

A schema is a collection of relational database objects. Two types of schemas are characteristic of a data warehouse:

- Star schema. Consists of one or more fact tables related to one or more dimension tables. The relationships are defined through foreign keys and metadata.
- Snowflake schema. A star schema that has been partially or fully normalized to reduce the number of duplicate values in the dimension tables.

For example, a star schema might have a single Geography dimension table with four columns: City, State, Region, and Country. Only the City column has predominately unique values, while the other columns have increasing numbers of duplicate values.

A snowflake schema might have three related geography dimension tables: One table with two columns (City and State) that define the relationship between cities and states, a second table with two columns (State and Country) that define the relationship between states and countries, and a third table with two columns (State and Country) that define the relationship between states and countries.

Your data warehouse can have either a star or a snowflake schema.

Transforming Tables into Multidimensional Data Structures

Differences in data models

When an OLAP application runs, OLAP Services fetches the required data from an Oracle9*i* database into a temporary cache. Within this cache, the data is stored in multidimensional data objects. For the data to be fetched correctly, you must identify which columns will be fetched and what role they will play.

The basic data model in a relational database is a table composed of one or more columns of data. All of the data is stored in columns. In contrast, the basic data model in a multidimensional cache is a cube, which is composed of measures, dimensions, and attributes. Once you determine that you will want to access a particular column through OLAP Services, you must identify whether the data from that column will function in it as a measure, a dimension, or an attribute. You also identify which columns are keys. These decisions are stored as metadata and constraints.

Types of data structures

The data warehouse built in your Oracle RDBMS and OLAP Services use different data structures. Note that even though both use dimensions, their implementations are different, as described in “Dimensions” on page 4-6.

A data warehouse has the following data structures:

- Tables
- Materialized Views
- Dimensions

OLAP Services has these data structures:

- Levels
- Attributes
- Dimensions
- Measures
- Cubes

Identifying your data requirements

Before you can begin mapping columns to multidimensional structures, you must know what data users want to view and at what levels they want to view it. If you have already created a data warehouse, then you have already done most of this research. You only need to verify that the requirements haven’t changed for the analytical applications that will be run using OLAP Services.

You can use Oracle Enterprise Manager to explore the existing schemas. Then make a note of the columns that you are going to use and the types of multidimensional objects you want to define them as: measures, dimensions, or attributes. Keep in mind that the OLAP API only has access to objects in the database through the metadata definitions. Thus, if an object (such as a column in a table) has not been defined in the metadata, then it is not available to OLAP applications.

Measures

Measures are the same as facts. The term “fact” is typically used in relational databases, and the term “measure” is typically used in multidimensional applications. You will encounter both terms in Oracle Enterprise Manager, since the creation of metadata is the process of associating relational objects with their multidimensional counterparts.

Measures are thus located in fact tables. A fact table typically has two types of columns: measures (or facts) and foreign keys to dimension tables.

Measures contain the data that you wish to analyze, such as Sales or Cost. Oracle Enterprise Manager requires that a column have a numerical or date data type to be identified as a measure. Most frequently, a measure is numerical and additive.

One or more columns in the dimension tables form constraints on the fact tables. These constraints are defined by foreign keys in the fact tables, by the metadata, or both.

Dimensions

Dimensions identify and categorize your data. Dimension members are stored in a dimension table. Each column represents a particular level in a hierarchy. In a star schema, the columns are all in the same table; in a snowflake schema, the columns are in separate tables for each level.

Because measures are typically multidimensional, a single value in a measure must be qualified by a member of each dimension to be meaningful. For example, a Sales measure might have dimensions for Product, Geographic Area, and Time. A value in the Sales measure (37854) is only meaningful when it is qualified by a product (DVD Player), a geographic area (Pacific Rim), and Time (March 2001).

Defining a dimension in your data warehouse creates a database dimension object, in addition to creating metadata. A dimension object contains the details of the parent-child relationship between columns in a dimension table; it does not contain data. The database dimension object is used by the Summary Advisor and query rewrite to optimize your data warehouse.

However, in the OLAP API, a dimension does contain data, such as the names of individual products, geographic areas, and time periods. The OLAP API uses the metadata, dimension objects, and dimension tables to construct its dimensions.

Hierarchies and levels

A *hierarchy* is a way to organize data according to levels. Dimensions are structured hierarchically so that data at different levels of aggregation can be manipulated together efficiently for analysis and display. Each dimension must have at least one level.

Each *level* represents a position in the hierarchy. Levels group the data for aggregation and are used internally for computation. Each level above the base (or lowest) level represents the aggregate total of the levels below it. For example, a Time dimension might have Day, Week, Quarter, and Year for the levels of a Time dimension hierarchy. If data for the Sales measure is stored in days, then the higher levels of the Time dimension allow the Sales data to be aggregated correctly into weeks, quarters, and years.

The members of a hierarchy at different levels have a one-to-many *parent-child* relationship. For example, “QTR1” and “QTR2” are the children of “YR2001,” thus “YR2001” is the parent of “QTR1” and “QTR2”.

If you define more than one hierarchy for a dimension, then the hierarchies must have the same base level. For example, you might define two hierarchies for your Time dimension, one for the calendar year and another for the fiscal year. Both hierarchies would use Day for the base level.

All levels of a dimension are stored in dimension tables. A dimension can have multiple hierarchies, but all of them must have the same base level. The values of that level are stored in the key used to join the dimension table to a fact table.

Attributes

Attributes provide supplementary information about the dimension members at a particular level. Attributes are often used for display, since the dimension members themselves may be meaningless, such as a value of “T296” for a time period.

For example, you might have columns for employee number (ENUM), last name (LAST_NAME), first name (FIRST_NAME), and telephone extension (TELNO). ENUM is the best choice for a level, since it is a key column and its values uniquely identify the employees. ENUM also has a NUMBER data type, which makes it more efficient than a text column for the creation of indexes. LAST_NAME, FIRST_NAME, and TELNO are attributes. Even though they are dimensioned by ENUM, they do not make suitable measures because they are descriptive text rather than business measurements.

Attributes are associated with a particular level of a dimension hierarchy and must be stored in the same table as that level.

Database Requirements

Summary check list

The following list summarizes the database requirements:

- ❏ You must have created a data warehouse with a star or snowflake schema.
- ❏ Each dimension must have at least one level.
- ❏ Attributes must be stored in the same dimension table as the level with which they are associated.
- ❏ The Time dimension table must be fully populated and contain end-date and time-span columns, as described in “Time dimensions” on page 4-8.
- ❏ Facts must be stored in columns defined with a data type supported by OLAP Services.
- ❏ All measures in a fact table must be dimensioned identically. You can store similarly dimensioned measures in different fact tables, however, each fact table must be referenced by a different cube. A cube references a single fact table.
- ❏ All hierarchies for a dimension must have the same base level.

Time dimensions

OLAP metadata considers Time dimensions as distinct from other dimensions. When you specify a dimension in OLAP management, you must identify whether it is a Time dimension. A Time dimension has special attributes that support both regular and irregular time periods.

Regular time periods, such as weeks, months, and years, are evident on standard calendars. Typically, they neither overlap nor have gaps between them. *Irregular time periods*, such as promotional schedules and seasonal time periods, are not evident on standard calendars. They often overlap (even to the extent that one time period is a subset of another time period) or have gaps between them.

The Time dimension table must contain the following columns:

- Values for all dimension members, with a column for each level of summarization (such as weeks, quarters, and years).
- An end-date attribute for each level, such as WEEK_ENDDATE, QUARTER_ENDDATE, and YEAR_ENDDATE. These columns must have a DATE data type. Their values identify the last day in the time period.

- A time-span attribute for each level, such as WEEK_TIMESPAN, QUARTER_TIMESPAN, and YEAR_TIMESPAN. These columns must have a NUMBER data type. Their values identify the number of days in the period.

Example: Time dimension in a star schema

The following table describes an example dimension table in a star schema.

Column Name	Sample Value	Data Type	Comment
WEEK_ID	W12000	VARCHAR2	Level 1
WEEK_DESC	Week Ending January 8, 2000	VARCHAR2	Attribute
WEEK_ENDDATE	8-JAN-00	DATE	Attribute
WEEK_TIMESPAN	7	NUMBER	Attribute
QUARTER_ID	1QTR2000	VARCHAR2	Level 2
QUARTER_DESC	1st Quarter in Year 2000	VARCHAR2	Attribute
QUARTER_ENDDATE	31-MAR-00	DATE	Attribute
QUARTER_TIMESPAN	91	NUMBER	Attribute
YEAR_ID	YR2000	VARCHAR2	Level 3
YEAR_DESC	Year 2000	VARCHAR2	Attribute
YEAR_ENDDATE	31-DEC-00	DATE	Attribute
YEAR_TIMESPAN	366	NUMBER	Attribute

Example: Time dimension in a snowflake schema

The following tables describe example dimension tables in a snowflake schema. The first table defines weeks, which is the lowest level of time data.

Column Name	Sample Value	Data Type	Comment
WEEK_ID	W12000	VARCHAR2	Level 1
WEEK_DESC	Week Ending January 8, 2000	VARCHAR2	Attribute
WEEK_ENDDATE	8-JAN-00	DATE	Attribute
WEEK_TIMESPAN	7	NUMBER	Attribute

A second table defines quarters.

Column Name	Sample Value	Data Type	Comment
WEEK_ID	W12000	VARCHAR2	Foreign key
QUARTER_ID	1QTR2000	VARCHAR2	Level 2
QUARTER_DESC	1st Quarter in Year 2000	VARCHAR2	Attribute
QUARTER_ENDDATE	31-MAR-00	DATE	Attribute
QUARTER_TIMESPAN	91	NUMBER	Attribute

A third table defines years.

Column Name	Sample Value	Data Type	Comment
QUARTER_ID	1QTR2000	VARCHAR2	Foreign key
YEAR_ID	YR2000	VARCHAR2	Level 3
YEAR_DESC	Year 2000	VARCHAR2	Attribute
YEAR_ENDDATE	31-DEC-00	DATE	Attribute
YEAR_TIMESPAN	366	NUMBER	Attribute

Supported data types

The OLAP API supports native Java data types. It does not support the following Oracle data types: BLOB, CLOB, NCLOB, RAW, and LONG RAW. Do not create measures from facts with these unsupported data types.

Note: The OLAP DML supports CLOB and NCLOB data types. If your application requires this type of data, then you should investigate a MOLAP or HOLAP solution. Search for “SQL (FETCH)” in OLAP DML Help for additional information about supported data types.

Creating Metadata

Using Oracle Enterprise Manager

Creating OLAP metadata is one of several data administration tasks that you can perform through Oracle Enterprise Manager. Using its OLAP management tool, you create the metadata that defines the multidimensional objects required by the

OLAP Services. Oracle Enterprise Manager enables you to map columns in your star or snowflake schema to dimensions, measures, and attributes. Even if you have other types of metadata in your database to support other types of applications, you must create OLAP metadata for OLAP Services.

You generate the SQL statements that create the metadata primarily by following the steps presented by a wizard or by completing a property sheet. If you wish, you can display the SQL statements before executing them.

Note: You may have already created the required metadata with an ETT tool as part of creating your data warehouse. If so, you can check and modify this metadata using OLAP management in Oracle Enterprise Manager.

Software prerequisites

To run OLAP management, you must have installed Oracle Enterprise Manager Client.

If Oracle Enterprise Manager Client is installed on the same computer with Oracle9*i* and OLAP Services, then a service name has already been defined as part of the installation. Otherwise, you may need to define a service name using Net8 Assistant so that Oracle Enterprise Manager Client can open a connection with the database.

Access rights

To create OLAP metadata, you must be able to log into your database through Oracle Enterprise Manager with credentials that have been granted the OLAP_DBA role. The OLAP_DBA role is necessary to create metadata through Oracle Enterprise Manager as described in this chapter, and to administer OLAP Services as described in Chapter 3.

The OLAP_DBA role has system privileges associated with it, such as the ability to create and drop tables, indexes, and dimensions. For a list of these privileges, follow these steps:

1. Log into your database through Oracle Enterprise Manager.
2. Expand the Security branch.
3. Choose **OLAP_DBA**.
4. Display the Role and System Privileges pages.

If you have the system DBA role, then you also have the OLAP_DBA role. You must also have the CONNECT role.

Note: To view existing metadata, you only need the CONNECT and SELECT_CATALOG_ROLE roles.

Procedure: Accessing OLAP management

Follow these steps to start Oracle Enterprise Manager and access OLAP management:

1. Open Oracle Enterprise Manager console.

   You see the main page.

2. Expand **Databases** by clicking the plus sign next to it.

   You see the list of service names for Oracle databases on your network.

3. Expand the database that you want to manage.

   You see the Database Connect Information dialog box.

4. Type in your user name (one with the appropriate credentials) and password for that database.

   Tip: Select the **Save as preferred credentials** box if you wish to eliminate this step in future sessions. Your user name and encrypted password will be saved in a local file. For security, make sure that *only you* can run Oracle Enterprise Manager with your stored credentials. Later, if you wish to change this information, then choose **Edit Local Preferred Credentials** from the Configuration menu.

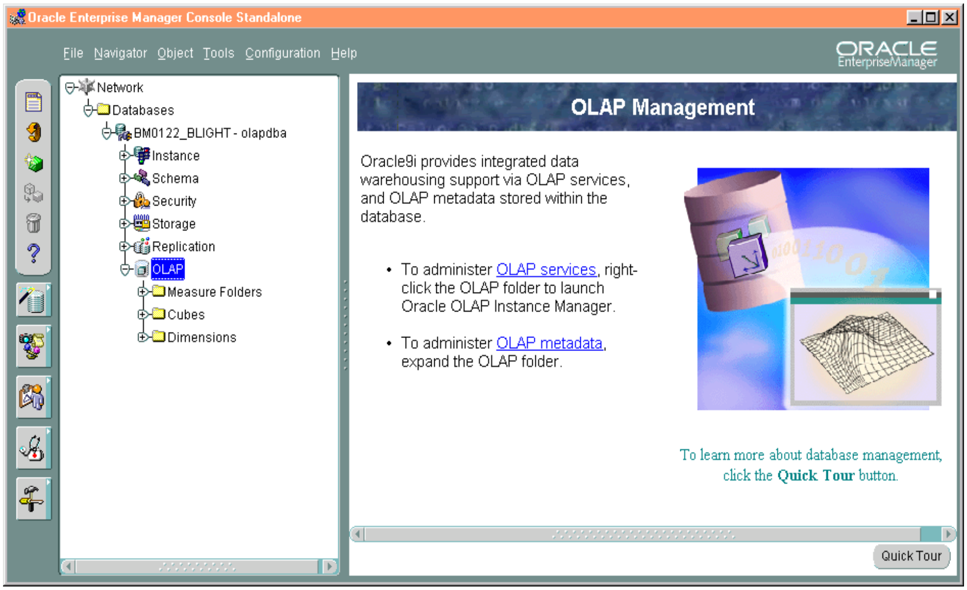
   The tree view will expand to show the various tools available for administering the database.

5. Expand **Olap**.

   You see the types of objects that you can create, as shown in “Figure: Location of OLAP management” on page 4-13. This part of the Oracle Enterprise Manager tree is for OLAP management.

Figure: Location of OLAP management

The following figure identifies the location of OLAP management in Oracle Enterprise Manager.



Oracle9i provides integrated data warehousing support via OLAP services, and OLAP metadata stored within the database.

- To administer OLAP services, right-click the OLAP folder to launch Oracle OLAP Instance Manager.
- To administer OLAP metadata, expand the OLAP folder.

To learn more about database management, click the **Quick Tour** button.

Defining Dimensions

Dimensions, hierarchies, and attributes

Before you can define any other objects, you must define the dimensions of your data. You can use the Dimension Creation Wizard or supply information directly in the Create Dimension dialog box.

To define a dimension, you provide all the information that will be needed to label and aggregate the measures dimensioned by it, including:

- The name of the dimension
- The tables that contain the data for the dimension
- The name of each level, and the columns that contain the data for each level
- The number and order of levels in each hierarchy

- Join keys for levels that are stored in separate tables
- The columns that contain attributes for the levels
- A display name and description for the dimension and each of its hierarchies, levels, and attributes

Time dimensions

Business analysis is performed on historical data, so fully defined time periods are vital. Special support for time dimensions is built into the metadata to allow for time-dependent analyses, such as comparisons with earlier time periods.

Your time dimension table must have columns for end-date and time-span, as described in “Time dimensions” on page 4-8. Typical levels and hierarchies for time dimensions are suggested by the Dimension Wizard, but you do not have to use them.

Procedure: Defining dimensions

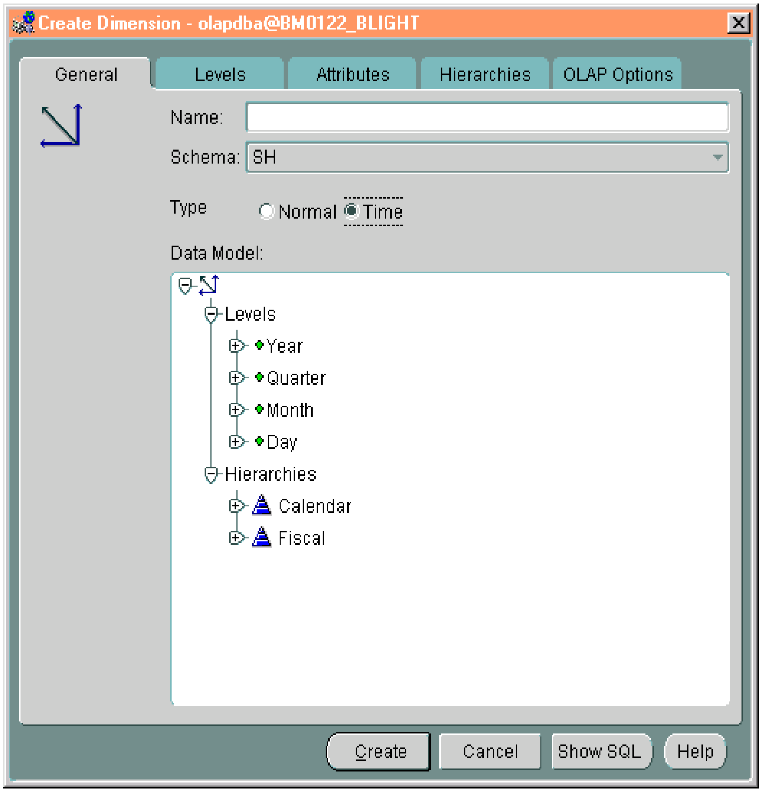
Follow these steps to create a dimension and its associated levels, hierarchies, and attributes:

1. Start Oracle Enterprise Manager and access OLAP management, as described in “Procedure: Accessing OLAP management” on page 4-12.
2. To create a new dimension, right click on **Dimensions**, then choose one of the following:
   - **Create Using Wizard** to run the Dimension Wizard

     or
   - **Create** to edit a new dimension property sheet
3. Choose **Help** if you need additional information.

Figure: Dimension property sheet

The following figure shows the General page of the dimension property sheet for a Time dimension.



Name:

Schema: SH

Type ○ Normal ● Time

Data Model:

Related information

Search for the following topics in Oracle Enterprise Manager Help for information about creating dimensions:

“Creating Dimensions”
“Dimension Property Sheet”
“OLAP Dimension Management”
“What Is a Dimension?”

Defining Cubes

Cubes and measures

A cube is a collection of identically dimensioned measures. When you define a cube, you identify information such as the following:

- The name of the cube and the fact table associated with it. All measures in a cube must be from a single fact table.
- The names of the dimensions and the levels in the dimension hierarchies that will be used in the cube.
- The names of the measures and the columns in the fact table where the values for each measure is stored.
- Default aggregation operators for each dimension of each measure (such as sum or average).
- Any calculation dependencies.

Cubes and measures are defined entirely in the OLAP metadata; there are no corresponding database objects. In OLAP Services, measures are created in which to cache the data for analysis and display.

Procedure: Defining cubes

Follow these steps to create a cube:

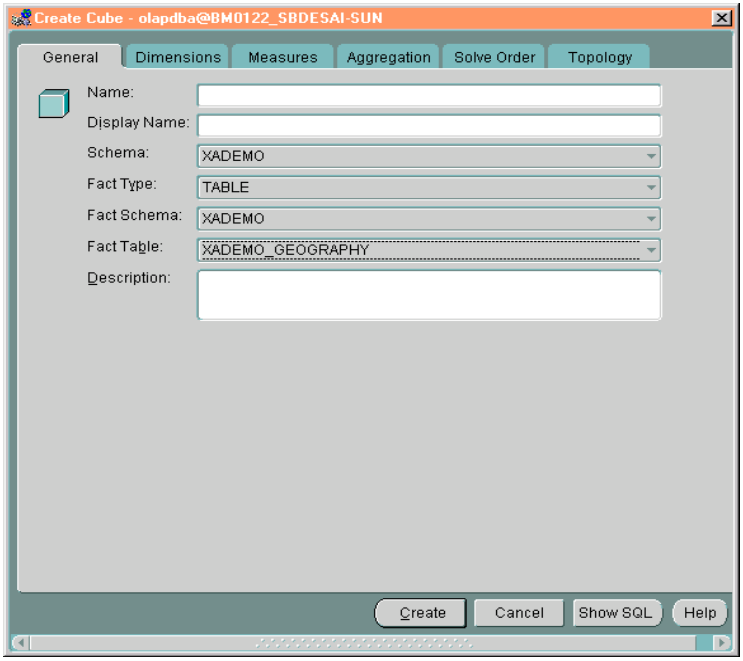
1. Start Oracle Enterprise Manager and access OLAP management, as described in “Procedure: Accessing OLAP management” on page 4-12.
2. Right-click on **Cubes**, then choose one of the following:
    - **Create Using Wizard** to run the Cube Wizard

      or

    - **Create** to edit a new cube property sheet
3. Choose **Help** if you need additional information.

Figure: Cube property sheet

The following figure shows the General page of the cube property sheet.



Create Cube - olapdba@BM0122_SBDESAI-SUN

Name:

Display Name:

Schema: XADEMO

Fact Type: TABLE

Fact Schema: XADEMO

Fact Table: XADEMO_GEOGRAPHY

Description:

Related information

Search for the following topics in Oracle Enterprise Manager Help for information about creating cubes:

“Creating Cubes”
“Cube Management”
“Cube Property Sheet”
“What Is a Cube?”

Viewing Cubes

Cube viewer

The cube viewer allows you to see the cube that you created in the same way that end-users might see it — with the data presented in a BI Beans crosstab, as described in “Crosstabs” on page 2-7. Moreover, you can select the data that you want to see by using the query builder.

The cube viewer uses Oracle Naming Service to locate an instance of OLAP Services.

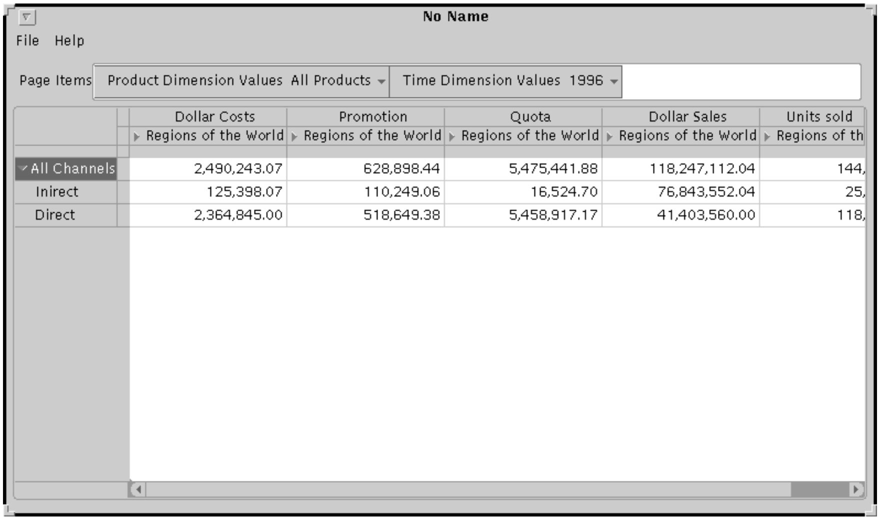
Procedure: Viewing cubes

Follow these steps to view a cube:

1. Start Oracle Enterprise Manager and access OLAP management, as described in “Procedure: Accessing OLAP management” on page 4-12.
2. Expand the OLAP tree so that you can see the list of cubes.
3. Right-click on the cube you want to examine, then choose **Cube Viewer**.
4. If you need additional information, then search for the Help topic “Viewing a Cube’s Data.”

Figure: A data cube

The following figure shows the Sales cube in the cube viewer.



Page Items: Product Dimension Values All Products; Time Dimension Values 1996

	Dollar Costs	Promotion	Quota	Dollar Sales	Units sold
	Regions of the World	Regions of the World	Regions of the World	Regions of the World	Regions of th
All Channels	2,490,243.07	628,898.44	5,475,441.88	118,247,112.04	144,
Inirect	125,398.07	110,249.06	16,524.70	76,843,552.04	25,
Direct	2,364,845.00	518,649.38	5,458,917.17	41,403,560.00	118,

Defining Measure Folders

Organizing data

Measure folders provide a means of grouping measures related to a particular business activity, such as finance or sales. These measures will typically be used together in applications tailored to the needs of a particular user group, such as marketing managers or financial advisors. Applications restrict their view of the database to a particular measure folder rather than a particular database schema. The measures in a folder can be associated with any number of different schemas.

Measure folders exist exclusively in the metadata. There are no corresponding objects in the data warehouse. They do not belong to any schema, so anyone can view them. However, only the creator of a measure folder (or those with administrative privileges) can modify or delete it.

Procedure: Creating measure folders

To create a measure folder, follow these steps:

1. Start Oracle Enterprise Manager and access OLAP management, as described in "Procedure: Accessing OLAP management" on page 4-12.
2. Right-click on **Measure Folders**, then choose **Create**.

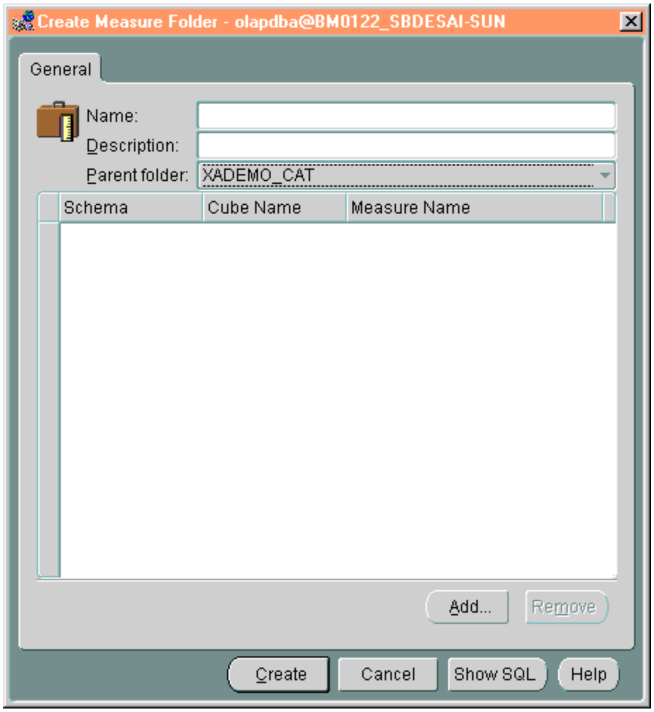
   or

   If you want to create a measure folder with many of the same measures as an existing one, then right-click on the existing folder and choose **Create Like**.

   You see the measure folder property sheet. If you chose **Create Like**, then the measures from the original folder will already be listed.
3. Choose **Help** if you need additional information.

Figure: Measure folder property sheet

The following figure shows the measure folder property sheet.



Create Measure Folder - olapdba@BM0122_SBDESAI-SUN

Name:

Description:

Parent folder: XADEMO_CAT

Related information

Search for the following topics in Oracle Enterprise Manager Help for further information about measure folders:

“Creating Measure Folders”
“Measure Folder Management”
“Measure Folder Property Sheet”
“What Is a Measure Folder?”

Modifying the Metadata

Locating metadata objects

After you create a metadata object, such as a dimension or cube, you can access that object from the OLAP management tree. Dimensions are grouped by schema under **Dimensions**, cubes are grouped by schema under **Cubes**, and measure folders are optionally grouped under parent folders.

You can view the properties of a metadata object, change those properties, or delete the object.

Procedure: Modifying your metadata

Follow these steps to modify a metadata object:

1. If necessary, start Oracle Enterprise Manager and access OLAP management, as described in “Procedure: Accessing OLAP management” on page 4-12.
2. Expand the OLAP tree until you see the object you want to modify, as shown in “Figure: Expanded OLAP tree” on page 4-23. For example, to modify a dimension, expand **Dimensions**, then expand the appropriate schema.
3. To simply view the properties of an object, select it on the tree. Its property sheet will appear in the right pane of the main window.

   or

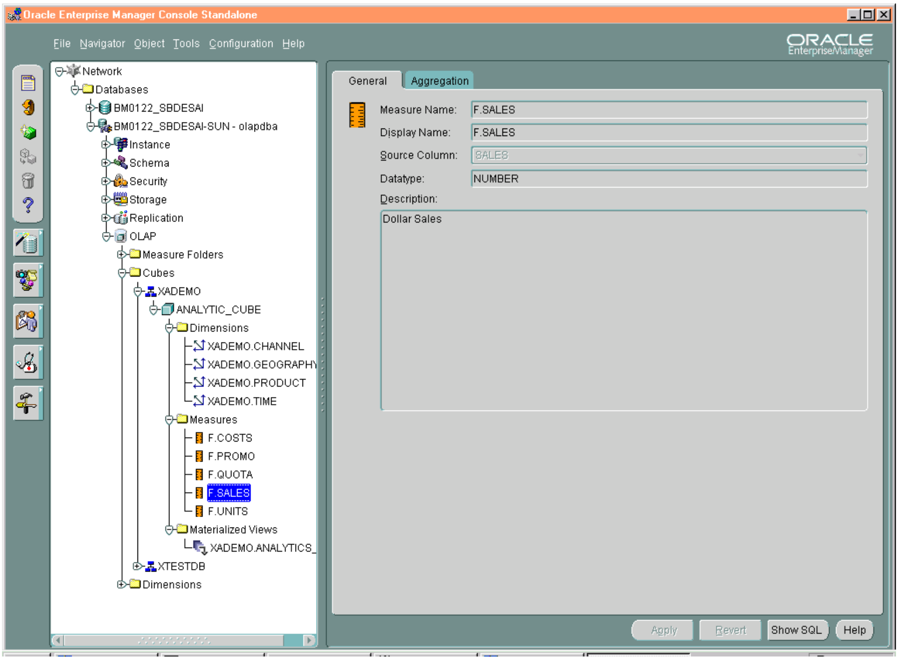
   To edit an object, right-click the object, then choose **Edit**. The property sheet for that object will appear in the Edit dialog box.

   or

   To delete an object, right-click it, then choose **Remove**. You will be prompted to confirm that you want to remove the object.

Figure: Expanded OLAP tree

The following figure shows **Cubes** expanded to show the dimensions and measures in a cube. The property sheet of the selected measure appears in the right pane.



Measure Name: F.SALES

Display Name: F.SALES

Source Column: SALES

Datatype: NUMBER

Description:

Dollar Sales

Related information

For more information about making changes to your metadata, search Oracle Enterprise Manager Help for the following topics:

“Editing Cubes”
“Editing Dimensions”

Creating Materialized Views

Storing aggregate data

Materialized views store data that is calculated from fact tables. When data in the master tables changes, you can refresh materialized views with the new data.

While a *view* only stores the query, a *materialized view* actually stores the results of a query. Thus, there is a trade-off between improved performance and increased storage requirements. Materialized views can greatly improve performance for some types of queries, but can require many additional gigabytes of disk space.

To balance query speed and hardware resources, you should create materialized views for the most frequently calculated aggregates. Less frequently calculated aggregates can continue to be calculated on an as-needed basis. The Summary Advisor Wizard will help you identify the materialized views that your schema should have, within the constraints imposed by the resources you are willing to commit to this use.

Query rewrite

The Oracle9*i* database's query rewrite capability allows it to use materialized views for queries written against fact tables. However, for query rewrite to recognize that the materialized views contain the query results, they must be created using basically the same type of SQL commands that OLAP Services will generate. Oracle Enterprise Manager generates two types of materialized views: those created specifically for use by analytical applications developed using the OLAP API, and those optimized for transactional queries. You must be sure to create materialized views for use by the OLAP API, by following the steps in "Procedure: Running the Summary Advisor wizard" on page 4-25 and "Procedure: Creating materialized views" on page 4-26.

Summary Advisor prerequisites

The Summary Advisor requires that you have already defined dimensions and measures in your metadata. You must also have collected usage data on which Summary Advisor can base its recommendations. Because usage patterns can change, you should rerun Summary Advisor periodically, particularly when the metadata changes or a new application is distributed to users.

Search for the topic "Summary Advisor Prerequisites" in Oracle Enterprise Manager Help for more specific information.

Procedure: Running the Summary Advisor wizard

Follow these steps to run the Summary Advisor wizard:

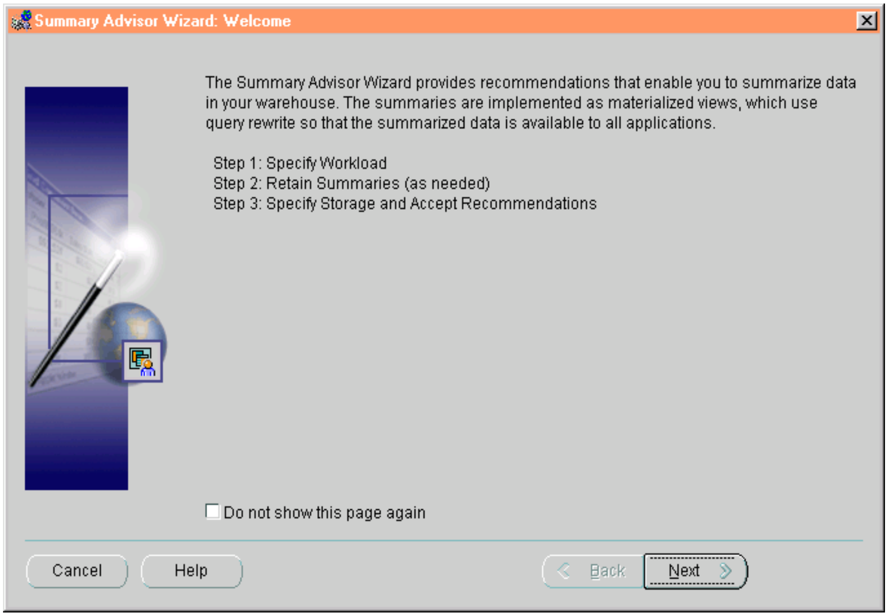
1. If necessary, start Oracle Enterprise Manager and access OLAP management, as described in "Procedure: Accessing OLAP management" on page 4-12.
2. Expand the OLAP tree so that you can see the list of defined cubes.
3. Expand the cube whose usage data you want to analyze.
4. Right-click **Materialized Views**, then choose **Summary Advisor** from the cascading menu.

   You see the first page of the Summary Advisor wizard.
5. Choose **Help** if you need more information while running the wizard.

Note: You can also run the Summary Advisor wizard from the final page of the Create Cube wizard.

Figure: Summary Advisor wizard

The opening page of the Summary Advisor wizard lists the steps you will follow.



Summary Advisor Wizard: Welcome

The Summary Advisor Wizard provides recommendations that enable you to summarize data in your warehouse. The summaries are implemented as materialized views, which use query rewrite so that the summarized data is available to all applications.

Step 1: Specify Workload
Step 2: Retain Summaries (as needed)
Step 3: Specify Storage and Accept Recommendations

☐ Do not show this page again

Procedure: Creating materialized views

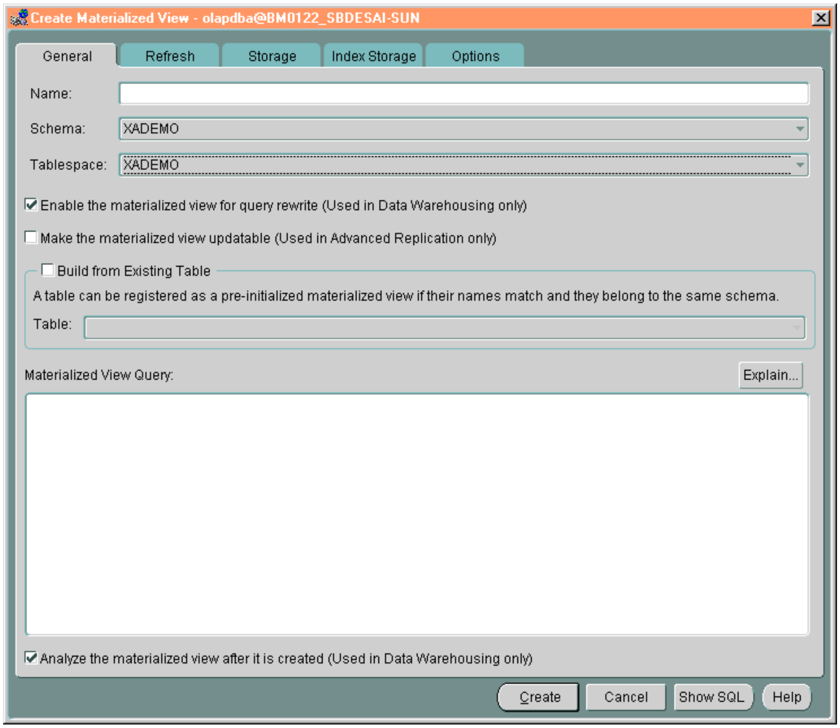
Follow these steps to create a materialized view for a cube that you previously created:

1. If necessary, start Oracle Enterprise Manager and access OLAP management, as described in “Procedure: Accessing OLAP management” on page 4-12.
2. Expand the OLAP tree so that you can see the list of defined cubes.
3. Expand the cube whose data you want to aggregate in a materialized view.
4. Right-click **Materialized Views**, then choose **Create** from the cascading menu.

   You see the Materialized View property sheet.
5. Choose **Help** if you need more information while completing each page of the property sheet.

Figure: Materialized view property sheet

The following figure shows the General page of the materialized view property sheet.



Create Materialized View - olapdba@BM0122_SBDESAI-SUN

Name:

Schema: XADEMO

Tablespace: XADEMO

☑ Enable the materialized view for query rewrite (Used in Data Warehousing only)

□ Make the materialized view updatable (Used in Advanced Replication only)

□ Build from Existing Table

A table can be registered as a pre-initialized materialized view if their names match and they belong to the same schema.

Table:

Materialized View Query:

☑ Analyze the materialized view after it is created (Used in Data Warehousing only)

Related information

For more information about the Summary Management wizard, search Oracle Enterprise Manager for “summary management.”

For more information about creating materialized views, search Oracle Enterprise Manager for “materialized views.”

5

Localization

Chapter summary

This chapter explains how to configure OLAP Services so that it can accurately and quickly process text data in native languages.

List of topics

This chapter includes the following topics:

- Introduction
- About Oracle Globalization Technology
- Understanding Encoding Schemes
- Introduction to Character Sets
- Choosing an Instance Character Set
- Identifying the Locale
- Supported Languages and Territories
- Supported Character Sets

Introduction

Handling text data in the languages of the world

OLAP Services, like the RDBMS, has been engineered so that it can be deployed internationally with a minimum of customization required on site. As the system administrator, you need to identify the smallest available character set that is able to represent all of your data, the language preferred by most of your users, and the geographic region where most of them live. With this information, OLAP Services is able to read, write, and manipulate data from a variety of sources without a loss of that data, and it is able to present the data in ways that are appropriate to its users.

The ability to operate gracefully in multiple languages and cultures thus has two aspects: the internal encoding of the data, and the external display of the data.

Internationalization and localization

OLAP Services has been *internationalized* so that it can be deployed in various languages and regions without requiring that the program code be modified and recompiled. For example, status messages are stored separately instead of being embedded in the program code. These error messages are provided in all of the supported languages, and the appropriate ones are determined by the configuration settings.

Localization is the process of adapting to the user's linguistic, cultural, and geographic environment. The most basic is the *character set*, which specifies the available characters as well as how the data is manipulated internally. Two properties modify the basic specification of the character set:

- *Language*. Indicates the terms that should be used for "Yes" and "No," the days of the week, the months of the year, and the language used to display error messages. It also identifies the appropriate sorting sequences.
- *Territory*. Identifies conventions such as calendar, date formats, local currency symbols, and numeric formats that vary by geographic region.

When the localization parameters are set correctly, OLAP Services can communicate with client applications, external files and data sources, and the operating system, even though they might use a variety of different character sets and support users throughout the world.

About Oracle Globalization Technology

What is Oracle Globalization Technology?

Oracle Globalization Technology (formerly called the National Language System or NLS) provides the Oracle standard for internationalizing and localizing Oracle products. Its use in the database and OLAP Services allows text data in native languages to be passed between them without data loss or performance degradation.

Differences between the RDBMS and OLAP Services

A few differences exist in the implementation of Globalization Technology in OLAP Services and the RDBMS, as shown in the following table.

Database	OLAP Services
Supports both ASCII and non-ASCII character sets.	Supports only those character sets that are strict ASCII super sets.
Stores Globalization Technology settings in a file named `init.ora`.	Stores Globalization Technology settings in the OLAP Services registry.
Supports definition of a national character set that is used for all columns with type NCHAR or NVARCHAR. The national character set can be either UTF-8 (variable width) or UTF-16 (fixed width).	NCHAR and NVARCHAR data from the database is converted to the instance character set when fetched. OLAP Services does not provide equivalent data types because it does very little of the kind of text processing that makes these data types desirable.

Related information

For complete information about Globalization Technology, refer to the *Oracle9i Globalization Technology Guide*.

Understanding Encoding Schemes

About encoded character sets

Computers handle character data as numeric codes instead of their graphical representations. A group of characters and symbols that are typically used together can be mapped to numeric codes. This set of numeric codes is an encoded character set. The earliest character sets were ASCII and EBCDIC, which were two different ways of encoding the letters A-Z and a to z, the numbers 0 to 9, and various punctuation and other symbols commonly used in the English language.

For example, the letter “A” is represented in ASCII as 0x41, and the letter “a” is represented as 0x61. Although many encodings retain the basic ASCII codes, a different character set could use 0x41 and 0x61 to represent mathematical symbols, or escape sequences, or two unrelated letters of a different alphabet. Without knowing which encoding is in use, you cannot know for sure what character is represented by a particular number.

To make their products available beyond the English-speaking market, hardware and software manufacturers developed their own encodings for other languages and alphabets. In the short term, they were successful in bringing their products to an international market. In the long term, these individual initiatives created a hodgepodge of hundreds of encodings that make accurate communications across hardware platforms and software products a challenge.

What is Unicode?

Unicode is an emerging standard for encoding text in all of the written languages of the world. It provides a unique number for every character, regardless of language, program, or platform. Products that use Unicode, such as the Oracle RDBMS and OLAP Services, can be deployed across multiple platforms, languages, and countries without re-engineering, and can handle text in multiple languages without data loss.

Use of UTF-8 encoding

Oracle supports Unicode using the UTF-8 standard, which is a format that transforms all Unicode characters into a variable-length encoding of bytes. One Unicode character can be one, two, or three bytes in this encoding. Surrogate pairs will require four bytes, but there are no such characters defined in the current version of the standard.

All ASCII characters are valid UTF-8 characters, so that existing English ASCII data does not require conversion.

Unicode also has a UTF-16 encoding, which is used by Java and Windows NT. Java and Windows NT are used for many client applications. Conversion is easy between UTF-8 (used by OLAP Services) and UTF-16 (used by client applications), and so does not noticeably impact performance.

Related information

Additional information about the Unicode standard is available in the *Oracle9i Globalization and National Language Support Guide* and at the Unicode Consortium Web site at `www.unicode.org`.

Introduction to Character Sets

Conversion requirements among components

For a single user session, character sets are used in several different areas of the software:

- Instance character set
- Client character set
- Analytic workspace character set

Moreover, you will need to import data from the RDBMS or from a text file. When the user makes a query, data is transferred among these different parts of the system. In the best scenario, the same character set is used throughout these various parts. In other scenarios, the use of different character sets might have no effect, or might render the data partially or fully undecipherable, or might slow I/O time.

Instance character set

The instance character set is the one used internally by OLAP Services to manipulate and process text. Because OLAP Services performs whatever character conversions are required to communicate with its clients and its data sources, you must choose the instance character set with care. This information is provided in “Choosing an Instance Character Set” on page 5-6.

Analytic workspace character set

The analytic workspace character set is used to encode the data stored on disk. When the workspace is created, it is assigned the same character set as the instance of OLAP Services.

If you are creating or using a persistent analytic workspace, note that the workspace character set cannot be changed. All of the text data in a workspace must be in the workspace character set.

To attach a workspace, the workspace character set must be the same as the instance character set or US7ASCII, which has the same encoding as all available instance character sets. A US7ASCII workspace must be attached read-only unless the instance character set is also US7ASCII.

To update a workspace, the workspace character set must be the same as the instance character set.

Client character set

The client character set is used by a client application to represent text. Applications developed using the OLAP API or the BI Beans are Java applications and always use UTF-16 Unicode, as described in “Use of UTF-8 encoding” on page 5-4. These applications identify the appropriate language and territory settings for their client sessions.

Choosing an Instance Character Set

Assessing your requirements

Before you can select the appropriate instance character set, you must assess the requirements for that particular instance. You must take into account the natural language and geographic location of your users and the character sets used by existing data sources and text data. The character set that you choose must be complete enough to represent all the data that users will want to store or manipulate, whether it is originating in the database, an external file, or a persistent analytic workspace. However, it should not be more complete than necessary.

Default character set: UTF-8

UTF-8 is a variable-width character set encoding that can represent all of the characters included in the Unicode standard, including graphical characters for Japanese, Korean, Chinese, English, Greek, Hebrew, and Russian, using 1, 2, or 3 bytes for each character.

Performance considerations

Processing data stored in UTF-8 is slower than processing data stored in either a single-byte character set or a smaller, fixed width, multi-byte character set.

You can improve processing time very quickly and easily just by changing the character set used by OLAP Services. However, if your Oracle database stores data from around the world, or your OLAP applications access NCHAR or NVARCHAR columns, then OLAP Services will need to use UTF-8.

Rules for choosing a character set

You need to identify a supported character set that is extensive enough to represent all of the data accessed by your OLAP applications, yet no larger than necessary. The basic rule is this:

The OLAP Services instance character set should be the same as the RDBMS database character set *whenever possible.*

In the following cases, OLAP Services might not have the same character set as the database:

- If OLAP applications are going to access any NCHAR or NVARCHAR columns in the database, then specify `UTF8`.
- If data will be read from external files or spreadsheets, and the data cannot be represented by the database character set, then specify `UTF8`. Note that this data cannot be stored in the database because of incompatibility between the data sets. It can only be stored in the analytic workspace or an external file.

  However, in these cases, the database character set can be used:

  - The external files and spreadsheets use the same character set as the database.
  - The external files or spreadsheets use a different character set than the database, but all characters can be represented in the database character set (that is, the character set of the file is a subset of the character set of the database.) For example, if the database uses Japanese EUC, and the files use Japanese Shift-JIS, the OLAP Services should use Japanese EUC.
- If the database uses a non-ASCII character set, then use the ASCII equivalent if possible. For example, if the database character set is `WE8EBCDIC37` (8-bit EDBDIC, Western European code page 37), then specify `WE8ISO8859P1` (8-bit ISO Latin-1). If there is no ASCII equivalent, then specify `UTF8`.

To specify the character set, set the NLS_LANG configuration parameter, described in “Specifying the instance character set” on page 5-8.

What happens when different character sets are used?

The following table indicates what happens when data moves from one part of the software to another, such as between OLAP Services and a client application or between a text file and OLAP Services.

IF the character set of a client or external file . . .	THEN . . .
uses the same numeric codes to represent the same graphic characters as the instance character set,	no translation is required. This is the optimal situation since no processing time is required to convert the data.
uses the same range of numeric codes but uses them to represent different graphic characters,	OLAP Services must know the identity of the other character set so that it can map the numeric codes to the correct graphic characters. No data is lost, but some processing time is required to convert the data from one character set to another.
contains numeric codes for graphic characters that are not defined in the instance character set,	data may be lost because it cannot be represented in OLAP Services. The instance character set must be a super set of the character sets used by other parts of the system.

Identifying the Locale

What is the locale?

The locale is the character set, language, and territory. The format for specifying this environment is *language_territory.character set*. For a list of valid keywords, refer to the following:

- “Supported Languages and Territories” on page 5-9
- “Supported Character Sets” on page 5-11

To specify the default values that apply to all sessions of a particular instance of OLAP Services, set the OLAP Services locale configuration parameters. Users can override the language and territory settings for their sessions.

Specifying the instance character set

NLS_LANG is a configuration parameter that identifies the locale for an instance of OLAP Services. The character set specified in NLS_LANG is the instance character set, and thus the database character set for all databases created or upgraded using this instance. To set NLS_LANG, follow the guidelines in “Choosing an Instance Character Set” on page 5-6.

The language and territory components are much less important since they are set on a session-by-session basis by client applications.

The NLS_CHARSET_OS parameter identifies the character set of the operating system when it is different from the character set identified in NLS_LANG. This setting allows OLAP Services to interact with the operating system to locate user files and persistent workspaces.

Be sure to test the new configuration by viewing text data in OLAP Services, preferably through the client applications that this instance is intended to support. If you see meaningless characters (such as boxes or question marks), then your choice of character set is incorrect.

Rules determining the instance character set

The instance character set for a session is determined by the following rules:

1. NLS_LANG parameter setting, if it has a supported value
2. WE8ISO8859P1 (ISO Latin I) if available
3. US7ASCII (7-bit ASCII)

Related information

For more information about configuration parameters, search for them by name in Instance Manager Help.

Supported Languages and Territories

Introduction to languages and territories

Java client applications identify the appropriate language and territory that should be used for a particular session. These settings can vary from session to session. For the initial setting, choose the language and territory that are appropriate for most of your users.

Languages

The following list identifies the supported languages.

AMERICAN
ARABIC
BENGALI
BRAZILIAN PORTUGESE
BULGARIAN
CANADIAN FRENCH
CATALAN
CROATIAN
CZECH
DANISH
DUTCH
EGYPTIAN
ENGLISH
ESTONIAN
FINNISH
FRENCH
GERMAN DIN
GERMAN
GREEK
HEBREW
HUNGARIAN
ICELANDIC
INDONESIAN
ITALIAN
JAPANESE
KOREAN
LATIN AMERICAN SPANISH
LATVIAN
LITHUANIAN
MALAY
MEXICAN SPANISH
NORWEGIAN
POLISH
PORTUGESE
ROMANIAN
RUSSIAN
SIMPLIFIED CHINESE
SLOVAK
SLOVENIAN
SPANISH
SWEDISH
THAI
TRADITIONAL CHINESE
TURKISH
UKRAINIAN
VIETNAMESE

Territories

The following list identifies the supported territories.

ALGERIA
AMERICA
AUSTRALIA
AUSTRIA
BAHRAIN
BANGLADESH
BELGIUM
BRAZIL
BULGARIA
CANADA
CATALONIA
CHINA
CIS
CROATIA
CYPRUS
CZECH
CZECHOSLOVAKIA
DENMARK
DJIBOUTI
EGYPT
ESTONIA
FINLAND
FRANCE
GERMANY

GREECE
HONG KONG
HUNGARY
ICELAND
INDONESIA
IRAQ
IRELAND
ISRAEL
ITALY
JAPAN
JORDAN
KAZAKHSTAN
KUWAIT
LATVIA
LEBANON
LIBYA
KOREA
LITHUANIA
LUXEMBOURG
MALAYSIA
MAURITANIA
MEXICO
MOROCCO
NEW ZEALAND
NORWAY
OMAN
POLAND
PORTUGAL
QATAR
ROMANIA
SAUDI ARABIA
SINGAPORE
SLOVAKIA
SLOVENIA
SOMALIA
SOUTH AFRICA
SPAIN
SUDAN
SWEDEN
SWITZERLAND
SYRIA
TAIWAN
THAILAND
THE NETHERLANDS
TUNISIA
TURKEY
UKRAINE
UNITED ARAB EMIRATES
UNITED KINGDOM
UZBEKISTAN
VIETNAM
YEMEN

Supported Character Sets

Introduction to character sets

The supported character sets are listed in three tables that represent broad languages groups:

- "Asian language character sets" on page 5-12
- "European language character sets" on page 5-13
- "Middle Eastern language character sets" on page 5-16

Some character sets are listed in more than one group because they provide multilingual support.

All supported character sets are strict super sets of ASCII.

Abbreviations used in the comments

The following symbols are used in the Comments column of the tables:

EURO	Supports the Euro currency symbol
MB	Multi-byte encoding
SB	Single-byte encoding
UDC	Supports user-defined characters

Asian language character sets

The following table identifies the available character sets that support Asian languages.

Character Set Value	Description	Comments
BN8BSCII	Bangladesh National Code 8-bit ASCII	SB
ZHT16BIG5	BIG5 16-bit Traditional Chinese	MB
ZHS16CGB231280	CGB2312-80 16-bit Simplified Chinese	MB
JA16EUC	EUC 24-bit Japanese	MB
ZHT32EUC	EUC 32-bit Traditional Chinese	MB
ZHS16GBK	GBK 16-bit Simplified Chinese	MB, UDC
ZHT16CCDC	HP CCDC 16-bit Traditional Chinese	MB
KO16KSC5601	KSC5601 16-bit Korean	MB
KO16KSCCS	KSCCS 16-bit Korean	MB
JA16VMS	JVMS 16-bit Japanese	MB
TH8MACTHAIS	Mac Server 8-bit Latin/Thai	SB
ZHT16MSWIN950	MS Windows Code Page 950 Traditional Chinese	MB, UDC
KO16MSWIN949	MS Windows Code Page 949 Korean	MB, UDC
VN8MSWIN1258	MS Windows Code Page 1258 8-bit Vietnamese	SB, EURO
IN8ISCII	Multiple-Script Indian Standard 8-bit Latin/Indian Languages	SB
JA16SJIS	Shift-JIS 16-bit Japanese	MB, UDC

Character Set Value	Description	Comments
ZHT32SOPS	SOPS 32-bit Traditional Chinese	MB
ZHT16DBT	Taiwan Taxation 16-bit Traditional Chinese	MB
TH8TISASCII	Thai Industrial Standard 620-2533 - ASCII 8-bit	SB, EURO
ZHT32TRIS	TRIS 32-bit Traditional Chinese	MB
UTF8	Unicode 3.0 UTF-8 Universal character set	MB, EURO
VN8VN3	VN3 8-bit Vietnamese	SB

European language character sets

The following table identifies the available character sets that support European languages.

Character Set Value	Description	Comments
US7ASCII	ASCII 7-bit American	SB
RU8BESTA	BESTA 8-bit Latin / Cyrillic	SB
TR8DEC	DEC 8-bit Turkish	SB
TR8PC857	IBM-PC Code Page 857 8-bit Turkish	SB
TR8MACTURKISHS	MAC Server 8-bit Turkish	SB
TR8MSWIN1254	MS Windows Code Page 1254 8-bit Turkish	SB, EURO
WE8DEC	DEC 8-bit West European	SB
WE8DG	DG 8-bit West European	SB
LA8PASSPORT	German Government Printer 8-bit All-European Latin	SB
WE8ROMAN8	HP Roman8 8-bit West European	SB
HU8CWI2	Hungarian 8-bit CWI-2	SB
HU8ABMOD	Hungarian 8-bit Special AB Mod	SB
LV8RST104090	IBM-PC Alternative Code Page 8-bit Latvian (Latin / Cyrillic)	SB
US8PC437	IBM-PC Code Page 437 8-bit American	SB
BG8PC437S	IBM-PC Code Page 437 8-bit (Bulgarian Modification)	SB

Character Set Value	Description	Comments
EL8PC437S	IBM-PC Code Page 437 8-bit (Greek modification)	SB
LT8PC772	IBM-PC Code Page 772 8-bit Lithuanian (Latin / Cyrillic)	SB
LT8PC774	IBM-PC Code Page 774 8-bit Lithuanian (Latin)	SB
BLT8PC775	IBM-PC Code Page 775 8-bit Baltic	SB
WE8PC850	IBM-PC Code Page 850 8-bit West European	SB
EL8PC851	IBM-PC Code Page 851 8-bit Greek / Latin	SB
EE8PC852	IBM-PC Code Page 852 8-bit East European	SB
RU8PC855	IBM-PC Code Page 855 8-bit Latin / Cyrillic	SB
WE8PC858	IBM-PC Code Page 858 8-bit West European	SB, EURO
WE8PC860	IBM-PC Code Page 860 8-bit West European	SB
IS8PC861	IBM-PC Code Page 861 8-bit Icelandic	SB
CDN8PC863	IBM-PC Code Page 863 8-bit Canadian French	SB
N8PC865	IBM-PC Code Page 865 8-bit Norwegian	SB
RU8PC866	IBM-PC Code Page 866 8-bit Latin / Cyrillic	SB
EL8PC869	IBM-PC Code Page 869 8-bit Greek / Latin	SB
LV8PC1117	IBM-PC Code Page 1117 8-bit Latvian	SB
WE8ISO8859P1	ISO 8859-1 West European	SB
EE8ISO8859P2	ISO 8859-2 East European	SB
SE8ISO8859P3	ISO 8859-3 South European	SB
NEE8ISO8859P4	ISO 8859-4 North and North-East European	SB
CL8ISO8859P5	ISO 8859-5 Latin / Cyrillic	SB
AR8ISO8859P6	ISO 8859-6 Latin / Arabic	SB
EL8ISO8859P7	ISO 8859-7 Latin / Greek	SB
IW8ISO8859P8	ISO 8859-8 Latin / Hebrew	SB
NE8ISO8859P10	ISO 8859-10 North European	SB
WE8ISO8859P15	ISO 8859-15 West European	SB, EURO

Character Set Value	Description	Comments
LA8ISO6937	ISO 6937 8-bit Coded Character Set for Text Communication	SB
AR8ARABICMACS	Mac Server 8-bit Latin / Arabic	SB
EE8MACCES	Mac Server 8-bit Central European	SB
EE8MACCROATIANS	Mac Server 8-bit Croatian	SB
WE8MACROMAN8S	Mac Server 8-bit Extended Roman8 West European	SB
CL8MACCYRILLICS	Mac Server 8-bit Latin / Cyrillic	SB
EL8MACGREEKS	Mac Server 8-bit Greek	SB
BG8MSWIN	MS Windows 8-bit Bulgarian Cyrillic	SB
LT8MSWIN921	MS Windows Code Page 921 8-bit Lithuanian	SB
ET8MSWIN923	MS Windows Code Page 923 8-bit Estonian	SB
EE8MSWIN1250	MS Windows Code Page 1250 8-bit East European	SB, EURO
CL8MSWIN1251	MS Windows Code Page 1251 8-bit Latin / Cyrillic	SB, EURO
WE8MSWIN1252	MS Windows Code Page 1252 8-bit West European	SB, EURO
EL8MSWIN1253	MS Windows Code Page 1253 8-bit Latin / Greek	SB, EURO
BLT8MSWIN1257	MS Windows Code Page 1257 8-bit Baltic	SB, EURO
BLT8CP921	Latvian Standard LVS8-92(1) Windows / Unix 8-bit Baltic	SB
LV8PC8LR	Latvian Version IBM-PC Code Page 866 8-bit Latin / Cyrillic	SB
WE8NCR4970	NCR 4970 8-bit West European	SB
WE8NEXTSTEP	NeXTSTEP PostScript 8-bit West European	SB
CL8KOI8R	RELCOM Internet Standard 8-bit Latin / Cyrillic	SB
UTF8	Unicode 3.0 UTF-8 Universal character set	MB, EURO

Middle Eastern language character sets

The following table identifies the available character sets that support Middle Eastern languages.

Character Set Value	Description	Comments
AR8APTEC715	APTEC 715 Server 8-bit Latin / Arabic	SB
AR8ASMO708PLUS	ASMO 708 Plus 8-bit Latin / Arabic	SB
AR8ASMO8X	ASMO Extended 708 8-bit Latin / Arabic	SB
AR8ADOS710	Arabic MS-DOS 710 Server 8-bit Latin / Arabic	SB
AR8ADOS720	Arabic MS-DOS 720 Server 8-bit Latin / Arabic	SB
TR8PC857	IBM-PC Code Page 857 8-bit Turkish	SB
IW8PC1507	IBM-PC Code Page 1507 / 862 8-bit Latin / Hebrew	SB
AR8ISO8859P6	ISO 8859-6 Latin / Arabic	SB
IW8ISO8859P8	ISO 8859-8 Latin / Hebrew	SB
WE8ISO8859P9	ISO 8859-9 West European & Turkish	SB
LA8ISO6937	ISO 6937 8-bit Coded Character Set for Text Communication	SB
IW8MACHEBREWS	Mac Server 8-bit Hebrew	SB
AR8ARABICMACS	Mac Server 8-bit Latin / Arabic	SB
TR8MACTURKISHS	Mac Server 8-bit Turkish	SB
TR8MSWIN1254	MS Windows Code Page 1254 8-bit Turkish	SB, EURO
IW8MSWIN1255	MS Windows Code Page 1255 8-bit Latin / Hebrew	SB, EURO
AR8MSWIN1256	MS Windows Code Page 1256 8-Bit Latin / Arabic	SB, EURO
AR8MUSSAD768	Mussa’d Alarabi / 2 768 Server 8-bit Latin / Arabic	SB
AR8NAFITHA711	Nafitha Enhanced 711 Server 8-bit Latin / Arabic	SB
AR8NAFITHA721	Nafitha International 721 Server 8-bit Latin / Arabic	SB
AR8SAKHR706	SAKHR 706 Server 8-bit Latin / Arabic	SB

Character Set Value	Description	Comments
AR8SAKHR707	SAKHR 707 Server 8-bit Latin / Arabic	SB
UTF8	Unicode 3.0 UTF-8 Universal character set	MB, EURO

6

Tuning

Chapter summary

This chapter explains the factors that affect performance and presents ways to optimize the performance of your system.

List of topics

This chapter includes the following topics:

- Evaluating Your Installation
- Configuring the Hardware
- Tuning Your Data Warehouse
- Monitoring Performance

Evaluating Your Installation

How fast is fast enough?

Before getting too involved with tuning your system, you should evaluate whether any tuning is necessary. If response time is acceptable, then you can decide to leave your installation in its current configuration. On the other hand, if response time falls far below what you would consider reasonable, then tuning is mandatory.

This chapter describes several different approaches that you can take to improve the performance of your system. Some are easy while others are hard. Some cost money while others cost time. You will need to decide how fast is fast enough, and which steps you are willing to take to reach your goal.

Tuning check list

The following checklist identifies various tuning options. Look over this list periodically to make sure that you have addressed the areas most likely to degrade performance on your system:

- ❏ Configure OLAP Services to use a smaller, more efficient server character set. See Chapter 5.
- ❏ Dedicate a new computer with at least two processors to OLAP Services.
- ❏ Optimize the data warehouse by creating indexes, materialized views and partitions, and striping the data.
- ❏ Use the Performance Manager to pinpoint performance problems.

Configuring the Hardware

A dedicated computer

OLAP Services will perform best if it is running on a new computer dedicated to its use. If your budget allows for this expenditure, then you should invest in new equipment. A computer that is only a year or two old is significantly slower than one that you purchase today. While this may appear to be an expensive solution, it can more than pay for itself in labor savings. Your time costs money.

Recommended hardware configuration

The computer should have at least two CPUs. Acquire the fastest CPUs available, even if that means having fewer of them.

Note: If you must use a computer with only one processor, then you need to change the FreePageHighCount, FreePageLowCount, and related configuration settings, since the default configuration assumes two CPUs. Search for “FreePageLowCount” in Instance Manager Help for guidelines.

Memory requirements

The amount of memory used by OLAP Services varies widely depending upon the type of application being used.

A strict relational OLAP application (typically called *ROLAP*) needs 16 MB per instance with the PageBufferCount configuration parameter set to 4096 buffers.

ROLAP applications use OLAP Services as a temporary cache for data that is selected and calculated in the database.

A multidimensional OLAP application (typically called *MOLAP*) can require significantly more resources, particularly if large amounts of data are being aggregated in the analytic workspace. On a 32-bit computer, OLAP Services can use up to 4 GB of memory per instance. On a 64-bit computer, OLAP Services can use as much memory as you provide. In determining how much to add, you can consider, in addition to your budget, the size of the persistent workspace or its largest variable.

If OLAP Services is installed on the same computer as the relational database, then its requirements must be added to those of the database. Moreover, you must set the PageBufferCount configuration setting to no more than half of physical memory to allow a sufficient amount for use by the database. Use the following equation for this calculation:

```
PageBufferCount = (Physical memory - memory for the RDBMS) / 2
```

On 32-bit computers, limit the result to a maximum of 524,288 buffers.

Evaluate your network

While you are evaluating your hardware requirements, be sure to include your network. Users of business intelligence applications can significantly increase the traffic on an existing network, since they typically access large volumes of data. If there is a bottleneck at any point between the server and the end user, then performance will suffer.

Tuning Your Data Warehouse

Overview

The speed and efficiency with which Oracle can respond to a query strongly affects the response time experienced by end users. All data warehouses can benefit from the creation of the *best possible* indexes and materialized views. The wrong type of indexes or materialized views generated by the wrong type of SQL commands may degrade performance rather than improve it.

Extremely large data warehouses should be striped and partitioned. Before taking this step, however, you should confirm that the data at the lowest levels of aggregation are truly needed for the types of analysis being performed. Eliminating unnecessarily low-level data from your data warehouse is much easier than striping.

If your data warehouse is configured in a snowflake schema, you should look at the frequency that queries must perform joins on dimension tables. Denormalizing can improve performance significantly.

You might also want to consider storing some of the data in a persistent analytic workspace. This option is discussed in “Data Storage Alternatives” on page 2-11.

Data types of key columns

In a dimension table, the key column should have a NUMBER data type for the best performance. A primary index is always created on the key column to ensure that each row has a unique value. A NUMBER data type reduces the amount of disk space needed to store the index values for the key, since the index values are also stored as numbers instead of text strings. The smaller the index, the faster the database can search it.

The larger the number of values in the dimension, the greater the improvement in performance of NUMBER keys over VARCHAR (or other text) keys. Since time dimensions are typically rather small, a NUMBER key will improve performance only slightly. Thus, time dimensions can have either NUMBER or VARCHAR keys with little difference in performance between the them. However, dimensions for products and geographical areas often have thousands of members, and the performance benefits of NUMBER keys can be significant.

Indexing

Indexing is a vital component of any data warehouse. It allows Oracle to select rows quickly that satisfy the conditions of a query, without having to scan every row in the table. B-tree indexes are the most common; however, bitmap indexes are often the most effective in a data warehouse.

A column identifying gender will have in each cell one of two possible values to indicate male or female. Because the number of distinct values is small, the column has *low cardinality*. In dimension tables, the parent level columns also have low cardinality because the parent dimension values are repeated for each of their children.

A column containing actual sales figures might have unique values in most cells. Because the number of distinct values is large, columns of this type have *high cardinality*. Most of the columns in fact tables have high cardinality. Dimension key columns have extremely high cardinality because each value must be unique.

In fine-tuning your data warehouse, you may discover factors other than cardinality that influence your choice of an indexing method. With that caveat understood, here are the basic guidelines:

- Create bitmap indexes for columns with low to high cardinality.
- Create B-tree indexes for columns with very high cardinality (that is, all or nearly all unique values).

Striping and partitioning

Striping and partitioning are techniques used on large databases that contain millions of rows and multiple gigabytes of data. *Striping* is a method of distributing the data over your computer resources (such as multiple processors or computers) to avoid congestion when fetching large amounts of data. *Partitioning* is a method of dividing a large database into manageable subsets. Using partitions, you can reduce administration down-time, eliminate unnecessary scans of tables and indexes, and optimize joins.

If other methods of optimizing your database have not been successful in bringing performance up to acceptable standards, then you should investigate these techniques.

Materialized views

Oracle will rewrite queries written against tables and views to use materialized views whenever possible. For the optimizer to rewrite a query, it must pass several tests to verify that it is a suitable candidate. If the query fails any of the tests, then the query is not rewritten, and the materialized views are not used. And when the aggregate data must be recalculated at runtime, performance degrades.

All materialized views for use by the OLAP API must be created from within the OLAP management tool of OEM. Materialized views created elsewhere in Oracle Enterprise Manager or directly in SQL are unlikely to match the SQL statements generated by the OLAP API, and thus will not be used by the optimizer. Refer to “Creating Materialized Views” on page 4-23 for information about generating appropriate materialized views for OLAP applications.

Related information

Refer to the *Oracle9i Data Warehousing Guide* for detailed information on indexing, partitioning, and striping.

Your performance solution might also be in designing a hybrid OLAP system in which your most frequently accessed data is stored in a a persistent analytic workspace. Refer to “Data Storage Alternatives” on page 2-11.

Monitoring Performance

Performance Manager

In Oracle Enterprise Manager, the Performance Manager allows database administrators to monitor database performance in real time. You can use it to record statistics about OLAP Services, as well as the RDBMS and the operating system. Then you can display these statistics in a variety of formats, including tables, strip charts, horizontal and vertical bar charts, and pie charts.

Performance Manager is fully integrated with Oracle Capacity Planner, so that you can drill down on previously collected historical data. Using Capacity Planner, you can perform trend analysis on historical data to help predict your future requirements.

Configuration requirements

Oracle Performance Manager is a component of the Oracle Enterprise Manager Diagnostics Pack.

Oracle Intelligent Agent must be installed on the same computer as OLAP Services. The Intelligent Agent is installed with the Oracle RDBMS, so if OLAP Services is on the same computer as the database, then the Intelligent Agent is already available. However, if OLAP Services is installed on a separate computer, then you must install the Intelligent Agent. The Intelligent Agent runs as a daemon on Unix.

OLAP Services must be configured with the Data Gatherer module included in the module list. This is the default configuration. Additional configuration steps may be required. Check the README file for details.

Global counters

Global counters are preprogrammed into OLAP Services and collect data that apply to the service as a whole rather than individual sessions. You can use these counters to monitor usage patterns over time. These counters are loaded dynamically at runtime.

User counters

If your applications execute DML programs, then you can create and set your own counters. Like the global counters, these user counters can be viewed through Performance Manager. For more information about creating user counters, search OLAP DML Help for DGCART.

Procedure: Viewing charts

Follow these steps to view a chart of statistics for OLAP Services:

1. Open Performance Manager.
2. Log into Oracle Management Server if you want to save your statistics for further analysis. You can connect in Standalone mode, but you will not have access to those features that require a repository.

   You see the main Performance Manager screen.
3. Click the + next to **OLAP Services**.

   You see the OLAP Services Login dialog box.
4. Type in the following information, then click **OK**:

   - User name and Password: Your credentials for administering OLAP Services.
   - OLAP Services Instance: The name of the OLAP Services instance, such as `OlapSrv900`.
   - Collection Cartridge Location: The name of the computer where the collection cartridges are running.

   The OLAP Services node expands to show you the available counters.
5. To monitor a different instance of OLAP Services, redisplay the OLAP Services Login dialog box. From the File menu, choose **Set Collection Cartridge Location**.
6. Use the facilities of Performance Manager to gather and display statistics about OLAP Services. Click **Help** for detailed information.

A

Upgrading from Express Server

Appendix summary

This appendix identifies some of the major differences between Express Server 6.*x* and OLAP Services Release 1 (9.0.1).

List of topics

This appendix includes the following topics:

- Upgrading Express Databases
- Support for Applications
- Administration Methods
- Localization Support

Upgrading Express Databases

New terminology: Analytic workspace

In Oracle9*i*, the term *database* refers only to the relational database. Express databases are now called *analytic workspaces*. In OLAP Services, an analytic workspace can be used either as a transient data cache or as a persistent data repository. You can also use it to store SPL programs, Express Common Metadata (ECM), and data definitions.

Procedure: Upgrading an Express database

Follow these steps to upgrade an Express database for use as an analytic workspace in Oracle9*i*:

1. Make a backup of the Express database. (Optional)
2. Note the location of the database. By default, the OLAP service will search `olap/db` of the installation directory for databases. Other directories can be added to the search path by adding them to the ServerDBPath configuration setting. If a database is not in the search path, you will need to specify the full path name each time you attach it.
3. Check the ownership of the database file, and make sure that the installation user (such as `oracle`) has write access.

   For example, if the database is owned by `oesdba`, and the primary group is `dba`, then the installation user (such as `oracle`) must also be a member of the `dba` group, and the `dba` group must have write access.
4. Use OLAP Worksheet to open an OLAP Services session.
5. Check the instance character set with the following command:

   ```
   show nls_lang
   ```

   Note: Refer to “Localization Support” on page A-5 for more information on character sets.
6. Assign the instance character set to the database with the following command, substituting the name of your Express database for *dbname*:

   ```
   database attach dbname updatecharset
   ```

   Important: Proceed to the next step only if you are sure that the server character set includes all of the characters in the database. If you have any doubt, you should check the data stored in text variables to verify that it is readable. After you have updated the database, you cannot change its character set.
7. Save the new analytic workspace:

   ```
   update
   ```
8. Revise any programs in the workspace to delete references to obsolete commands, as described in “Express SPL” on page A-4.

Support for Applications

Programming environment

Applications for OLAP Services are developed in Java using the OLAP API, as explained in Chapter 2. You cannot run Windows C++, HTML, or Java applications that were developed for use with Express Server.

Programs developed in the Express SPL can be modified using the OLAP Worksheet and executed by the OLAP API. These programs may need to be revised to remove SPL commands that are no longer available.

Communications

OLAP Services uses CORBA communications for its Java applications. Communications through SNAPI and Express Web Agent are no longer available. Session sharing is not supported.

OLAP Worksheet uses XCA for communication with the analytic workspace. However, XCA is not supported for user-developed applications and may produce unexpected results.

Access to relational data

An OLAP service has a child-to-parent relationship with an Oracle relational database. An OLAP Services session is always connected to the database. You do not open a connection with the database as a separate or optional step.

You can retrieve data from the relational database in the following ways:

- The OLAP API uses the analytic workspace as a temporary or *transient* cache during a session. Selection and manipulation of the data is done entirely in the relational database before the data is fetched into a workspace. At the end of a session, the workspace is discarded.
- The OLAP DML's SQL command fetches data into dimensions and variables for further manipulation. At the end of a session, this *persistent* analytic workspace can be saved or discarded.

ODBC is not available, and thus access to third-party databases is not available through OLAP Services.

Oracle Express Relational Access Administrator and Oracle Express Relational Access Manager are not available in OLAP Services.

Express SPL

Numerous changes have been made to the Express Stored Procedure Language (now called the OLAP Data Manipulation Language). Support in the following areas has been *added* to the OLAP DML to provide compatibility with the RDBMS:

- Parallel aggregate
- Globalization Technology options
- Byte manipulation functions
- Data conversion functions
- New data types

Support in the following areas has been *dropped*:

- EXTCALL
- ODBC
- SNAPI
- XCA
- Operating system commands

For comprehensive lists of new, obsolete, and significantly revised commands, open OLAP DML Help and click **List of Changes** on the Contents page.

Metadata

In OLAP Services, the database administrator defines multidimensional objects and associated metadata in the relational database using the OLAP Management tool of Oracle Enterprise Manager, as described in Chapter 4.

The OLAP Worksheet allows DBAs to create objects in the analytic workspace by issuing DML commands. For the OLAP API to access these objects, the appropriate Express Common Metadata (ECM) must be defined.

Oracle Express Administrator is not available in OLAP Services, and the Oracle Express Objects metadata that it generated is not used by the OLAP API.

Administration Methods

Authentication

OLAP Services uses an operating system identity (the installation user or "oracle" identity) only for creating and accessing files, such as analytic workspaces and EIF files. This identity is established during installation and cannot be changed. You can delete other operating system identities created for use by Express Server (such as the DBA user, the Initialize user, the Default user, and individual user names) if they have no other purpose.

All other authentication is performed by the Oracle RDBMS. Applications must always present credentials before opening a session, and those credentials must match a user name and password stored in the relational database. Before users can access OLAP Services, you must define user names and passwords in the database, as described in "Granting Access Rights to Users" on page 3-27.

Because the relational database is required for authentication, you can open a session with OLAP Services only when the database is available. If the database is unavailable, then you cannot run administrative tools or business analysis applications.

Administrative tools

Oracle Enterprise Manager encompasses the tools for administering OLAP Services, as described in Chapter 3, providing a common user interface across all platforms. Instance Manager runs through Oracle Enterprise Manager, rather than as a standalone or Web application. Performance data is also gathered and displayed by Oracle Enterprise Manager instead of the Windows Performance Monitor.

The `xscosvc` command-line administrative utility replaces `oesmgr`. You can execute OLAP DML commands as schedule batch jobs through Oracle Enterprise Manager. This expanded functionality in Oracle Enterprise Manager replaces the `oescmd` utility.

Localization Support

Full support by Globalization Technology

The Globalization Technology used by OLAP Services provides much more extensive localization support and is much easier to administer than the localization features of Express Server. The NLS_LANG configuration parameter identifies the instance character set, and the default language and territory, as described in Chapter 5. Setting NLS_LANG correctly is the only localization task that you need to perform for an OLAP service.

If you are upgrading Express databases that used translation tables, then you can delete those tables because they are not needed by OLAP Services. Likewise, you should check your Express programs for use of obsolete commands and keywords that supported translation tables. If you will use OLAP Services to access multibyte data (either in an Express database or external files), then you might find "Multibyte character set equivalents" on page A-6 helpful in choosing a global environment. Note that the CHARSET option is now obsolete.

Support for Globalization Technology has been added to the OLAP DML. These options allow an application to query the current localization settings and override the behaviors controlled by the default language and territory.

Multibyte character set equivalents

The following table identifies equivalent locale settings for multibyte character sets in OLAP Services and Express Server:

Express Server DefaultCharacterSet Parameter or CHARSET Option Value	OLAP Services NLS_LANG Parameter Setting
EUC	JAPANESE_JAPAN.JA16EUC
SHIFTJIS	JAPANESE_JAPAN.JA16SJIS
HANGEUL	KOREAN_KOREA.KO16KSC5601
SCHINESE	SIMPLIFIED CHINESE_CHINA.ZHS16GBK
TCHINESE	TRADITIONAL CHINESE_TAIWAN.ZHT16BIG5

Index