

Oracle® Sales Analyzer

Administrator's Guide

Release 11*i*

March 2002

Part No. A96145-01

ORACLE®

Part No. A96145-01

Copyright © 1996, 2002, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited

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 Express is a trademark or registered trademark of Oracle Corporation. Other names may be trademarks of their respective owners.]

Contents

Send Us Your Comments	xi
Preface.....	xiii
Conventions.....	xvi
Documentation Accessibility	xvii
1 Introduction to Oracle Sales Analyzer	
The Analyzer Product Family.....	1-2
Database Configurations.....	1-3
Windows Client Access Mode: Thick Client	1-5
Windows Client Access Mode: Thin Client	1-7
Issues for Thick and Thin Client Access.....	1-9
Web Client Access Mode.....	1-10
2 Your Role as Administrator	
Summary of Tasks.....	2-2
Summary of Tools.....	2-3
Relational Access Administrator	2-4
Batch Manager.....	2-5
Express	2-6
Analyzer Web Administrator	2-10
Application Manager	2-11

Database Manager	2-12
Analyzer.....	2-14
Tools for Troubleshooting.....	2-15
3 Building the Database	
Measures, Variables, and Formulas.....	3-2
Dimensions	3-3
Levels and Hierarchies.....	3-6
Attributes.....	3-9
Data Access.....	3-9
The Build Process.....	3-11
4 Working With Express Databases and Catalogs	
Working with Express Databases	4-2
Remote Express Databases.....	4-3
Remote Personal Databases	4-4
Local Express Databases	4-6
Working with Catalogs	4-7
Catalogs in the Server Database	4-8
Catalogs in the Catalog Database	4-8
Integrated Catalogs.....	4-9
5 Setting Up Database Access for the Windows Client	
General Instructions for Database Setups	5-2
Creating a Configuration File.....	5-6
Selecting an Access Mode	5-10
Specifying the Database Path for a Slice.....	5-11
Maintaining a Shell Database	5-12
Setting Up Login to the Database or to a Relational Manager	5-16
Creating an Icon to Start a Database	5-19
Setting Up a Workstation User Name	5-22
Changing the Default Directory for Thin Client Publish to File	5-24

6	Setting Up Communications for a Windows Client	
	General Instructions for Communications Setups.....	6-2
	Communications Setups for Thick Client Access.....	6-3
	Communications Setups for Thin Client Access.....	6-5
7	Managing User Access to a Remote Database	
	Establishing User Access.....	7-2
	Working with Groups.....	7-2
	Working with Users.....	7-8
	Importing User Names from Oracle Applications.....	7-11
	Working with Libraries.....	7-12
	Implementing Security.....	7-17
8	Setting Up Database Access for Web Users	
	About Web Access.....	8-2
	Defining Virtual Directories in the Web Listener.....	8-3
	Implementing Single Sign-on.....	8-3
	Setting Up the Analyzer Home Page.....	8-6
	Customizing the Web Application.....	8-16
	Providing Web Access to Users.....	8-18
9	Scoping a Database	
	Overview of Scoping.....	9-2
	Defining Data Cubes.....	9-3
	Copying Data Cubes.....	9-6
	Defining Scopes.....	9-7
	Enabling Scoping.....	9-12
	Associating Scopes with User Groups.....	9-12
	Refreshing Scopes for a Shared Shell.....	9-13
	Modifying Scopes.....	9-14
	Removing Scopes.....	9-14

10 Building a Slice

General Instructions for Building a Slice	10-2
Defining a Slice	10-3
Creating a Slice Directly	10-8
Creating a Slice from Files	10-10
General Instructions for Updating a Slice	10-13
Updating a Slice Directly	10-14
Updating a Slice from Files	10-16
Upgrading a Slice	10-17

11 Customizing a Database

Changing Dimension Value Names	11-2
Changing Level Names	11-5
Providing Permanent Custom Aggregates and Measures	11-8
Hiding Measures	11-11
Modifying Aggregation Methods	11-13
Configuring Databases for Forecasting	11-19
Adjusting for Fluctuations in Time	11-20
Defining the Latest Time Period	11-21
Modifying Translation Tables	11-24

12 Controlling the Display of Data

Attributes for Displaying Measure Values	12-2
Formatting Measure Values	12-3
Installing Measure Value Formats	12-6
Changing the Order of Dimension Values	12-9
Setting Dimension Value Limits	12-11
Specifying Default Documents	12-15

13 Customizing the Windows Client Interface

Customizing Menus and the Toolbar	13-2
Managing Customizations	13-7
Installing Custom Help	13-8
Installing a Custom Application.....	13-11
Using Languages Other Than English	13-13

A Catalog Descriptions

AG.CATALOG (Custom Aggregate Catalog)	A-2
AP.CATALOG (Applications Catalog)	A-4
CA.CATALOG (Custom Analysis Catalog)	A-5
CO.CATALOG (Component Catalog)	A-5
DA.CATALOG (Data Catalog)	A-7
DACAT (Data Catalog)	A-11
DF.CATALOG (Data Format Catalog)	A-15
DI.CATALOG (Display Catalog)	A-17
DMCAT (Dimension Catalog).....	A-19
DV.CATALOG (Data Cube or Data View Catalog)	A-21
FC.CATALOG (Forecast Catalog)	A-22
FD.CATALOG (Folder Catalog)	A-23
GP.CATALOG (Group Catalog)	A-24
GR.CATALOG (Graph Catalog)	A-25
INCAT (Infer Catalog)	A-27
MP.CATALOG (Map Catalog)	A-28
MRCAT (Measure Catalog).....	A-28
MX.CATALOG (Maximum Values Catalog).....	A-31
OP.CATALOG (User Option Catalog).....	A-33
SC.CATALOG (Slice Catalog)	A-34
SG.CATALOG (Segment Catalog).....	A-35
SL.CATALOG (Saved Selection Catalog).....	A-36
SO.CATALOG (Source Catalog)	A-37
SOCAT (Source Catalog).....	A-39
SP.CATALOG (Scope Catalog)	A-40
TB.CATALOG (Report or Table Catalog).....	A-41
UR.CATALOG (User Catalog)	A-43

B Initialization Files

Overview of Initialization Files	B-2
Application Manager (APPMGR.INI)	B-2
Analyzer (ANALYZER.INI)	B-8
Database Manager (DBMANAGE.INI)	B-14

C RDC File

Overview of RDC File	C-2
RDC File Functions	C-2
Example: RDC File	C-2

D Web Configuration File

Overview of OSAWEB.CFG File	D-2
OSAWEB.CFG File Sections	D-2
Example: OSAWEB.CFG File	D-2

Index

Send Us Your Comments

Oracle Sales Analyzer Administrator's Guide, Release 11*i*

A96145-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

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

- FAX: 781-744-0001 . Attn: Oracle Sales Analyzer
- Postal service:
OLAP Products Documentation Manager
Oracle Corporation
Oracle Sales Analyzer Documentation
10 Van de Graaff Drive
Burlington, MA 01803
USA

If you would like a reply, please give your name, address, and telephone number below.

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

Preface

What this manual is about

This manual describes how to set up Oracle® Sales Analyzer (hereinafter referred to as “Analyzer.”) It provides an overview of Analyzer, briefly describes the process for building a database, and details how to set up database access for Windows client and Web users.

The manual also explains how to modify a database and customize Analyzer to suit your environment.

Intended audience

This manual is intended for DBAs.

Analyzer documentation set

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

- *Oracle Sales Analyzer User’s Guide* — Familiarizes users with the Analyzer’s capabilities as a reporting and analysis system for on-line analytical processing (OLAP).
- *Oracle Sales Analyzer Installation and Update Guide* — Provides instructions for using Oracle Installer to install the Analyzer client and server, and contains information about system requirements.
- *Express Installation and Configuration Guide* — Explains how to install and configure Oracle Express Server (hereinafter referred to as Express Server) or Personal Express.

- *Oracle Express Web Products Installation Guide* — Explains how to install and configure the server-side components of an Express Web Agent or Express Web Publisher installation.
- *Oracle Express Programmer's Guide to the Express Language* — Explains the Express programming environment and illustrates programming strategies for accessing and analyzing data.
- *Oracle Express Database Administration Guide* — Explains how to configure existing Express databases for optimal use with Express Client products, and how to create new databases. Describes the Express data model and database objects.
- *Oracle Express SNAPI Guide* — Explains how to use SNAPI to develop client applications that interact with Express Server.
- *Oracle Express Relational Access Manager User's Guide* — Describes Oracle Express Relational Access Manager, and explains fundamental concepts pertaining to data warehouse design and optimization. The guide also provides instructions for using Relational Access Administrator, a graphical utility and component of Relational Access Manager.
- *Oracle Sales Analyzer Data Loader Guide* — Provides information about the data loader program and instructions for building an Analyzer database from data that is stored outside of Personal Express and Oracle Express Server databases.

Structure of this document

The *Oracle Sales Analyzer Administrator's Guide* is structured as follows:

- **Chapter 1** introduces Oracle Sales Analyzer, as well as products and components in the Oracle Sales Analyzer family. This chapter also describes database configurations and methods of access.
- **Chapter 2** summarizes the administrative tasks that are detailed throughout this guide, and introduces the tools and methods that you can use to accomplish these tasks.
- **Chapter 3** describes the remote and local Express databases that make up an Analyzer database. It also describes the catalogs that reside in the Express databases, which determine how Analyzer stores, formats, and uses data.
- **Chapter 4** provides an overview of the structure of Analyzer data, methods of data access, and procedures for designing and building an Analyzer database.

- [Chapter 5](#) explains how to set up Windows client access to an Analyzer database.
- [Chapter 6](#) describes how to set up communications between a Windows client and a remote system.
- [Chapter 7](#) discusses database security and describes how to set up users, groups, and libraries in order to control user access to data and to saved objects in the database.
- [Chapter 8](#) explains how to use set up client access to an Analyzer database via a Web browser.
- [Chapter 9](#) details the procedures associated with scoping a database, in order to restrict group access to specified data.
- [Chapter 10](#) describes methods and procedures for building local slice databases, in order to provide portability of or faster access to specified data.
- [Chapter 11](#) explains how to modify an Analyzer database in order to tailor it to users' requirements for data analysis.
- [Chapter 12](#) explains how you can control the format and presentation of data displayed by Analyzer.
- [Chapter 13](#) describes how you can modify the Analyzer user interface to best suit the needs of your users.
- [Appendix A](#) lists Analyzer catalogs and their properties.
- [Appendix B](#) describes Analyzer initialization files.
- [Appendix C](#) describes the Remote Database Configuration (RDC) file.
- [Appendix D](#) describes the Web configuration file (OSAWEB.CFG) for Analyzer.

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 (for example, Result).
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 <code>show eversion</code> , you would type all the characters exactly as shown in the fixed-width font.
<i>Italic</i> text	Indicates variables, including variable text. Variable text is used when dialog boxes or their components are unlabeled or have labels that change dynamically based on their current context. The wording of variable text does not exactly match what you see on your screen. Italic type is also used for emphasis, for new terms, and for titles of documents.
<u>Underlined</u> text	Indicates a default value in descriptions of Express language syntax.
UPPERCASE text	Indicates Express commands and objects and acronyms.

Mouse usage

Always use the left mouse button unless you are specifically instructed to use the right mouse button.

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.

Formats for key combinations and sequences

Key combinations and key sequences appear in the following formats.

you see the format
Key1+Key2,	press and hold down the first key while you press the second key. For example: “Press Alt+Tab” means to press and hold down the Alt key while you press the Tab key.
Key1, Key2,	press and release the keys one after the other. For example: “Press Alt, F, O” means to press and release the Alt key, press and release the F key, then press and release the O key.

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 to facilitate access by the disabled community. 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/>.

Introduction to Oracle Sales Analyzer

Chapter summary

This chapter introduces Oracle® Sales Analyzer (hereinafter referred to as “Analyzer”), as well as products and components in the Analyzer family. This chapter also describes database configurations and methods of access.

List of topics

This chapter includes the following topics:

- [The Analyzer Product Family](#)
- [Database Configurations](#)
- [Windows Client Access Mode: Thick Client](#)
- [Windows Client Access Mode: Thin Client](#)
- [Issues for Thick and Thin Client Access](#)
- [Web Client Access Mode](#)

The Analyzer Product Family

Analyzer

Analyzer is an application for on-line analytical processing (OLAP) that provides a wide range of tools for ad hoc analysis of corporate data. Analyzer retrieves data from a variety of sources and from local or remote locations. It integrates data into one logical database, and presents requested analyses at the desktop.

Analyzer is based on the Express multidimensional data model. This model enables professionals in diverse areas such as sales, marketing, manufacturing, distribution, and administration to analyze and evaluate data effectively.

Express

The term *Express* refers to two similar OLAP systems by Oracle Corporation. These are Oracle Express Server (hereinafter referred to as “Express Server”) on remote platforms and Personal Express on PCs.

Oracle Sales Analyzer – Database Manager

Oracle Sales Analyzer – Database Manager (hereinafter referred to as “Database Manager”) is a graphical utility that facilitates many of your tasks as database administrator (DBA).

Oracle Application Manager

A component of Analyzer, Oracle Application Manager (hereinafter referred to as “Application Manager”) is a graphical utility that you use to set up access to Analyzer databases and communications between your PC and a remote system. You can also use Application Manager to create slice databases, which are subsets of larger remote databases.

Oracle Express Relational Access Manager

Oracle Express Relational Access Manager (hereinafter referred to as Relational Access Manager) builds an Analyzer database by accessing data from a relational database management system (RDBMS) Actual data does not have to be stored in Express, but can be fetched at runtime from the RDBMS in response to user requests for data. Alternatively, a *hybrid* database can store specified data permanently in Express, while fetching remaining data from the RDBMS at runtime.

Relational Access Administrator

Relational Access Administrator is a graphical utility that is a component of Relational Access Manager. You can use Relational Access Administrator to define an Express data model that is based on the data in an RDBMS, determine how Express accesses data from an RDBMS, and define an Analyzer database.

Data loader program

The data loader program is used to build a multidimensional Express database from data that is stored in flat files.

Oracle Express Batch Manager

Oracle Express Batch Manager (hereinafter referred to as Batch Manager) is a graphical utility that you use to create, monitor, and control batch jobs. You can use Batch Manager to launch jobs such as building an Analyzer database with Relational Access Manager or the data loader, as well as running script files of Express commands.

Analyzer Web Administrator

Analyzer Web Administrator is a Web page that you use to set up the Analyzer Home page for users who will access the application through a Web browser.

Database Configurations

Types of database configurations

An Analyzer database can be set up for user access in one of the following configurations:

- Remote database
- Slice database

Remote database configuration

A remote database configuration has the following characteristics:

- The database is created and maintained on a remote system that has storage facilities capable of handling the vast quantity of data typically stored in data warehouses. When accessed by a Windows client, an active communications link is used throughout the user session.
- There is a public library system for controlling user access to data and saved objects.
- User access to the database is provided by Express Server, with or without Personal Express.
- Data in the database can be stored in Express Server, an RDBMS, or a hybrid. (Access to an RDBMS from the server is via direct connection or ODBC.)
- User-defined objects such as reports, graphs, custom aggregates, and custom measures are saved on the remote system. This feature allows users to access personal data and other saved objects from any location and with any access method.

Methods for accessing a remote database

Users can access a remote database using any of the following methods:

- Thick client, as described in the topic [“Windows Client Access Mode: Thick Client” on page 1-5](#)
- Thin client, as described in the topic [“Windows Client Access Mode: Thin Client” on page 1-7](#)
- A Web browser, as described in the topic [“Web Client Access Mode” on page 1-10](#)

Slice database configuration

A slice database provides users with quick access to frequently used data from a larger database without relying on active communications with the remote database. In a slice database, there is no public library system for controlling user access to data and saved objects. All user-defined objects are saved in the user’s personal database.

For information about creating slice databases, see [Chapter 10](#).

Using multiple database configurations

You may find that you will use multiple database configurations for the same site. For example, you may allow some users to access a remote database, while others access a slice (subset) of the remote database distributed to the PC or LAN, and still others do both.

[Chapter 5](#) explains how to define a database setup, including specifying the database configuration.

Windows Client Access Mode: Thick Client

Overview of thick client mode

When a user accesses a remote database in thick client mode, Analyzer uses Personal Express on the PC to access metadata, fetch data from Express Server on the remote system, and cache data locally on the PC. In addition, Analyzer uses Personal Express to integrate catalogs in the public and personal databases, and to perform many data calculations.

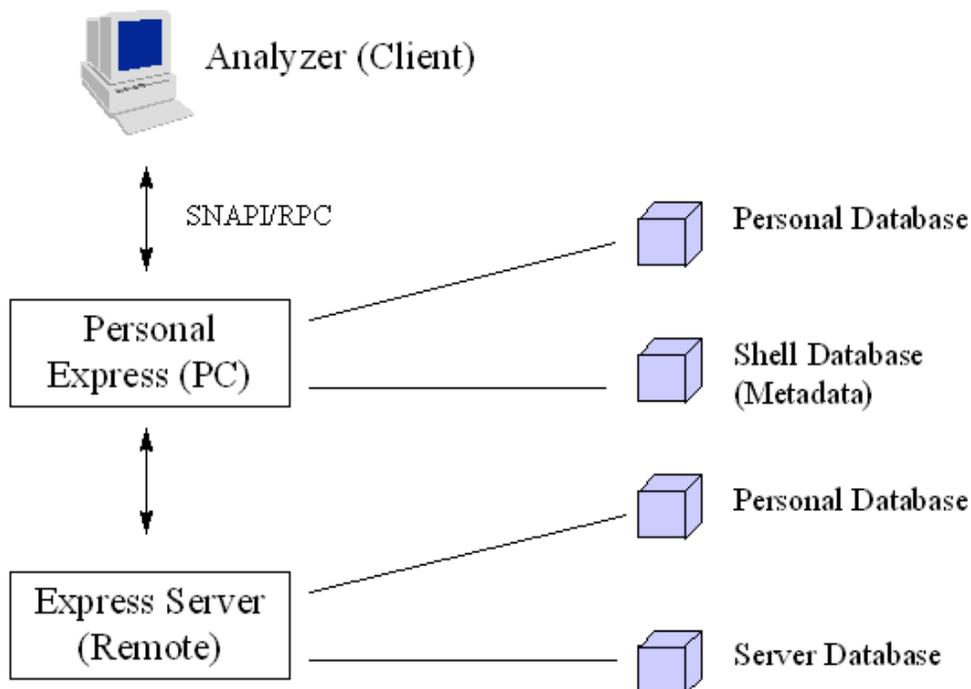
Database access in thick client mode

Database access in thick client mode is as follows:

- The application programming interface (API) between the Microsoft Windows client application (Analyzer) and Personal Express is SNAPI (Structured N-Dimensional API). The transport mechanism between Analyzer and Personal Express is local RPC.
- The interface between Personal Express on the PC and Express Server on the remote system is XCA (Express Communications Architecture). A network connection (via TCP/IP) is the transport mechanism between Personal Express and Express Server.
- Metadata (e.g., dimension names and information describing measures) from the server and catalog databases on the remote system is stored in the shell database on the PC or LAN. If the remote database has been updated, metadata is refreshed to the shell database at startup.
- User-defined objects (e.g., reports, graphs, custom aggregates, and custom measures) are saved in the personal database both on the remote system and on the PC.

Illustration of thick client mode

The following is an illustration of database access in thick client mode.



Related information

For information about setting up thick client access to a database, see [Chapter 5](#).

For information about Express databases (e.g., the server, shell, catalog, and personal databases) and catalogs, refer to [Chapter 4](#).

Windows Client Access Mode: Thin Client

Overview of thin client mode

When a user accesses a remote database in thin client mode, Analyzer does not use Personal Express on the PC. Instead, Analyzer fetches data directly from Express Server on the remote system. All calculations that are performed in Express are done in Express Server.

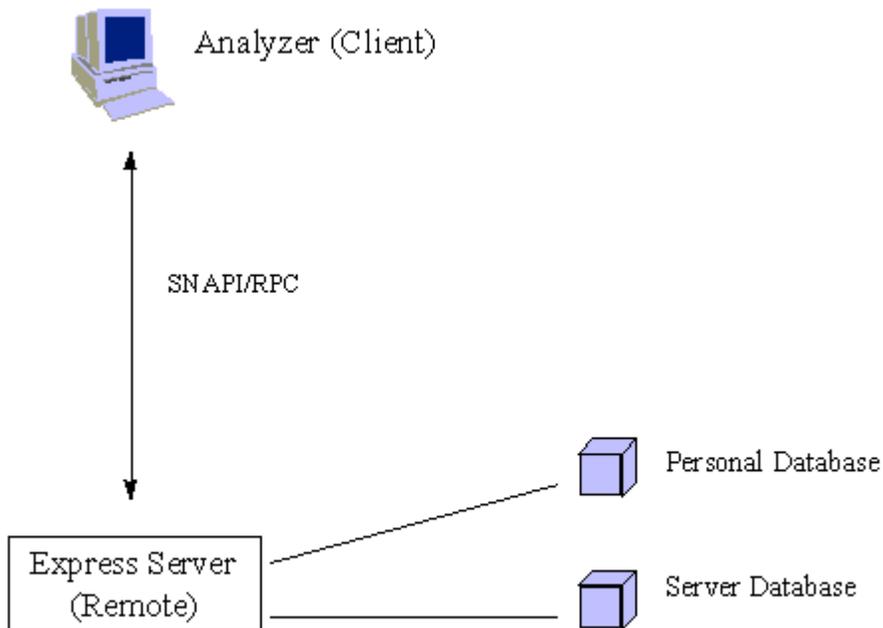
Database access in thin client mode

Database access in thin client mode is as follows:

- The API between the Windows client application (Analyzer) and Express Server is SNAPI. An RPC (Remote Procedure Call) connection (via TCP/IP) is the transport mechanism.
- Metadata, user-defined objects, and data are stored on the remote system only. Metadata from the updated database is not refreshed to the PC at startup.

Illustration of thin client mode

The following is an illustration of database access in thin client mode.



Related information

For information about setting up thin client access to a database, see [Chapter 5](#).

For information about Express databases (e.g., the server, catalog, and personal databases) and catalogs, refer to [Chapter 4](#).

Issues for Thick and Thin Client Access

Comparison of thick and thin client modes

The following table compares and contrasts performance issues for thick and thin client modes.

Performance Issues	Thick Client Mode	Thin Client Mode
Startup	Metadata must be refreshed to the PC at startup each time the remote database has been updated. If communications are slow, or if database updates are frequent, this might be impractical.	Metadata is not refreshed to the PC; therefore, startup time can be greatly reduced.
Runtime	Because the metadata exists locally on the PC, runtime performance is usually better. This advantage is even more pronounced if communications to your remote system are relatively slow.	Because metadata does not exist locally, runtime performance is usually slower, especially if communications are slow or if the remote system is overloaded or under-powered.
Network traffic	Network traffic is lower.	Network traffic is higher.
Demands on remote system	There are lower demands on the remote system because most processing of data occurs on the PC in Personal Express.	The load on the remote system is relatively high.
Creation of slices	Users can create slices by downloading data directly from a remote database. Alternatively, users can create slices using slice files that the DBA has distributed.	Users cannot create slices directly from a remote database. Users must create slices using slice files that the DBA has distributed.
Ease of improving system performance	System performance can be improved by upgrading the PC. However, a relatively fast PC is required.	Since most processing is done on the remote system, upgrading the remote system will yield significantly better performance for all thin client mode users.

Using both thick and thin client modes

If your system configuration and performance allows, users can access a remote database using the client mode of choice. For example:

- To save time at startup, salespeople who run only one 5-minute report daily might use thin client mode.
- To save time at runtime, analysts who manipulate data extensively throughout the day might use thick client mode.
- A user might use thick client mode in the office, and then switch to thin client mode to access data quickly from another location.

Web Client Access Mode

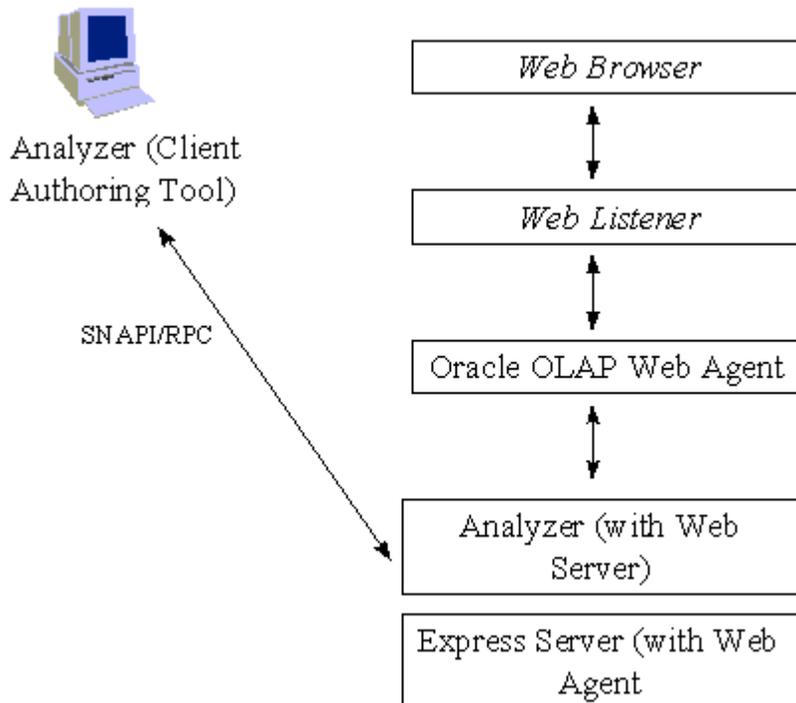
About database access via a web server

Database access via a web server is as follows:

- A standard Web browser (Netscape, for example) is the user application for requesting and displaying Analyzer documents and data via the Web. The browser passes a user request as a URL to a Web listener. The URL contains an Express command or indicates the location of a Web document on the server.
- Oracle OLAP Web Agent (OOWA), acting as an interface, transmits commands from the Web listener to the Analyzer server and Express Server.
- The user request is fulfilled by the Analyzer server and Express Server. To construct Web documents, Express uses Oracle Express Web Agent. The fulfilled request is then transmitted through OOWA to the Web listener.
- The Web listener routes the document or data back to the user's browser, where it is displayed.

Illustration of web access

The following diagram illustrates the web access process.



Related information

For information on setting up Web access, see [Chapter 8](#).

Your Role as Administrator

Chapter summary

This chapter summarizes the administrative tasks that are detailed throughout this guide, and introduces the tools and methods that you can use to accomplish these tasks.

List of topics

This chapter includes the following topics:

- [Summary of Tasks](#)
- [Summary of Tools](#)
- [Relational Access Administrator](#)
- [Batch Manager](#)
- [Express](#)
- [Analyzer Web Administrator](#)
- [Application Manager](#)
- [Database Manager](#)
- [Analyzer](#)
- [Tools for Troubleshooting](#)

Summary of Tasks

Introduction

There are several tasks that you can perform in order to enable users to access and analyze data effectively. Whether you will perform a particular task depends on your system configuration and user requirements.

Procedures for all of the tasks listed in this topic are described throughout this guide. However, the guide may direct you to another manual or a Help system for greater detail.

List of tasks

The following table lists each administrative task that is described in this guide, and references the corresponding chapter where that task is discussed.

Administrative Task	Chapter
Building the Analyzer database, and updating it with new data	Chapter 3
Working with the Express databases and catalogs that make up the Analyzer database	Chapter 4
Setting up Windows client access to an Analyzer database	Chapter 5 and Chapter 6
Controlling user access to data	Chapter 7
Setting up Web client access to an Analyzer database	Chapter 8
Scoping a remote database in order to restrict groups of users to specific views of data	Chapter 9
Providing slices of a remote database on users' PCs or on a LAN server; for example, to provide rapid access to frequently used data	Chapter 10
Modifying and customizing an Analyzer database, for example, changing names for dimension values, modifying aggregation methods, and customizing time periods	Chapter 11
Modifying translation tables in Express Server in order to allow successful translation of variations in character sets between platforms	Chapter 11
Controlling the format of measure values that will be displayed to users	Chapter 12

Administrative Task	Chapter
Setting maximum limits on the number of dimension values allowed in reports, graphs, custom aggregates, and actions in the Selector	Chapter 12
Specifying a default report or graph that replaces the system-defined default	Chapter 12
Customizing Analyzer menus and the toolbar for users	Chapter 13
Providing a custom application or Help system to users	Chapter 13
Adapting Analyzer for languages other than English	Chapter 13

Summary of Tools

Introduction

You can use a variety of tools to accomplish your tasks as administrator. Often, you can use a graphical utility, such as Database Manager, to perform a task that you would otherwise do using Express. When a graphical alternative is available, this guide provides instructions for using it rather than Express.

List of tools

Your tools for administrative tasks include the following:

- Relational Access Administrator
- Batch Manager
- Express
- Analyzer Web Administrator
- Application Manager
- Database Manager
- Analyzer

Each of these tools is defined in the topic [“The Analyzer Product Family”](#) on page 1-2.

Relational Access Administrator

Using Relational Access Administrator

You can use Relational Access Administrator to do the following:

- Use a wizard to design a default data warehouse model. As an alternative to using the wizard, you can design a data warehouse model that uses other supported schemas.
- Specify the relational tables that are included in your data warehouse schema, designate each table as a lookup or fact type, and then associate lookup tables with corresponding fact tables.
- Model the database objects with which the data warehouse is structured. These objects include dimensions, levels, attributes, hierarchies, and variables.
- Map the columns in the data warehouse tables that correspond to these objects.
- Tune the dimensions, levels, attributes, hierarchies, and variables that you have identified for your data warehouse.
- Determine how Express will access data from the RDBMS.
- Define the Analyzer database.

Related Information

For general information about building a database, see [Chapter 3](#) of this guide.

For more information about Relational Access Manager and Relational Access Administrator, see the *Oracle Express Relational Access Manager User's Guide*.

For detailed information about using Relational Access Administrator, refer to the Relational Access Administrator Help system.

Batch Manager

Using Batch Manager

You can use Batch Manager to run database builds or script files of Express commands as follows:

- Define batch jobs for specified instances of Express
- Use a defined communications setup for thin client mode and an RPC connection or create a new setup for communications with the remote system
- Launch a job immediately or at a later time that you specify
- Identify the priority of a job, so that Batch Manager will schedule it accordingly within a given time period
- Monitor the status of jobs that are pending, running, or finished
- Specify the creation of a log file to record the output of a particular job
- Create multiple build configurations that apply to a single project

How Batch Manager connects to Express

Batch Manager runs batch jobs without having to rely on a continuous connection to the server. Instead, Express Server launches “partner” XCA sessions periodically, only as needed. As a result, your PC is not tied up as a job runs; also, there is little risk of a job running unsuccessfully due to a lost connection.

Related information

For general information about building a database, see [Chapter 3](#) of this guide.

For detailed information about using Batch Manager, refer to the Batch Manager Help system.

Express

Using Express Server vs. Personal Express

As a tool, the term *Express* refers collectively to the Express Server and Personal Express command languages with which Analyzer is, in part, built. Some of your work as administrator may involve modifying catalogs in remote Express databases — the catalog, server, and public databases — directly in Express. In a remote database configuration, you will work directly in Express Server using thin client mode access.

Definition: Master startup

Before you can modify remote catalogs and perform other DBA tasks directly in Express, you must perform a master startup of the database. A master startup ensures that Express databases are attached appropriately for your work, as follows:

- It attaches the catalog and public databases with read/write access.
- It attaches the server database and code databases with read-only access.
- It does not create or attach the personal database.
- It synchronizes any changes between the catalog, server, and public databases.

To enable a master startup, you must have read/write access to the remote Express databases at the operating-system level.

Procedure: Performing a master startup

To perform a master startup, follow these steps:

1. Start Database Manager, as described in the topic [“Database Manager” on page 2-12](#).
2. From the File menu, open your Analyzer database.

Note: Alternatively, you can open your Analyzer database using a database icon that starts Database Manager. (Creating a database icon is described in [“Creating an Icon to Start a Database” on page 5-19](#).)

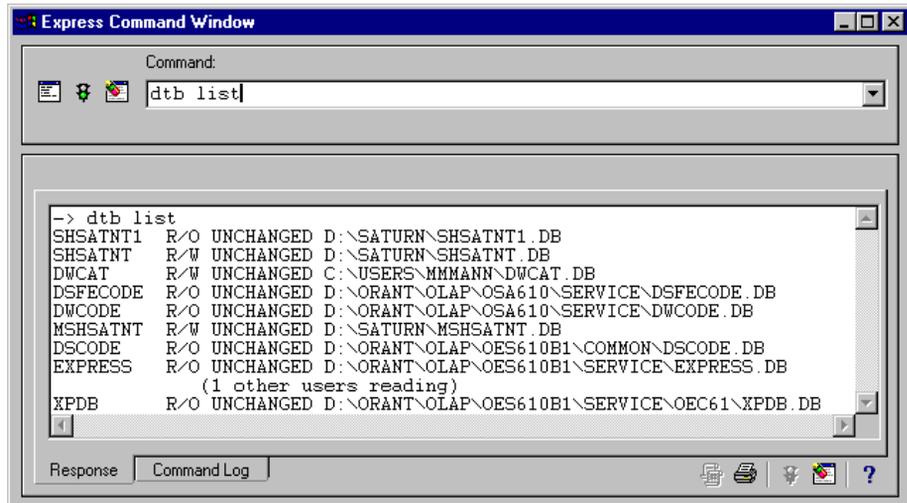
Issuing Express commands

To issue commands locally (to Personal Express) and remotely (to Express Server) you use Database Manager. After you have opened the database from Database Manager, you can work directly in Express by choosing the **Express Command Window** from the Database Manager Help menu.

Important: Although you can also issue Express commands from the Help menus of Analyzer and Application Manager, always access Express from Database Manager if you plan to modify catalogs directly in Express. Opening a database from Database Manager performs a master startup, which is required in order to work directly in Express.

Example: Issuing an Express command

The following example shows the Express databases that are listed for the sample Saturn database.



The screenshot shows a window titled "Express Command Window" with a command input field containing "dtb list". Below the input field is a text area displaying the output of the command. The output lists several databases with their respective permissions and file paths. At the bottom of the window, there are buttons for "Response" and "Command Log", and a toolbar with icons for print, copy, paste, and help.

```

-> dtb list
SHSATNT1  R/O UNCHANGED D:\SATURN\SHSATNT1.DB
SHSATNT   R/W UNCHANGED D:\SATURN\SHSATNT.DB
DWCAT     R/W UNCHANGED C:\USERS\MMANN\DWCAT.DB
DSFECODE  R/O UNCHANGED D:\ORANT\OLAP\OSA610\SERVICE\DSFECODE.DB
DWCODE    R/O UNCHANGED D:\ORANT\OLAP\OSA610\SERVICE\DWCODE.DB
MSHSATNT  R/W UNCHANGED D:\SATURN\MSHSATNT.DB
DSCODE    R/O UNCHANGED D:\ORANT\OLAP\OES610B1\COMMON\DSCODE.DB
EXPRESS   R/O UNCHANGED D:\ORANT\OLAP\OES610B1\SERVICE\EXPRESS.DB
          (1 other users reading)
XPDB      R/O UNCHANGED D:\ORANT\OLAP\OES610B1\SERVICE\OEC61\XPDB.DB
  
```

Procedure: Modifying catalogs using Express commands

To modify catalogs using Express commands, perform the following steps:

1. From the Database Manager Help menu, choose **Express Command Window**.

Result: The Express command window is displayed, from which you will execute Express commands.

2. Verify that the Express database that houses the catalog that you plan to modify is attached with read/write access.

Example: `->database list`

3. You have these options:

IF ...	THEN ...
the database is attached with read-only access,	detach the database, and reattach the database properly, as shown in procedures throughout this guide. Example: if you planned to modify an object in the server database directly, you would issue commands such as these: <code>->database detach dsaturn•</code> <code>->database attach dsaturn first rw</code>
someone else has the database attached with read/write access,	you will receive a message telling you that you cannot have read/write access. You must arrange for that user to exit before you can make catalog modifications.

4. You can use a qualified data reference (QDR) to modify a catalog.

Example: This example enters the name Henry Thoreau in the NAME property of the User catalog.

```
->ur.catalog(ur.prop 'NAME' ur.entry 'HTHOREAU') -
```

```
Continued> = 'Henry Thoreau'
```

Note: When numerous modifications are required, you might prefer to write an Express program. The advantage to using an Express program instead of issuing commands at the Express command window is that you can duplicate your work should you have to rebuild your databases for any reason. You can use the custom database in your user directory to store these Express programs. (If you are working in Express Server, the custom database resides in your user directory on the remote system.)

5. Your catalog modifications (e.g., a change in scope) might require a refresh in order to synchronize changes between the remote Express databases. In this case, ensure that the server database is attached with read/write access. Then, use a QDR to update the time stamp property in SOCAT.

Example:

```
->socat(soprop 'TIME.MODIFIED') = ds.timestamp
```

Note: Even if a refresh is not required, it is a good idea to identify yourself in the MODIFIER and TIME.MODIFIED properties of a catalog so that there is a record of who last changed the catalog and when it was changed.

6. Save your changes by typing `update`. When you finish your work, press the close button to close the Express command window and return to Database Manager. From Database Manager, close the database.
7. If you have updated the time stamp catalog property in Step 5, reopen the database from Database Manager. Changes will be synchronized as the remote Express databases are refreshed.
8. If you are refreshing a shell database, close the database. From Analyzer, open the database, using a special “refresh ID” and a database setup with thick client mode and read/write access to the shell.

Note: Refreshing a shell is detailed in [Chapter 5](#) in the topic “[Maintaining a Shell Database](#)” on page 5-12.

Procedure: Backing up work in Express

If you want to back up your work in Express, you can export catalogs to a file using the Express EXPORT command. Then, if necessary, you can recover your work using the IMPORT command.

If you want to back up your work in Express, perform the following steps:

1. Verify that the Express database that houses the catalog that you plan to export is attached first with read/write access.

Example:

```
->database list
```

2. If the database is not attached first with read/write access, detach the database, and then reattach the database properly.

Example: if you planned to export a catalog in the catalog database, you would issue commands such as these.

```
->database detach saturn  
->database attach saturn first rw
```

3. Set the status of all dimensions in the database to all their values.

Example: ->allstat

4. Back up the catalog by exporting it to an EIF file. Be sure to back up any other catalogs that contain properties that interact with that catalog.

Example:

```
->export ur.catalog gp.catalog to eif file -  
Continue>'users.eif' data dfns
```

Note: If you are using Express Server, the EIF file will be created on the remote system.

Note: After you have backed up your work, you can recover it, if necessary, using the IMPORT command.

Related information

For more information about Express commands, functions, and options, as well as a glossary definition of QDRs, see the *Oracle Express Language Reference Manual*.

Analyzer Web Administrator

Using Analyzer Web Administrator

You use Analyzer Web Administrator to perform the following tasks:

- Create an Analyzer Home page from which users can select the appropriate database and start Analyzer.
- Identify the Analyzer databases that will be available for Web access.
- Specify how users will access Analyzer's Java foundation classes when they start Analyzer on their PCs.
- Determine the appearance of Analyzer Web pages, including customizing the banner and background.
- Set up Analysis Library reports.

Related Information

For instructions for using Analyzer Web Administrator, see [Chapter 8](#). For a description and illustration of Web access, see [Chapter 1](#).

Application Manager

Using Application Manager

You can use Application Manager to do the following:

- Set up Windows client access to a slice or remote database using the Database Setup dialog box. (See [Chapter 5](#).)
- Set up communications with a remote system using the Communications Setup dialog box. You can set up communications for thick client mode access and a network connection, or for thin client mode access and an RPC connection. (See [Chapter 6](#).)
- Create a slice of a source database for local access using the Maintain Slice dialog box. You can create slices directly from a remote database accessed in thick client mode. Alternatively, you can create slices from slice files that have been distributed. (See [Chapter 10](#).)
- Arrange for a refresh of one or more remote databases at a scheduled date and time (a “delayed refresh”) using the Open Later dialog box. (See [Chapter 5](#).)
- Issue Express commands. However, to modify catalogs directly in Express, ensure that you issue Express commands from Database Manager, as described in the topic “[Issuing Express commands](#)” on page 2-7.
- Access the System Information and Version Information dialog boxes, as described in the topic “[Tools for Troubleshooting](#)” on page 2-15.

Starting Application Manager

You should start Application Manager from its icon in the Oracle Sales Analyzer program group. Do not use the Run command in Windows to start Application Manager. The user directory will not be set correctly and the program will not be able to find the initialization file (APPMGR.INI).

Related information

For more information, refer to the Application Manager Help system.

Database Manager

Using Database Manager

You can use Database Manager to perform the following functions:

- Define data cubes using the Maintain Data Cube dialog box. Data cubes are used in the creation of scopes and slices. (See [Chapter 9](#).)
- Define a scope using the Maintain Scopes dialog box. Scoping restricts group access to a specific data set within a remote database. (See [Chapter 9](#).)
- Define a slice using the Maintain Slices dialog box. A slice is a subset of a remote database, which can be used to provide groups with faster access to data. (See [Chapter 10](#).)
- Define users, groups, and libraries to control user access to data and shared objects. (See [Chapter 7](#).)
- Enable user validation for database security. User validation prevents users with invalid user names from accessing the database. (See [Chapter 7](#).)
- Customize descriptions of the dimensions in your database using the Maintain Dimensions dialog box.
- Reset a catalog using the Catalog Manager dialog box. For example, this might be necessary if a hardware failure during catalog transfer causes a catalog to be out of synchronization or unavailable for access.
- Use the Governors dialog box to limit the number of dimension values that will be allowed in reports, graphs, and custom aggregates, as well as in drag-and-drop sorting and the Family tool in the Selector. You can also specify the error condition that will result when a user attempts to reach or exceed a value limit. (See [Chapter 12](#).)
- Issue Express commands. Use this feature if you plan to modify catalogs directly in Express.
- Access system and version information, as described in the topic “[Tools for Troubleshooting](#)” on page 2-15.

Prerequisites for using Database Manager

The following database access and configuration settings are required in order to use Database Manager:

- You have read/write access to the catalog, server, and public databases at the operating-system level.
- You are a member of the group that has Administrator access. The group has read/write access to the MASTER library and other public libraries.
- The access mode in the configuration file for the database setup specifies a remote database with thin client mode and read-only access to the database, as described in [Chapter 5](#).
- You are the only user. Only one user can access Database Manager at a time.

Starting Database Manager

You can start Database Manager from its own icon or from a database icon that starts Database Manager, and then opens a database. From Database Manager, you can open and close databases, as well as switch between Database Manager modules.

Master startup

When you open a database from Database Manager, Database Manager performs a master startup of the database. A master startup is required before you can modify remote catalogs and perform tasks directly in Express.

If you plan to work directly in Express, be sure to open the database from Database Manager, and then access Express from the Database Manager Help menu. For more information, see the topic [“Express” on page 2-6](#).

Related information

For more information about using Database Manager [Database Maintenance] and Database Manager [Analyzer], refer to their respective Help systems

Analyzer

Using Analyzer

In your role as administrator, you can use Analyzer to do the following:

- Customize Analyzer menus and the toolbar or provide a custom application or Help system using the Customize dialog box. (See [Chapter 13](#).)
Note: Customizations that you make are stored independently of any database; therefore, you can provide a single customized interface for your users, regardless of how many databases they access.
- Create a new Analyzer report or graph, and then save it for use as the default document. For example, you can specify a default report that uses dimension values that are already stored in your data warehouse rather than values that must be summarized at runtime. (See [Chapter 12](#).)
- Create saved objects such as reports, saved selections, custom measures, and custom aggregates. These objects reside initially in your personal database. However, using the Maintain Library dialog box, you can copy these objects to public libraries that you have created for general use. (See [Chapter 7](#).)
- Control user access to data by defining data cubes and slices using the Maintain Data Cube dialog box and the Maintain Slice dialog box, respectively. (See [Chapter 9](#) and [Chapter 10](#).)
- Create backup copies of custom aggregates, custom measures, data cubes, or similar definitions by copying the affected catalog entries to a file using the Maintain Library dialog box. (See [Chapter 7](#).)
- Issue Express commands. However, to modify catalogs directly in Express, ensure that you access Express from Database Manager, as described in the topic [“Express” on page 2-6](#).
- Access the System Information and Version Information dialog boxes, as described in the topic [“Tools for Troubleshooting” on page 2-15](#).

Starting Analyzer

You can start Analyzer from Application Manager after you have opened a database. You can also start Analyzer from its own icon or from a database icon that starts Analyzer and then opens a database. From Analyzer, you can open and close databases as needed.

Related information

For more information about using Analyzer, refer to the *Oracle Sales Analyzer User's Guide*.

For more information about performing administrative tasks using Analyzer, search for the following topics in the Analyzer Help system:

- “Customize Dialog Box”
- “Maintain Data Cubes Dialog Box”
- “Maintain Libraries Dialog Box”
- “Maintain Slices Dialog Box”

Tools for Troubleshooting

System environment information

Information about the system environment, as well as installed and required versions of software, is available from the About *Application* menu item on the Help menu in Application Manager, Database Manager, and Analyzer running in the Windows client environment. The summary of information that is provided in two dialog boxes is helpful if troubleshooting becomes necessary.

Overview of system information

The System Information dialog box provides information about hardware and software variables that apply to the application environment. As DBA, you will find this information useful for troubleshooting problems that may arise due to differences that exist among users' hardware, networks, and installed applications. Such problems might consist of mismatched DLL and VBX files, incorrect DOS or database path information, or insufficient system resources.

The System Information dialog box includes the following four tabs, which provide information about the user's computer hardware, Windows setup, current application, and DOS setup:

- System tab — Includes Processor, Resources, and Memory sections.
- Windows tab — Includes Operating System, Video, and General sections.
- Application tab — Includes Application Executable, Application Directory, Application .INI File, Application Arguments, Personal Express Executable, and Express Database Path sections.
- DOS tab — Includes DOS version, Path, and Environment sections.

Overview of version information

The Version Information dialog box provides important information about the version of the installed application and lists DLL, VBX, and database versions that are installed and required. It also provides the path name for each listed database or database object.

Procedure: Accessing system and version information

To access system or version information, perform the following steps:

1. From the Help menu of Application Manager, Database Manager, or Analyzer running in the Windows client, choose *About Application*. The *About Application* box is displayed.
2. You can access system or version information:

IF you want to access . . .	THEN . . .
system information,	choose Info. Result: The System Information dialog box is displayed.
version information,	choose Version. Result: The Version Information dialog box is displayed.

Related information

For detailed information about accessing system and version information, including tips for troubleshooting, search for the following topics in the Help system for Application Manager, Database Manager, and Analyzer:

“System Information Dialog Box”

“Version Information Dialog Box”

Building the Database

Chapter summary

This chapter provides information about the structure of Analyzer data and methods of data access. The chapter also provides an overview of the process that you follow to build an Analyzer database. This is introductory material only; for detailed information, refer to the documentation for the tool that you will use to build your database: the *Oracle Sales Analyzer Data Loader Guide* for a database created from flat files; the *Oracle Express Relational Access Manager User's Guide* for a database that accesses an RDBMS.

List of topics

This chapter includes the following topics:

- [Measures, Variables, and Formulas](#)
- [Dimensions](#)
- [Levels and Hierarchies](#)
- [Attributes](#)
- [Data Access](#)
- [The Build Process](#)

Measures, Variables, and Formulas

Introduction

Measures, variables, and formulas are among the database objects that Analyzer uses to define data and relationships within the data. You should be familiar with these structural elements and how they interact when you design a database.

Variable

Analyzer stores data in Express objects called *variables*. A variable is a multidimensional array structure whose cells hold individual data values in a database. A variable can store data temporarily for a single session, or permanently on disk for use from one session to the next.

Formula

A *formula* is a special kind of variable that calculates values instead of storing them — actually, a formula stores an algorithm that defines how values are to be calculated when the formula is used. A formula provides data from another source (either calculated from another Express variable or fetched from an external source, such as an RDBMS) as needed to fulfill a user request for data.

Measure

The values provided by a given variable or formula represent actual measurement data of one particular type, known as a *measure*. For example, the sample Global Computing database includes measures, such as “Sales,” “Units Sold,” “Customer Discount,” “Cost,” “Price,” and “Margin.” A measure can be defined as either a variable or a formula.

A measure is always accessed through a formula that can calculate *custom aggregates*. A custom aggregate consists of multiple values at a lower level that are rolled up into a single, higher-level value.

Dimensions

Definition: Dimension

A dimension is a list of values that provide categories for data. For example, dimensions in the sample Global Computing database represent products, customers or geographic areas, channels, and time periods. Other types of databases might have these or similar dimensions, plus additional dimensions, or a different combination of dimensions entirely.

Definition: Dimension value

One of the elements in the list that makes up a dimension is referred to as a *dimension value*. For example, in the sample Global Computing database, the Product dimension includes values, such as “Portable PCs,” “Desktop PCs,” and “Monitors.”

The concept of multidimensionality

Analyzer uses a *multidimensional* array structure to organize data. Each variable in the database can be thought of as an array that is associated with, or “dimensioned by,” multiple dimensions. In any given array, each cell contains data associated with a specific combination of the values of the dimensions that apply to that variable. To illustrate, assume that there is a variable called “Sales” that has the dimensions Time, Channel, and Product. Each cell in the array for the variable “Sales” would represent a different, unique combination of Product, Channel, and Time values — sales of portable PCs via mail-order catalog in the month of May 2000 would be one example.

Dimensions are stored as database objects that are separate from any one variable or formula. In fact, the same dimensions can be shared by many variables to define the data.

Number of dimensions for a variable

Each data variable can have up to 10 base dimensions.

Typical dimensions and dimension values

The following table describes some typical dimensions and related dimension values.

Name	Values	Description
Product dimension	Values might include the following: “All Products” An entire class of products, such as “Hardware” A family of products, such as “Portable PCs” An item, such as “Envoy Executive”	Reveals how data varies by product. The term <i>product</i> refers to any value in the Product dimension, regardless of its level. For example, “Hardware” and “Envoy Executive” are both considered products.
Geography or Customer dimension	Values might include the following: <ul style="list-style-type: none"> ■ “All Clients” ■ An entire region, such as the “United States” or “Europe” ■ Territories within regions, for example, the “Western U.S.” or all of “France” ■ Specific accounts, such as “Business World San Jose” or the “Ministry of Finance Paris” 	Reveals how data varies by geographic area or customer. The term <i>geography</i> or <i>customer</i> refers to any value in the Geography or Customer dimension, regardless of its level. For example, “Western U.S.” and “Business World San Jose” are both considered customers.

Name	Values	Description
Channel dimension	Values might include the following: <ul style="list-style-type: none"> ▪ “All Channels” ▪ Individual channels, such as “Direct/Phone,” “Direct/E-Mail,” “Distributors,” “Mail-Order Catalogs,” and “On-line Catalogs” 	Reveals how data varies according to each distribution channel that you employ. The term <i>channel</i> refers to any value in the Channel dimension, regardless of its level. For example, “All Channels” and “Direct/Phone” are both considered channels.
Time dimension	Values might include the following: <ul style="list-style-type: none"> ▪ Years, such as “2001” and “2002” ▪ Quarters, such as “Quarter 1 - 2002” and “Quarter 2 - 2002” ▪ Months, such as “May 2002” and “June 2002” 	Reveals how data varies over time. If your data is dimensioned by time, Analyzer detects the periodicity of your data as part of the process of building your Analyzer database. Analyzer sorts time periods and defines chunks of similar time periods (i.e., one-week time periods are in a separate chunk from four-week time periods) so that it can manipulate the data correctly. It identifies year-ago periods, prior periods, and the latest period. This allows Analyzer to calculate time-related aggregations, such as moving averages.

Measure dimension

Although measures themselves exist in the database as either variables or formulas, a Measure dimension contains the names of all of the measures. This enables Analyzer to store information about the measures in variables dimensioned by the Measure dimension, and to select a group of measures easily for processing. As noted previously, the sample Global Computing database includes measures, such as “Sales,” “Units Sold,” “Customer Discount,” “Cost,” “Price,” and “Margin.”

Levels and Hierarchies

Definition: Level

Data is typically collected at the lowest level of detail. This data is often aggregated into higher level totals, which is more useful for analysis. In Analyzer, all of the data for a particular measure (such as “Sales”) is eventually stored in one Express variable regardless of the level, so that low-level and aggregate data are stored together in one object.

Embedded levels in the dimensions of a measure variable represent the data’s *aggregation levels*. Each embedded level is referred to as a *level* in its respective dimension.

Example: Levels

In a Product dimension, data collected at the item level (e.g., “Envoy Standard” or “Sentinel Standard”) might be aggregated into totals for family (e.g., “Portable PCs” or “Desktop PCs”), class (e.g., “Hardware” or “Software”), and “All Products.” In a Geography or Customer dimension, data collected for key accounts (e.g., “Business World San Jose” or “Ministry of France Paris”) might be aggregated into totals for territory (e.g., “Western U.S.” or “France”), region (e.g., “United States” or “Europe”), and “All Clients.” For the Time dimension, data collected monthly might be aggregated into totals for quarters and years.

Definition: Hierarchy

The levels in a dimension are organized into one or more *hierarchies*. Within a hierarchy, each level above the base level represents the aggregated total of the data from the level below.

Example: Hierarchy

The following example illustrates the *product hierarchy* in the sample Global Computing database.

Level	Value
CLASS	Hardware
FAMILY	Portable PCs
ITEM	Envoy Executive

You can see how the values at the lower levels contribute to the aggregate values of the higher levels. For example, the “Envoy Executive” item contributes to the aggregate value of “Portable PCs” at the FAMILY level.

Family structure within a hierarchy

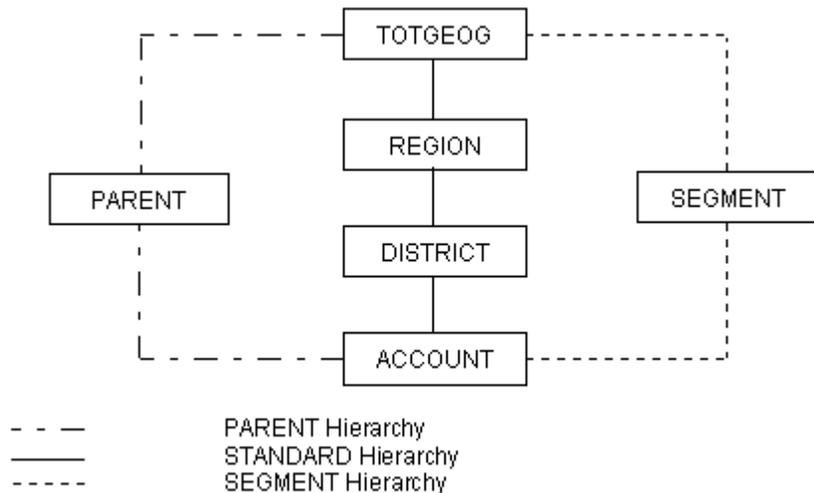
The hierarchical structure of a dimension imposes a family structure on its dimension values. For a particular dimension value, the value at the next higher level is its *parent*, and a value at the next lower level is its *child*. For example, in the sample Global Computing database, “Portable PCs” is the child of “Hardware” and the parent of “Envoy Executive.” Other values that exist at the FAMILY level, for example, “Desktop PCs,” are *siblings* of “Portable PCs.”

Secondary hierarchies

A database might have more than one hierarchy within a dimension. This means that a level can have more than one parent. The absolute (overall) order of the levels, however, must remain the same; levels cannot be inverted between hierarchies. For example, if level A is higher than level B in one hierarchy, level A cannot be lower than level B in another hierarchy.

Example: Secondary hierarchies

The following figure shows an example of a dimension (Geography) with two secondary hierarchies (Parent and Segment), in addition to the Standard hierarchy.



In this example, the TOTGEOG, REGION, DISTRICT, and ACCOUNT levels constitute the Standard hierarchy. However, each value at the ACCOUNT level has a parent at the PARENT and SEGMENT levels, as well as at the DISTRICT level. Both PARENT and SEGMENT, in turn, have TOTGEOG as their parent.

Using levels and hierarchies in data analysis

Users can select by level when requesting data in a report or graph. For example, a user might select Product dimension values by ITEM to see how a particular product is performing in comparison with other items. The hierarchical structure of a dimension also makes it possible for users to access individual values in a dimension by *drilling down* on or *drilling up* on the levels in the hierarchy. Drilling down allows the user to view child values that are associated with parent values in the hierarchy. By drilling up, the user can collapse the list of descendant values that are associated with a parent value in the hierarchy. By using a secondary hierarchy, a user can skip levels when drilling.

Attributes

Definition: Attribute

Unlike levels, which are groupings based on actual “physical” (numerical) values, *attributes* are logical groupings that allow users to select diverse dimension values based on common characteristics, such as brand, size, color, or flavor.

Example: Using attributes

For example, in a soft drink products database, dimension values at various levels in the Product dimension — BRAND, FORM, and SIZE, for instance — might all have the same descriptive value for the attribute FLAVOR, such as grape or orange. Users could select data for reports and graphs based on the FLAVOR attribute to see which dimension values, at any level, are associated with a particular flavor.

Levels as attributes

Hierarchical levels may also serve as attributes of the dimension values at levels above the lowest hierarchical level. Using the sample Global Computing database as an example, suppose that the family level in the product hierarchy also serves as an attribute in the database. The corresponding attribute value of “Envoy Executive” tells you that it is a portable PC, independent of its position in the hierarchy.

Data Access

Methods of accessing data

Analyzer is designed to access data from a variety of sources. Your data is typically collected and stored using a large-scale data storage system with a type of data format that can store vast amounts of data efficiently.

As the following table shows, Analyzer uses different methods of accessing your data, depending on the data source and your access needs:

IF the data source is . . .	THEN Analyzer uses . . .
an RDBMS,	Relational Access Manager to fetch data on an as-needed basis. Specified aggregate data may be stored in an Express cache.
flat files,	the data loader program to copy data into multidimensional data variables that are stored permanently in Express. The data in its original format does not have to remain on-line for use by Analyzer.

Accessing data with Relational Access Manager

Relational Access Manager accesses data that is stored permanently in an RDBMS defined through SQL. Data is accessed according to the data warehouse schema that you have defined in the RDBMS using Relational Access Administrator as a tool. The data warehouse schema corresponds to the Express multidimensional data model.

After the data model is complete, users can request data in a report or graph. In response to user requests for data, Relational Access Manager retrieves actual data values from the relational database and places them temporarily in variables in an Express cache. Analyzer then uses both the data values in these variables and the structures in Express to present the requested information to the user.

Using Relational Access Manager's hybrid mode

If your users typically access certain data, you may want to make specific data available on a permanent basis. Using the hybrid mode of Relational Access Manager, you can preload specific aggregate data, while allowing users to fetch remaining data dynamically at runtime. Whether you use hybrid mode is a performance issue. This mode allows you to conserve system resources; however, you must be sure to update data for your users as new data becomes available.

Using the data loader program

The data loader program loads data that has been extracted from another system and stores it in an Analyzer database. The data loader program creates an Express database from flat files, which define the database structures. It then reads source data into these structures. The result is a server database with the data stored

“permanently” in Express variables. As new data becomes available, you use the data loader to update the database.

Related information

For information about using Relational Access Manager, refer to the *Oracle Express Relational Access Manager User’s Guide*.

For information about using the data loader program, refer to the *Oracle Sales Analyzer Data Loader Guide*.

The Build Process

Step 1: Identify measures and dimensions

The first step in building a database is to identify the measures that you want to access and define the dimension hierarchy for these measures. For Relational Access Manager, you use a series of fact and lookup tables in the data warehouse to provide the information. For the data loader program, you provide the information in a series of files.

Step 2: Define the database structure in Express

The second step in building a database is to define the database structure in Express. Using the information in the RDBMS warehouse tables or the data loader control file, a set of programs called the *extractor* reads the data and creates a set of rigidly defined structures in an extractor database in Express. Among these structures are dimensions and variables. These structures contain all of the information needed to create your database, but they are not the final Analyzer structures.

Step 3: Create the Analyzer database

In the third and final step, a set of programs called the *common module* creates the server database from the data in the extractor database. In addition, the data loader program actually loads the data into the structures created by the common module.

Important: Adding a dimension to a variable after the build

You should not add a dimension to a variable after the database is built. Adding a new dimension is a major design change, the impact of which can be very

significant in terms of work already done on custom measures, custom aggregates, and reports.

If you must add a dimension to a variable after the database is built, do your work in a development or non-production environment. Then use Analyzer to check the results before implementing the changes on the production database

Related information

For information about using Relational Access Manager, refer to the *Oracle Express Relational Access Manager User's Guide*.

For information about using the data loader program, refer to the *Oracle Sales Analyzer Data Loader Guide*.

For information about using Relational Access Administrator and Batch Manager, refer to their respective Help systems.

Working With Express Databases and Catalogs

Chapter summary

This chapter describes the remote and local Express databases that make up an Analyzer database. It also describes the various catalogs that reside in the Express databases, and that determine how Analyzer stores, formats, and uses data.

List of topics

This chapter includes the following topics:

- [Working with Express Databases](#)
- [Remote Express Databases](#)
- [Remote Personal Databases](#)
- [Local Express Databases](#)
- [Working with Catalogs](#)
- [Catalogs in the Server Database](#)
- [Catalogs in the Catalog Database](#)
- [Integrated Catalogs](#)

Working with Express Databases

Definition: Express databases

A user is presented with one logical Analyzer database; however, the user is really accessing numerous Express databases. Each is a collection of data objects such as variables, dimensions, formulas, catalogs, and programs, as described in [Chapter 4](#).

Analyzer automatically attaches required Express databases remotely or locally, depending on the database configuration and access mode.

Viewing a list of Express databases

Using Express, you can view a list of the Express databases that are attached. The following example refers to a remote Analyzer database that was built using the data loader program. In the data loader control file, the value for the DATADB keyword is DSATURN and the value for the MASTERDB keyword is SATURN.

```
->database list
SATURN    R/O UNCHANGED F:\SATURN\SATURN.DB
DSFECODE R/O UNCHANGED F:\SERVER\DSFECODE.DB
PSATURN  R/W UNCHANGED C:\WINNT\SYSTEM32\PSATURN.DB
MSATURN  R/O UNCHANGED F:\SATURN\MSATURN.DB
DSATURN  R/O UNCHANGED F:\SATURN\DSATURN.DB
DSCODE   R/O UNCHANGED F:\SERVER\DSCODE.DB
EXPRESS  R/O UNCHANGED F:\ORAHOME\SERVICE\EXPRESS.DB
XPDB     R/O UNCHANGED F:\SERVER\XPDB.DB
```

The name shown to the left — the simple file name, without a path or file name extension — is the database name used by Express. You use the database name to attach or detach a particular database with the Express DATABASE command. The full file name is shown to the right.

Modifying Express databases

As administrator, you can list, examine, and modify objects in the Express databases, using Express directly or a graphical tool such as Database Manager. For an overview of using Express and Database Manager, see [Chapter 2](#).

For a discussion of the structure of Analyzer data, including descriptions of database objects such as variables and dimensions, refer to [Chapter 3](#).

Remote Express Databases

Definition: Remote Express databases

In a remote database configuration, some databases always reside on the remote system. Regardless of your database configuration, these are always referred to as *remote Express databases*.

In a remote database configuration, remote Express databases are built in Express Server. In a slice database configuration, remote Express databases are built in Personal Express.

List of remote Express databases

The following table lists and describes the remote Express databases.

Database	Name	Description
Catalog database	For example, SATURN	Contains many of the catalogs that are used to define an Analyzer database.
Public database (remote configuration only.)	Remote: "M" plus catalog database name (e.g., MSATURN)	Contains a set of catalogs with entries published by the DBA or individual users, providing saved objects such as reports, custom aggregates, and saved selections. Catalogs in the public database correspond to integrated catalogs in the personal database.
Server database	e.g., DSATURN	Contains dimensions, variables, formulas, and relations. When used to access data from an RDBMS, it can contain formulas for retrieving some data.
Personal database	"P" plus the catalog database name (e.g., PSATURN)	Contains catalogs that provide the structures for storing personal reports, custom measures, custom aggregates, saved selections, and similar objects created by the user. In a remote database configuration, integrated catalogs in the personal database correspond to those in the public database. The personal database also contains catalog entries for objects saved by a user during the Analyzer session.

Database	Name	Description
Custom database (optional)	For example, CUSTOM	Contains your custom Express programs (e.g., preload or postload programs used in conjunction with the data loader). Note: If you want to customize the server database itself, you can run a custom program at the end of the build process. You can identify this program in the control file for the build.
Code databases	XPDB, DSCODE, DSFECD, EXPRESS, etc.	Contain Express and Analyzer code.

Remote Personal Databases

Default location for remote personal databases

By default, Analyzer creates remote personal databases in users' home directories on the server where Express is installed.

Alternative: Root directory for remote personal database directories

You have the option to override the default and specify a root path for remote personal database directories. When you specify a root path, the individual user home directories are assumed to be subdirectories of the root.

The user home subdirectories are named as follows:

- If the user provides a database ID, the ID is used as the name of the user home subdirectory.
- If the user does not provide a database ID (e.g., user validation is not enabled), the login ID is used as the name of the user home subdirectory.

You use the Remote Database Configuration (RDC) file associated with an Analyzer database to specify the root path for user homes. The RDC file is usually located in the directory on the server where the database is built. If users access more than one database, you might want to modify several RDC files.

Considerations for specifying a root directory

Specifying a root path for remote personal database directories offers the following features:

- You can optimize the setup of the server to suit the requirements of the Analyzer installation.
- Multiple database IDs can share a single login account.
- The user home directories for one database can be in a different location than the user home directories for another database.
- User home directories can exist on the same server on which Express Server is running, or on another remote server.

If the user home directory and Express are on different servers

If you want to establish a root user home directory on a different server than the one on which Express Server is running, ensure that you consider issues such as access rights, performance, and network traffic. Also ensure that when you specify a drive letter for the directory path, it is a mapped drive that has been specified in the Drive Map tab of Express Instance Manager.

Procedure: Specifying a root directory for remote personal databases

Follow these steps to specify a root directory for remote personal databases.

1. Identify the Remote Database Configuration (RDC) file for the database.

Note: The RDC file is usually located in the directory on the server where the database was built.

2. Using a text editor, open the RDC file and add the root path for the personal databases to the [Paths] section.

Example: `personal=d:\users\`

Important: If the path that you specify is on another server, the drive letter must be a mapped drive that is specified in the Drive Map tab of Express Instance Manager.

3. Save the file.

Example: RDC file with personal database path

The following example illustrates the [Paths] section of an RDC file. The section includes a path specification for the root directory for personal databases.

```
[Paths]
code=c:\orahome\olap\oes6xx\common\
data=d:\db\coffee\
extend=e:\db\coffee\
export=d:\export\
personal=d:\users\
```

Local Express Databases

Definition: Local Express databases

In a remote database configuration, some databases may reside on the user's PC. Regardless of your database configuration, these are referred to as *local Express databases*.

List of local Express databases

The following table lists and describes the local Express databases.

Database	Name	Description
Shell or server database	Shell database: the same name as the server database on the remote system (e.g., DSATURN).	When the user accesses a remote database in thick client mode, a local <i>shell</i> database is created to hold metadata from both the server and catalog databases on the remote system. Each user can maintain the shell on the PC or users can share one shell across a LAN, as described in Chapter 5 . In a slice database configuration, the user accesses a local server database.
Personal database	"P" plus the catalog database (e.g., PSATURN)	Contains catalogs for storing and saving objects such as reports, custom measures, custom aggregates, and saved selections. When the user accesses a remote database in thick client mode, a mirror image of the personal database on the remote system is maintained on the PC.

Database	Name	Description
Custom database (optional)	For example, CUSTOM	If Personal Express is installed, a custom database can contain custom Express programs. It can be shared on a LAN or each user can have a personal copy. Note: If each user has a copy, the user should ensure that the database is backed up.
Code databases	DSFECODE, DSCODE, EXPRESS,	If Personal Express is installed, local code databases contain Express and Analyzer code.

Working with Catalogs

Definition: catalogs

Several Express databases contain catalogs. Catalogs are simply two-dimensional Express variables that provide an organized and logical method of storage. Throughout a user's session, Analyzer uses the information provided by catalogs to determine how data is stored, formatted, and used.

Modifying catalogs

You can modify catalogs in the remote Express databases (e.g., the catalog, server, and public databases) using Express directly. However, a graphical tool such as Database Manager or Analyzer is available as an easy alternative.

You will notice that some catalogs in the catalog database have corresponding catalogs in the server database. If you modify such a catalog directly in Express, you will simply update the time stamp in the catalog that you are modifying to synchronize the corresponding catalog.

Any catalogs that you modify in the public database are automatically integrated in the user's personal database at startup.

Related information

Modifying catalogs using Express directly or using a graphical tool (e.g., Database Manager) is described in [Chapter 2](#). When a graphical alternative to Express is available, this guide provides instructions for using it rather than Express.

The contents of all of the catalogs described in this chapter are detailed in [Appendix A](#).

Catalogs in the Server Database

Function of catalogs in the server database

The server database contains catalogs that are created and populated as part of the build procedure. You can change the default values in some of the catalogs, as described in this guide.

List of catalogs in the server database

The following table lists and describes the catalogs that reside in the server database.

Catalog	Name	Description
Data catalog	DACAT	Describes all base and inferred measures in the server database.
Dimension catalog	DMCAT	Describes the dimensions defined for the database.
Infer catalog	INCAT	Describes the set of measures in your database that will be inferred or calculated.
Measure catalog	MRCAT	Describes the set of base measures defined for the database.
Source catalog	SOCAT	Contains values that you can use to control the behavior of Analyzer. This catalog corresponds to SO.CATALOG in the catalog database.

Catalogs in the Catalog Database

Function of catalogs in the catalog database

Catalogs that reside in the catalog database provide information that Analyzer needs to maintain the server database.

List of catalogs in the catalog database

The following table lists and describes the catalogs that reside in the catalog database.

Catalog	Name	Description
Applications catalog	AP.CATALOG	Stores information needed in order to start Analyzer, Analyzer modules, or custom applications.
Group catalog	GP.CATALOG	Describes the groups of users defined for this database. Groups are used for scoping, which restricts the members of the group to a portion of the data, and for controlling access to shared objects in public libraries.
Maximum Values catalog	MX.CATALOG	Specifies the maximum number of dimension values for the Across, Down, and Page dimensions in a report, and for the Across, Legend, and Page dimensions in a graph. Also specifies maximum value limits for members in a custom aggregate and for actions in the Selector.
Source catalog	SO.CATALOG	Contains values that you can use to control the behavior of Analyzer. This catalog corresponds to SOCAT in the server database.
Scope catalog	SP.CATALOG	Describes any scopes that are defined for this database. Scopes restrict user access to data.
User catalog	UR.CATALOG	Identifies the users defined for this database, for the purpose of controlling user access to data and to shared objects.

Integrated Catalogs

Definition: integrated catalogs

Catalogs that store information about saved objects such as reports, graphs, forecasts, custom measures, and custom aggregates are referred to as *integrated catalogs*. These catalogs reside in the personal database. In a remote database configuration, integrated catalogs reside both in the personal database and in the public database.

Function of integrated catalogs

Analyzer requires the information in public database catalogs during a user's session in order to fulfill requests for data. Each of these catalogs has a counterpart in the personal database for personal definitions of reports, graphs, and other saved objects. During a user's session, an integrated version of each public database catalog is created in the personal database. The personal database catalogs contain personal saved objects in the private library, base measures in the MASTER library, and saved objects in public libraries to which the user has access.

Note: In thick client mode, a mirror image of the integrated catalogs in the personal database on the remote system is maintained in the personal database on the PC.

List of integrated catalogs

The following table lists and describes the integrated catalogs.

Catalog	Name	Description
Custom Aggregate catalog	AG.CATALOG, PB.AG.CATALOG	Defines each custom aggregate.
Custom Analysis catalog	CA.CATALOG, PB.CA.CATALOG	Required to support compatibility with previous releases of Analyzer.
Component catalog	CO.CATALOG, PB.CO.CATALOG	Describes the component dimensions for reports and graphs.
Data catalog	DA.CATALOG, PB.DA.CATALOG	Describes all base and inferred measures. It corresponds to DACAT in the server database.
Display catalog	DI.CATALOG, PB.DI.CATALOG	Describes the type of document created by a report or graph.
Data Cube or Data View catalog	DV.CATALOG, PB.DV.CATALOG	Stores dimension status used to select data for scopes and slices.
Forecast catalog	FC.CATALOG PB.FC.CATALOG	Stores specific information about each forecast.
Folder catalog	FD.CATALOG, PB.FD.CATALOG	Stores specific information about each defined folder.
Graph catalog	GR.CATALOG, PB.GR.CATALOG	Stores specific information about each defined graph.

Catalog	Name	Description
Map catalog	MP.CATALOG, PB.MP.CATALOG	Reserved for future use.
Slice catalog	SC.CATALOG, PB.SC.CATALOG	Describes the slices defined for this database.
Segment catalog	SG.CATALOG, PB.SG.CATALOG	Identifies the Across, Down, and Page dimensions of a report or graph.
Saved Selection catalog	SL.CATALOG, PB.SL.CATALOG	Defines lists, scripts, and datascripts so that the status of various dimensions can be recalled and used later (e.g., to reproduce a particular report or create a slice).
Table or Report catalog	TB.CATALOG, PB.TB.CATALOG	Stores specific information about each defined report.

Prefix for integrated catalogs in the public database

In the public database, integrated catalogs have a prefix of "PB." For example, the custom aggregate catalog in the personal database is called AG.CATALOG; in the public database it is called PB.AG.CATALOG.

The User Option catalog

In addition to the integrated catalogs, the personal database also contains the User Option catalog (OP.CATALOG). This catalog stores information about user options such as the display of dimensions in the Selector dialog box.

Note: In thick client mode, a mirror image of the User Option catalog in the personal database on the remote system is maintained in the personal database on the PC.

Setting Up Database Access for the Windows Client

Chapter summary

This chapter explains how to set up end user access to a database for the Analyzer Windows client.

List of topics

This chapter includes the following topics:

- [General Instructions for Database Setups](#)
- [Creating a Configuration File](#)
- [Selecting an Access Mode](#)
- [Specifying the Database Path for a Slice](#)
- [Maintaining a Shell Database](#)
- [Setting Up Login to the Database or to a Relational Manager](#)
- [Creating an Icon to Start a Database](#)
- [Setting Up a Workstation User Name](#)
- [Changing the Default Directory for Thin Client Publish to File](#)

General Instructions for Database Setups

About database setup

A *database setup* provides the information that Analyzer needs in order to locate and access files or data sets and open the database. The procedure for defining a database setup includes the following tasks:

- Selecting a configuration file
- For a remote database, defining a communications setup
- For a remote database, specifying information that may be required for login to the server, or to the server and an RDBMS from the server
- For a Windows client installation, creating an optional database icon, in order to simplify starting an application and opening the database

Defining multiple database setups

You will probably define several different database setups as illustrated by the following examples:

- You need at least one setup for each database used at your site.
- If you create a slice (subset) of a remote database on the PC or LAN, you need a setup for the slice.
- If some users require read-only access to a database while others require read/write access (e.g., to refresh a shell database), you need database setups for both access modes.
- If you or your users use thick client mode in the office and thin client mode while on the road to access the same remote database, you need database setups for both client modes.
- To use Database Manager for your DBA tasks on a remote database, you need a database setup for thin client mode.
- If different database icons (e.g., one that starts Analyzer and another that starts Database Manager) are required to open the same database, you can provide a database setup for each icon.

Preparing and distributing database setups

Any database setups that you define are saved in DBSETUP.INI, which is a text file located in your Analyzer user directory. If you want to make database setups available to users, you can distribute the file on disk or via a public LAN directory. When you distribute database setups, you should consider the following:

- By default, DBSETUP.INI includes all of the database setups that you have defined. If you do not want to distribute all of your setups, you can copy DBSETUP.INI and edit it before distributing it.
- Your DBSETUP.INI may include user-specific information such as your database user name or your user ID and password for login to the server or a relational database. You should remove personal information from the file before distributing it.
- Each database setup in DBSETUP.INI references a configuration file (with the file extension .DSC). Any setup for a remote database also references a communications setup. (Communications setups are stored in the COMSETUP.INI file, as described in [Chapter 6](#).) In addition to DBSETUP.INI, you should distribute all referenced configuration files and communications setups.

Procedure: Defining a database setup

To define a database setup, perform the following steps:

1. From the Application Manager File menu, choose **Database Setup**.
Result: The Database Setup dialog box is displayed.
2. To create a new database setup, perform the following steps:
 - a. Choose **Create**.
Result: The Enter New Database Name dialog box is displayed.
 - b. Type a name for the new database setup.
 - c. Choose **OK** to return to the Database Setup dialog box, which now displays the name.
3. Choose **Create**.
Result: The Enter New Database Name dialog box is displayed.
4. Type a name for the new database setup.

5. Choose **OK** to return to the Database Setup dialog box, which now displays the name.
6. In the Configuration frame, choose **Browse** to select or create a configuration (DSC) file for the database, as described in the topic [“Creating a Configuration File” on page 5-6](#).
7. For a remote database, in the Communications box, select the name of the communications setup or choose **Define** to define a communications setup for the database, as described [Chapter 6](#).
8. In the Login frame, you can specify information such as an ID and password, which may be required for login to the following:
 - The remote server
 - The remote server and a remote relational database from the serverThis is described in the topic [“Setting Up Login to the Database or to a Relational Manager” on page 5-16](#). This topic also explains how you can arrange for users to be prompted for login information at each login, as an alternative to providing the information in the database setup.
9. Choose **Save** to save the database setup. When you have finished defining database setups, choose **Close**.

Example: Database Setup dialog box

The following is an example of the Database Setup dialog box for a remote database with login to the remote server.

Database Setup

Name: SATURN

Configuration

Configuration Filename: d:\analyzer\saturn.dsc **Browse...**

Communications: Dillenger-RPC **Define...**

Login:

Server Relational

User ID: GTAYLOR

Password: XXXXXX Domain: WALT-DCNT

Create Database Icon **Setup...**

Save
Close
Create...
Rename...
Delete
Help

Related information

For more information about defining a database setup, search for the following topic in the Application Manager Help system:

“Defining Database Setups”

Creating a Configuration File

Overview of the configuration file

A database setup requires that you select a Data Set Configuration (DSC) file, generally referred to as a *configuration file*. The configuration file specifies the configuration and location of a database, as well as user access to the database.

A configuration file is a text file that you prepare and name with a DSC file extension (e.g., SATURN.DSC). Oracle Corporation provides a sample configuration file (GLOBE.DSC) for the sample Global database. This file and any configuration files that you create reside in the Analyzer directory on the PC.

Contents of the configuration file

The configuration file can contain the following information:

- The access mode, which identifies the database configuration and user access to the database.
- For a slice database, the name of the database file.
- For a remote database, the name of the Remote Database Configuration (RDC) file on the remote system. For information about the RDC file, see [Appendix C](#).
- For a remote database accessed in thick client mode, the location of the shell database, if other than the user directory on the PC.

Important: Mapped drives and path names for slice or shell databases on a LAN

The path name that you enter in the configuration file for a slice database or a thick client shell database on a LAN must refer to a mapped drive. You cannot specify a UNC path such as `\\servername\sharename\subdirectory\dbname.db`.

For example, if you are creating a slice database configuration file for the database, GLOBAL, and the database is located on a machine called “FileServer” under the directory structure `c:\orahome\olap\osannn\program` and the `olap` directory is shared, you should map an available drive, “X” in this case, to the network resource `\\FileServer\orahome\olap`, and create a database configuration that specifies the file `X:\osannn\program\GLOBAL.db`.

Preparing and distributing configuration files

You should prepare one or more configuration files for users to select in the database setup. The easiest way to do this is to use Application Manager to create the configuration file, and then distribute the file to users either on disk or via a public LAN directory.

Requirement for creating a configuration file

When you use Application Manager to create or edit a configuration file, the AllowConfigEdit option in the APPMGR.INI file must be set to its default of YES. If the **Edit** button does not appear in the Select Database Configuration File dialog box, change this setting in APPMGR.INI.

Procedure: Creating a configuration file

To create a configuration file, perform the following steps:

1. In the Configuration frame of the Database Setup dialog box, choose **Browse**. The Select Database Configuration File dialog box is displayed.
2. In the File Name box, type a name for the new configuration file.
Note: A configuration file must have the file extension .DSC (e.g., SATURN.DSC).
3. Choose **Edit**. The Edit Configuration File dialog box is displayed.
4. From the drop-down list in the Access Mode box, select the access mode that applies to the database. For detailed information, see the topic [“Selecting an Access Mode” on page 5-10](#).

Note: The actual label for the Filename box in Step 5 depends on the access mode that you have selected in the Access Mode box. If you have selected an access mode for a slice database, the actual label is Database Filename. If you have selected an access mode for a remote database, the actual label is Remote Configuration Filename.

5. Complete the *filename* box by following the steps below:

IF, in Step 4 . . .	THEN . . .
you have selected an access mode for a slice database,	type the name of the database file in the Database Filename box. This is the name of the catalog database, including the .DB extension (e.g., GLOBE .DB). Include the full path unless you have specified the path in the Personal Express Configuration Manager. For more information, see the topic “Specifying the Database Path for a Slice” on page 5-11.
you have selected an access mode for a remote database,	type the name of the database file in the Database Filename box. This is the name of the catalog database, including the .DB extension (e.g., GLOBE .DB). Include the full path unless you have specified the path in the Personal Express Configuration Manager. For more information, see the topic “Specifying the Database Path for a Slice” on page 5-11. Note: Be sure to use the drive designation that is used by the server’s mapping configuration, as opposed to the designation used by your own network mapping configuration.

6. The Shell Directory box applies only if you have selected an access mode in Step 4 for a remote database with thick client mode.

If you have selected an access mode in Step 4 for a slice database, or for a remote database with thin client mode, the Shell Directory box is dimmed.

The two options for storing the shell are described below:

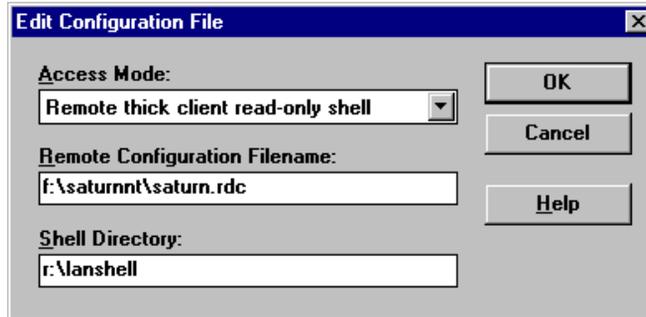
IF you want to . . .	THEN . . .
store the shell in a directory other than the user directory on the PC,	in the Shell Directory box, type the full path of the directory in which the shell will reside (e.g., R : \LANSHHELL). Do not include the name of the shell file itself (e.g., DSATURN .DB).
store the shell in the user directory on the PC,	leave the box blank.

7. Choose **OK** to save the new configuration file and return to the Select Database Configuration File dialog box.

Tip: You can also create a new configuration file by editing an existing file. To do this, copy the existing configuration file (e.g., using Windows File Manager or Explorer) and rename the copy. When you open the new configuration file in Application Manager, it will have the values of the old file, which you can then edit.

Example: Edit Configuration File dialog box

The following is an example of the Edit Configuration File dialog box in Application Manager. In this example, a remote database will be accessed in thick client mode, and the shell will be shared in a directory on the LAN.



The configuration file created in the dialog box in this example would appear as follows.

```
[Database Configuration]
Access Mode=5
Remote Program=f:/saturnnt/saturn.rdc
Shell Directory=r:/lanshell
```

Related information

For more information about creating and editing configuration files, search for the following topics in the Application Manager Help system:

- “Creating Configuration Files”
- “Editing Configuration Files”

For information about the RDC file, see [Appendix C](#).

Selecting an Access Mode

Overview of the access mode

One of the settings in the configuration file is the access mode. The access mode identifies a database as a remote configuration or slice, and defines access rights to the database or shell as read/write or read-only. For a remote database, the access mode also designates whether database access is in thick client mode or thin client mode.

You specify the access mode in Application Manager in the Edit Configuration dialog box.

Types of access modes

The following table lists and describes Windows client access modes.

Access mode	Description
Slice read/write	Slice database with read/write access to the database. When a user builds his or her slice, this access mode is assigned automatically. The designation for this access mode in the DSC file is 1. Note: Ensure that the file attribute of the database is not set as "Read Only" at the operating system level.
Slice read-only	Slice database with read-only access to the database. This is the access mode for a user who will share a slice with other users or who will access a slice that has been built by a DBA. The designation for this access mode in the DSC file is 2.
Remote thick client read/write shell	Remote database with thick client mode and read/write access to the shell. This is the access mode for a DBA who will refresh a shared shell or for a user who will refresh the shell in the user's own area. For more information, see the topic "Maintaining a Shell Database" on page 5-12 . The designation for this access mode in the DSC file is 4.

Access mode	Description
Remote thick client read-only shell	Remote database with thick client mode and read-only access to the shell. This is the access mode for a user who will share a shell with other users across a LAN. For more information, see the topic “Maintaining a Shell Database” on page 5-12 . The designation for this access mode in the DSC file is 5.
Remote thin client read/write	Remote database with thin client mode and read/write access to the database. This is the access mode for thin client users and for DBA tasks using Database Manager. The designation for this access mode in the DSC file is 6. As DBA, ensure that your OS level ID has have read/write access to the remote Express databases at the operating system level.

User access to the personal database

Regardless of the access mode, Analyzer automatically grants read/write access to the user’s personal database.

By default, users’ personal databases are located in users’ home directories on the server where Express Server is installed. You can change this default, as described in [“Remote Personal Databases” on page 4-4](#).

Specifying the Database Path for a Slice

Slice requires Personal Express

In order to access a slice database, a user must have Oracle Personal Express installed on the local machine. The Express Server client installation CD includes Personal Express.

Overview of the database path

One of the settings in the configuration file for a slice database is the name of the catalog database file. You identify the database file name in Application Manager in the Edit Configuration File dialog box.

You must include the full path of the database file name in the Edit Configuration dialog box, unless you have specified the path in the Personal Express Configuration Manager.

Important: The path name that you provide for a shell on a LAN must be a mapped drive. For more information, see [“Important: Mapped drives and path names for slice or shell databases on a LAN”](#) on page 5-6.

Specifying the Catalog database path in Personal Express

To specify the Catalog path with the Personal Express Configuration Manager, start up the Configuration Manager and type the full path name in the ServerDBPath setting on the File I/O tab.

Maintaining a Shell Database

Overview of the shell

When a user accesses a remote database in thick client mode, Analyzer creates a *shell database* (or *shell*) for the user. The name of the shell file is the same as the server database file on the remote system (e.g., DSATURN.DB). The shell contains a local copy of metadata from both the server and catalog databases. You must refresh the shell whenever the remote database has been updated with new data.

Options for maintaining the shell

The following two options are available for setting up and refreshing the shell database:

- **Individual shell** — You can allow each user to maintain and refresh the shell on the user’s PC. If you choose this option, each user must use a database setup that specifies thick client mode and read/write access to the shell. When the user opens the database using this database setup, if the remote database has been updated, the local shell is refreshed automatically.
- **Shared shell** — Because of space constraints, it is sometimes inefficient for each user to refresh the shell in the user’s own area. In this case, you can create and maintain one read-only shell for users to share across a LAN. You refresh the shell by opening the database in thick client mode with read/write access. Then you move the refreshed shell to the LAN as a read-only and shareable file. Users then access the database using a database setup that specifies thick client mode with read-only access to the shell.

Considerations for refreshing a shared shell

Special considerations apply when you refresh a shared shell on a LAN. You must refresh the shell in thick client mode with read/write access. However, if you have created any personal measures, they will be loaded into the shell during the refresh. Subsequently, users trying to access the refreshed shell will get error messages because they do not have these measures in their personal database catalogs. To avoid this problem, create a special “refresh” user name for yourself as DBA. Set up this refresh user name to access the database in thick client mode with read/write access.

Important: Do not create any custom objects using this user name.

Procedure: Refreshing a shared shell (recommended method)

To refresh a shared shell database, perform the following steps:

1. Ensure that no users are currently accessing the shell database. Then, set the network file access flags on the shell database file to read/write and non-shareable.
2. From the Analyzer File menu, open the remote database, using your refresh user name and a database setup that specifies the following:
 - Thick client mode
 - Read/write access to the shell
3. Copy the shell file from your PC to a directory on the LAN using a tool such as Windows Explorer.
4. Flag the shell file as read-only and shareable. It is a good idea to provide a configuration file on the LAN that specifies thick client mode with read-only access, as well as the location of the shell directory. Users can point to this file in their database setups in order to access the shell.

Procedure: Refreshing a shared shell directly on the LAN

Alternatively, you can refresh a shared shell directly on the LAN, although this is not the recommended method. To refresh a shared shell database directory on the LAN, perform the following steps

1. Flag the shell file as read/write exclusively (*not* shareable).

2. From the Analyzer File menu, open the remote database using your refresh user name and a database setup that specifies the following:
 - thick client mode
 - read/write access to the shell
 - the location of the shell directory on the LAN
3. Flag the shell file as read-only and shareable.

Refreshing a remote database at a scheduled time and date

You can use the Open Later dialog box in Application Manager to refresh one or more remote databases automatically at a scheduled time. This process is also known as a “delayed refresh.” For example, you can refresh databases overnight, when the demand for system resources is reduced.

Note: In order to use the Open Later option, your communications setup must include automatic login and logout programs.

Procedure: Refreshing a remote database at a scheduled time and date

To refresh a remote database at a scheduled time and date, perform the following steps:

1. From the Application Manager File menu, choose **Open Later**. The Open Later dialog box appears. The Open Database box is blank and the **Open Later Active** and **Leave Database Open** check boxes are dimmed.
2. In the Open Database box, select a database. The current time and date are displayed.
3. Select the time and date for opening this database. The time that you select must be later than the current time.

Note: You can only select half-hour increments for time. You cannot type a value: use the arrows to make time and date selections.

4. Select **Open Later Active** to activate the delayed refresh.
5. To open another database automatically, repeat Steps 2, 3, and 4. Choose a time that is at least one-half hour later than the time set for the previous database.

Tip: You do not have to know the precise timing for opening multiple databases. Just select a time that is later than the time selected for the previous database. For example, if you selected 10:00 pm as the opening time for the first

database, you can select 10:30 pm for the second database and 11:00 pm for the third database, without being concerned about exact times.

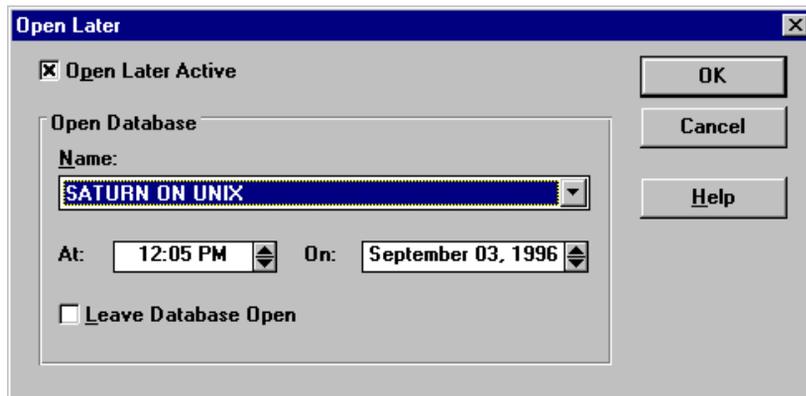
6. If this is the *only* database you are opening, or the *last* database of a series, you can choose **Leave Database Open** to keep the database open.

Important: You can have only one database open at a time. Do not choose this option if you are refreshing more than one database and the database is not the last in the series.

7. When you are done with your work, choose **OK**. Do *not* exit Application Manager because this will disable your Open Later selections.

Example: Refreshing a remote database at a scheduled date and time

The following example shows the Open Later dialog box set to open a database at 12:05. The **Leave Database Open** option is not selected.



Related information

For more information about using delayed refresh, search for the following topics in the Application Manager Help system

“Open Later Dialog Box”

“Opening a Database at a Scheduled Time and Date”

Setting Up Login to the Database or to a Relational Manager

About login

To open a remote database, users may be required to log into the remote server or to a remote RDBMS from the server. Information such as a login user ID and password may be required to permit login.

Methods of providing login information

You can ensure that users provide required login information as follows.

IF you want to . . .	THEN . . .
arrange for all users to be prompted to provide information in the Login dialog box at each login,	change the LoginLogout setting in the Preferences section of the APPMGR.INI file to NO. The boxes for specifying a password and user ID on the Login tabs in the Database Setup dialog box will be dimmed. As long as the appropriate Login tabs are enabled, users will be prompted to provide information at login.
allow users to specify login information in Application Manager in the database setup,	keep the LoginLogout setting in the Preferences section of the APPMGR.INI file set to its default of YES. If a user specifies login information in the database setup, the user will not be prompted to provide the information at each login.

Tabs for specifying login information in the Database Setup dialog box

Users can specify login information on one or two Login tabs in the Database Setup dialog box. Login tabs are as follows:

- Server tab — Applies to a remote database. This tab allows the user to specify information required for login to the remote server.
- Relational tab — Applies to a remote database. This tab allows the user to specify information required for login to a remote RDBMS (from the server).

Procedure: Specifying information for login to the remote server

To specify information in the database setup for login to the remote server, perform the following steps:

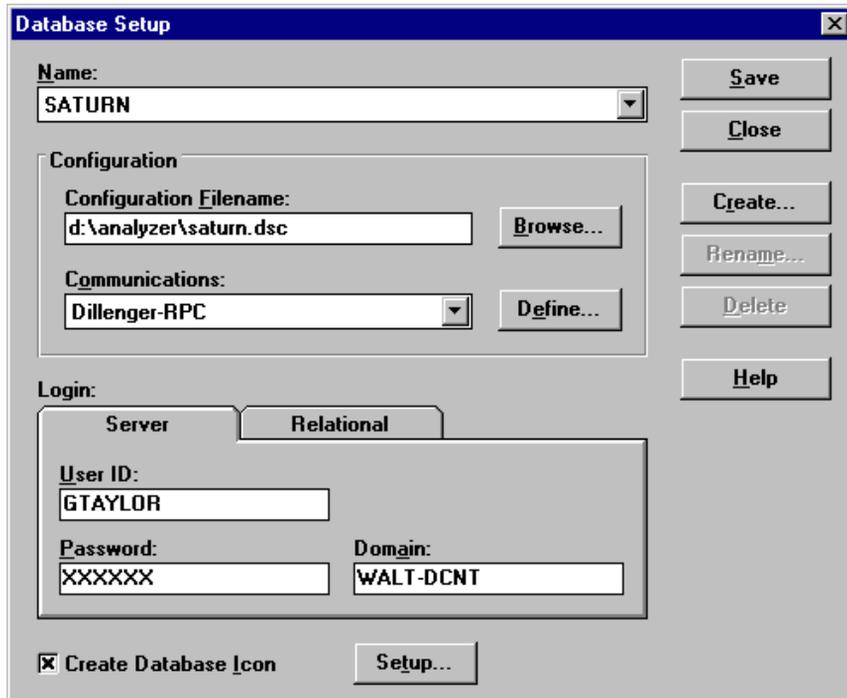
1. In the Login frame of the Database Setup dialog box, select the Server tab.
2. You can take either of the two steps described below:

IF your domain account for Windows NT . . .	THEN . . .
is not set up on the remote computer itself,	in the Domain box, type the domain string that identifies the Windows NT domain in which you have an account.
is set up on the remote computer itself	leave the Domain box blank.

3. In the User ID box, type a user ID for login to the remote server.
4. In the Password box, type a password for login to the remote server.

Example: Remote database setup with login to remote server

The following example shows the database setup for a remote database with information specified for login to a remote server.



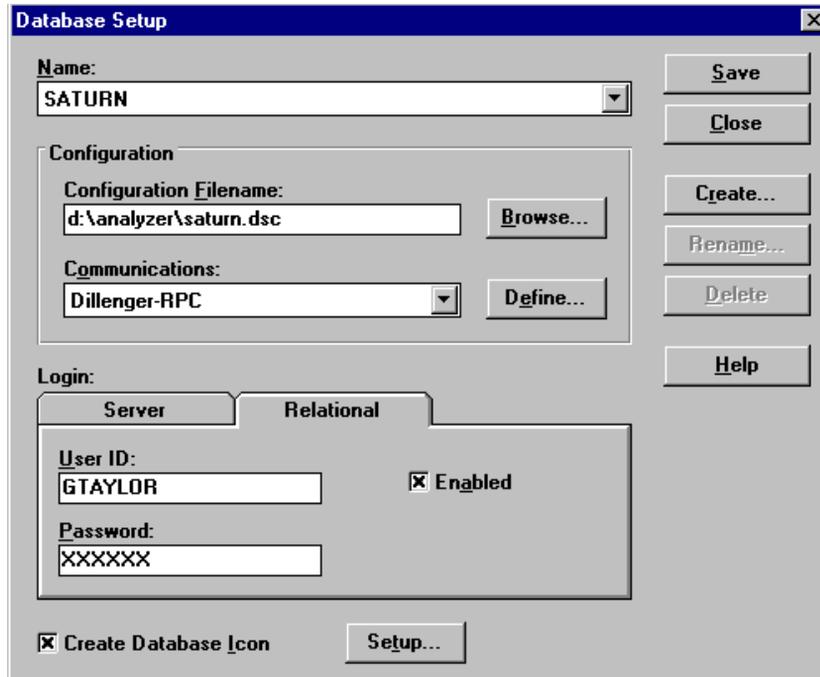
Procedure: Specifying information for login to a remote RDBMS

To specify information in the database setup for login to a remote RDBMS, perform the following steps:

1. In the Login frame of the Database Setup dialog box, select the Relational tab.
2. Select the **Enabled** check box.
3. In the User ID box, type a user ID for login to the remote RDBMS.
4. In the Password box, type a password for login to the remote RDBMS.

Example: Remote database setup with login to a remote RDBMS

The following is an example of the Database Setup dialog box for a remote database. In the example, information is specified for login to a remote RDBMS from the server.



Creating an Icon to Start a Database

Overview of the database icon

If you want to be able to open a database automatically, you can create a database icon in the database setup in Application Manager. A database icon starts Application Manager — or an application such as Analyzer or Database Manager *without* running Application Manager — and then opens a database. Using a database icon to start a specified application speeds application startup and keeps more memory resources available.

Conversely, you may choose not to create a database icon for a particular database setup. For example, as DBA, you may not want an icon for every database that you access. Note, however, that creating a database icon does not *prevent* you from being able to start Application Manager, open a database, and then start an application.

Procedure: Creating a database icon

To create a database icon, perform the following steps:

1. In the Database Setup dialog box, select the **Create Database Icon** check box, and then choose **Setup**. The Create Database Icon dialog box is displayed with the **Start Application Manager** button chosen as the default.
2. You can create several types of database icons, as described in the steps below:

IF you want to create . . .	THEN . . .
a database icon that starts Application Manager,	skip this step and proceed to Step 3.
a database icon that starts an application such as Analyzer without running Application Manager,	choose Start Application, and select the application that you want to start from the Start Application list.

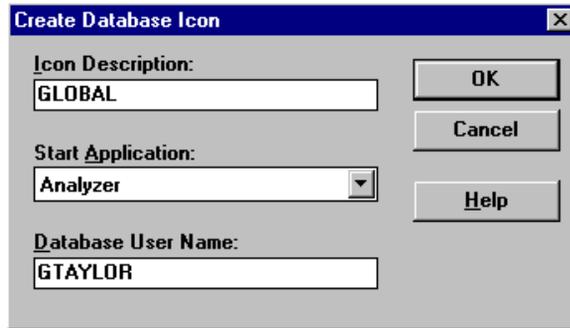
3. In the Icon Description box, type the name that will be displayed with the icon.
4. Decide whether or not to add a user name to the command line, as described below.

IF . . .	THEN . . .
you want to add a user name to the command line for the database icon,	type a user name in the Database User Name box. The user will <i>not</i> be prompted to enter a user name when using this icon to open the database for the <i>first</i> time in a session.
you do not want to add a user name to the command line for the database icon,	leave the Database User Name box clear. The user <i>will</i> be prompted to enter a user name when using this icon to open the database for the first time in a session.

5. Choose **OK**. The Database Setup dialog box is displayed again.

Example: Create Database Icon dialog box

The following is an example of the Create Database Icon dialog box. In this example, Analyzer is the application that will start and open the database. The icon has the database name, GLOBAL, as its description.



Procedure: Creating a database icon by modifying an application icon

You can configure a database icon for a previously defined database by adding an argument to the command line for an application icon.

To create a database icon by modifying an application icon, perform the following steps:

1. In Windows, select the application icon by clicking on it once.
2. From the File menu, choose **Properties**.
3. In the Command Line box, at the end of the command line, type `/DB=DATABASE_NAME ='` where `DATABASE_NAME` matches the application name stored in the `DBSETUP.INI` file.

Example: The following would modify the application icon for Analyzer to open the GLOBAL database:

```
d:\orahome\olap\osannn\program\Analyzer.exe/DB=GLOBAL="
```

Icon files for database icons

Database icons use the icon files that are specified for each application in the QuickStartIcons section of APPMGR.INI. You can use the icon file of your choice for a database icon. For example, to use the icon ANDBS.ICO for a database icon that starts Analyzer, you might enter the following:

```
[QuickStartIcons]
Analyzer=d:\orahome\olap\osannn\program\andbs.ico
```

Related information

For more information about creating database icons, search for the following topic in the Application Manager Help system:

“Creating Database Icons”

Setting Up a Workstation User Name

Prompting for user names

The PromptforUID setting in the APPMGR.INI file determines whether users are prompted to enter their user name when they open a database, as follows:

- If PromptforUID is set to NO (the default), users are not prompted to enter a user name.
- If PromptforUID is set to YES, users are prompted to enter a user name. If user validation has been enabled for the database, the user name must be the same as the user’s operating system name.

To simplify user login when user validation is turned on, Oracle Corporation recommends that you set the PromptforUID setting to NO.

Hiding user names

If the PromptforUID is set to YES, you can conceal the user name with “X’s” as the user enters it at the prompt. This is controlled by the HideUID setting in the Preferences section of the APPMGR.INI file. By default, the user name is visible.

To hide user names, set the HideUID setting in APPMGR.INI to YES.

Adding a user name to the command line

If the PromptforUID setting is set to **NO**, you can add a user name to the command line for an application icon (e.g., an icon that starts an Analyzer Windows client) or a database icon (e.g., an icon that starts an Analyzer Windows client *and* opens a database). With the user name at the command line, the user will *not* be prompted to enter a user name the first time a database is opened in a session. Provided the PromptforUID option is set to **YES**, the user *will* be prompted to enter an ID each time a database is opened subsequently in a session.

Procedure: Adding a user name to the command line for an application icon

To add a user name to the command line for an application icon, perform the following steps:

1. In Windows, select the icon by clicking on it once.
2. From the File menu, choose **Properties**.
3. In the Command Line box, type `/ID username` at the end of the command line, where “username” is the actual ID for the user.

Example: “KBARRY” is added to the command line for the icon for Analyzer.

```
d:\orahome\olap\osannn\program\Analyzer.exe /ID kbarry
```

4. Choose **OK**.

Procedure: Adding a user name to the command line for a database icon

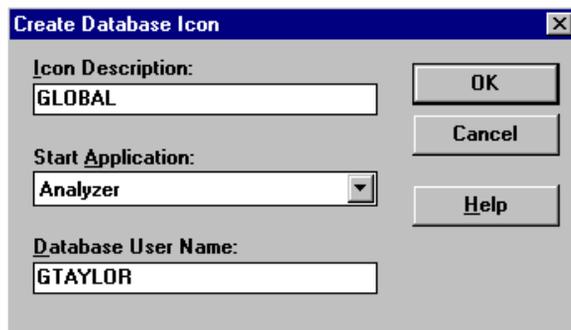
To add a user name to the command line for a database icon, perform the following steps:

1. In Application Manager, in the Database Setup dialog box, select the **Create Database Icon** check box. Then choose **Setup**. The Create Database Icon dialog box is displayed.
2. Type a user name in the Database User Name box
3. Choose **OK**. The Database Setup dialog box is displayed again.

Result: The user name will be added to the command line for that database icon. The user will not be prompted to enter a user name when using this icon to open the database for the *first* time in a session.

Example: Create database icon with user name specified

The following example shows the Create Database Icon dialog box. In this example, the user name “GTAYLOR” will be added to the command line for the database icon.



Changing the Default Directory for Thin Client Publish to File

Default directory

The EXPORT setting in the RDC file indicates the default directory on the server that will be displayed in the File name area in the Maintain Libraries File dialog box when a user who is set up as a thin client publishes an object to a file. By default, the value for EXPORT is set to UserHome. This sets the default path to the user’s home directory on the server where Express Server is installed.

You can change the default by changing the value for the EXPORT property.

Note: If you do not want thin client users to have the ability to publish to a file, delete the EXPORT property from the RDC file.

Procedure: Changing the default directory for thin client publish to file

To change the default for the server directory that a thin client user sees when publishing to a file, perform the following steps:

1. Identify the Remote Database Configuration (RDC) file for the database.

Note: The RDC file is usually located in the directory on the server where the database was built.

2. Using a text editor, edit the path for the EXPORT in the [Paths] section.

Important: If the path that you specify is on another server, the drive letter must be a mapped drive that is specified in the Drive Map tab of Express Instance Manager.

3. Save the RDC file.

Example: RDC file with EXPORT property

The following example illustrates the [Paths] section of an RDC file. The section includes a specification for the default server directory that thin client users will see when they publish an object to a file.

```
[Paths]
code=c:\orahome\olap\oes6xx\common\
data=d:\db\coffee\
extend=e:\db\coffee\
export=d:\export\
personal=d:\users\
```

Setting Up Communications for a Windows Client

Chapter summary

This chapter describes how to set up communications between an Analyzer Windows client and a remote server.

List of topics

This chapter includes the following topics:

- [General Instructions for Communications Setups](#)
- [Communications Setups for Thick Client Access](#)
- [Communications Setups for Thin Client Access](#)

General Instructions for Communications Setups

Overview of the communications setup

A communications setup establishes communications between the PC client and a remote computer. The communications setup is referenced in the database setup. For detailed information about defining database setups, see [Chapter 5](#).

The following table describes the connections that are used for thick client and thin client access, respectively.

IF database access is . . .	THEN . . .
thick client mode,	the transport mechanism between Personal Express and Express Server is a network connection.
thin client mode,	the transport mechanism between Analyzer and Express Server is an RPC (Remote Procedure Call) connection.

Using a modem

Both network and RPC connections are via TCP/IP. If you use a modem to connect to the remote system, you must have modem support with your TCP/IP installation. This usually means that the user must dial into a SLIP or PPP account to access the remote database.

Preparing and distributing communications setups

Any communications setups that you define are saved in the COMSETUP.INI file, which is a text file located in your Analyzer directory on the PC. If you want to make communications setups available to users, you can distribute COMSETUP.INI on disk or via a public LAN directory.

Accessing the Communications Setup dialog box

You can access the Communications Setup dialog box from Application Manager in two ways:

- From the File menu, choose **Communications Setup**.
- or
- In the Database Setup dialog box, choose **Define**.

Dialog box components for thick or thin client access

In the Communications Setup dialog box, you first select the type of access for which you are defining communications: thick client or thin client. In the Settings frame, you then specify the settings for the connection (network or RPC) that applies to that type.

If you selected thick client mode, components in the Settings frame are enabled for specifying a network connection. If you selected thin client mode, components in the Settings frame are enabled for specifying an RPC connection.

Communications Setups for Thick Client Access

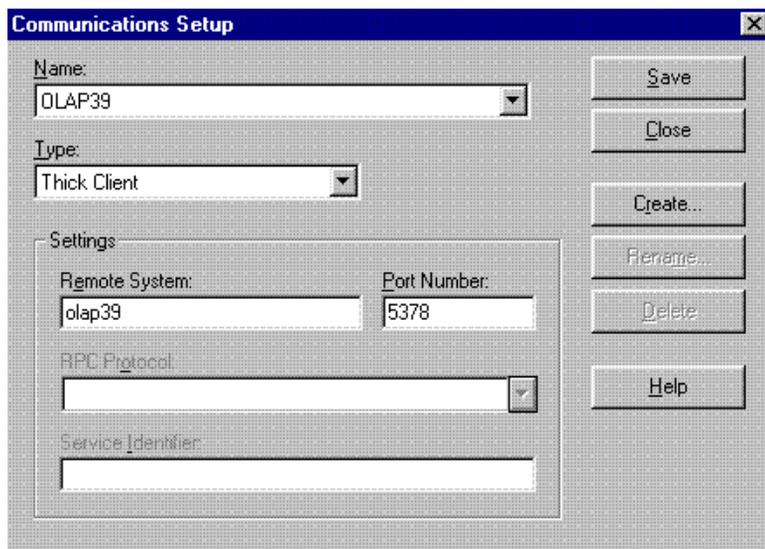
Procedure: Defining a communications setup for thick client access

To define a communications setup for thick client mode and a network connection, perform the following steps:

1. From the Application Manager File menu, choose **Communications Setup**. Alternatively, in the Database Setup dialog box, choose **Define**.
2. To create a new communications setup, perform the following steps.
 - a. Choose **Create**.
Result: The Enter New Communications Name dialog box is displayed
 - b. Type a name for the communications setup.
 - c. Choose **OK** to return to the Communications Setup dialog box, which now displays the name.
3. In the Type box, select **Thick Client**. This enables components in the Settings frame that are appropriate for specifying a network connection.
4. In the Remote System box, specify the host DNS name or IP address for the remote system.
5. In the Port Number box, type the port number for your remote system. The port number can range from 5000 to 15000.
6. Choose **Save** to save the communications setup that you have defined. When you finish defining communications setups, choose **Close** to exit the Communications Setup dialog box.

Example: Communications Setup dialog box for thick client access

The following is an example of the Communications Setup dialog box for thick client mode and a network connection.



Creating a communications log

If you are using thick client mode and a network connection and are having difficulty accessing the remote server, it might be useful to create a communications log file. A communications log file captures all data that is sent to and from the remote server.

To create a communications log file, set the ConnectDebug setting in the Communications section of the APPMGR.INI file to YES. This automatically creates a communications log file named XTLDEBUG.LOG.

Viewing the communications log file

To view the communications log file, type SHOWLOG at the DOS prompt. SHOWLOG.EXE is located in the Analyzer program directory on the PC.

Adjusting the timeout

If you are using thick client mode and a network connection, the interface between Personal Express on the PC and Express Server on the remote system is XCA (Express Communications Architecture). Express uses a parameter that controls the interval before an XCA timeout.

By default, the timeout parameter is set at 30 seconds. Timeout problems may occur if there are unusually long SQL queries to the database. If so, you can adjust the XCA timeout parameter to allow a longer interval before timeout occurs. To change the default for the timeout parameter, add an XCATimeout parameter to the Communications section of the APPMGR.INI file and set it to the desired number of seconds.

Example: Changing XCA timeout by modifying the APPMGR.INI file

The following example adds a parameter for XCA timeout to the Communications section of the APPMGR.INI file.

```
[Communications]
ConnectDebug=No
XCATimeout=40
```

Related information

For more information about setting up communications for thick client access, search for the following topic in the Application Manager Help system:

“Defining Communications Setups for Thick Client Mode and a Network Connection”

Communications Setups for Thin Client Access

Procedure: Defining a communications setup for thin client access

To define a communications setup for thin client mode and an RPC connection, perform the following steps:

1. From the Application Manager File menu, choose **Communications Setup**. Alternatively, in the Database Setup dialog box, choose **Define**.

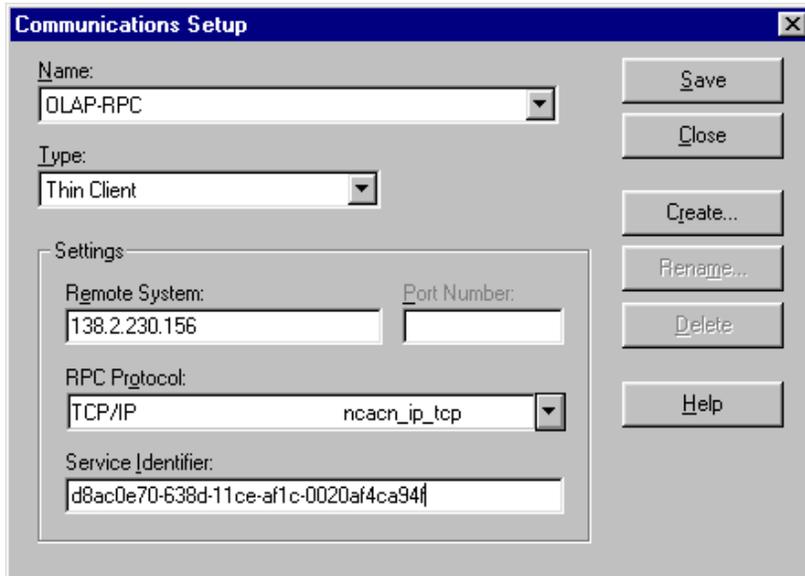
2. To create a new communications setup, perform the following steps.
 - a. Choose **Create**.

Result: The Enter New Communications Name dialog box is displayed.
 - b. Type a name for the communications setup.
 - c. Choose **OK** to return to the Communications Setup dialog box, which now displays the name
3. In the Type box, select **Thin Client**. This enables components in the Settings frame that are appropriate for specifying an RPC connection.
4. In the RPC Protocol box, select the appropriate protocol from the drop-down list. For example, if your remote system is a Unix system, select **Remote Operations** (ora_ro_tcp).
5. If you selected Remote Operations (ora_ro_tcp) as the protocol in Step 4, then type the port number of the Remote Operations Listener in the Port Number box.
6. If you selected a protocol other than Remote Operations in Step 4, then type a service identifier for Express Server in the Service Identifier box.

The service identifier is also referred to as a universal unique identifier (UUID). If more than one instance of Express Server is running on the remote system, then the UUID specifies the particular instance.
7. Choose **Save** to save the communications setup that you have defined. When you finish defining communications setups, choose **Close** to exit the Communications Setup dialog box.

Example: Communications Setup dialog box for thin client access

The following is an example of the Communications Setup dialog box for thin client mode and an RPC connection to an NT system.



Related information

For more information about setting up communications for thin client access, search for the following topic in the Application Manager Help system:

“Defining Communications Setups for Thin Client Mode and an RPC Connection”

Managing User Access to a Remote Database

Chapter summary

This chapter describes how to manage user access to a remote Analyzer database.

List of topics

This chapter includes the following topics:

- [Establishing User Access](#)
- [Working with Groups](#)
- [Working with Users](#)
- [Importing User Names from Oracle Applications](#)
- [Working with Libraries](#)
- [Implementing Security](#)

Establishing User Access

Set up groups, users, libraries, and scopes

Groups, users, libraries, and scopes provide the framework for setting users' access to an Analyzer database.

- **Groups** — Collections of users that establish members' rights to data and to public libraries. A special group can perform administrative (DBA) functions. For more information, see [“Working with Groups” on page 7-2](#).
- **Users** — Unique identifiers that allow access to an Analyzer database. For more information, see [“Working with Users” on page 7-8](#).
- **Libraries** — Public repositories that enable authorized groups to have access to shared objects. For more information, see [“Working with Libraries” on page 7-12](#).
- **Scopes** — Predefined sets of dimension values that can be used to limit group access to data. For more information, see [Chapter 9](#).

Set up security

You set several Express and Analyzer parameters that determine the level of security under which users will access the system. For more information, see [“Implementing Security” on page 7-17](#).

Working with Groups

About groups

A group is a collection of Analyzer users. The group to which a user belongs determines his or her access rights to the MASTER library and to other public libraries. A user's group assignment also determines whether the user has access to all data or to a scoped portion of the data. A special Administrator group enables administrative (DBA) access to the database.

Analyzer stores information about each group in the Group catalog (GP.CATALOG). This information includes the group name and description, access rights to libraries, and association with a defined scope.

Group access modes

The Access Mode that you assign to a group determines which applications the users who belong to the group can use and their mode of access. You can assign one of the following access modes to a group:

- Administrator
- Standard
- Read Only Web

Administrator access mode

Users who are in the group that is assigned the Administrator access mode have administrative (DBA) privileges in the Analyzer database. Administrators can use Database Manager in the Windows interface and Web Administrator in the Web interface. Administrators can also use Application Manager in the Windows interface and Analyzer in the Windows or Web interface. Users with Administrator access have individual personal databases.

You must assign the Administrator access mode to a single group. Oracle Corporation recommends that you define the group that will have Administrator access *before* you define other groups. Grant this group read/write access to the MASTER library and to new public libraries as you create them. This will enable DBA users to copy documents and other objects to the MASTER library and other public libraries. Do not assign a scope to the group with Administrator access; DBA users require access to the entire database.

Standard access mode

Users who are in a group that is assigned the Standard access mode have standard user privileges in the Analyzer database. Standard users can use Application Manager in the Windows interface and Analyzer in the Windows or Web interface. Standard users have individual personal databases.

You can assign the Standard access mode to multiple groups. You can associate a group with a scope, or allow a group to have access to the entire database. You can grant a group read-only or read/write access to the MASTER public library and read/write or read-only access to other public libraries.

Read Only Web access mode

Users who are in a group with Read Only Web access mode have read-only privileges in the Analyzer database. Read Only Web users can use Analyzer through the Web interface. They do not have a personal database and cannot create and save new objects.

You can assign the Read Only Web access mode to multiple groups and associate the groups with scopes and libraries. Library access is always read-only.

Procedure: Creating a group

To create a new group, perform the following steps:

1. Open Database Manager.
2. From the Catalogs menu, choose **Maintain Groups**.
Result: The Maintain Groups dialog box opens.
3. Choose **Create** to define the group.
Result: The Create Group dialog box appears.
4. Enter a name for the group (up to 16 characters long without spaces or leading numbers) and choose **OK**.
Result: You are returned to the Maintain Groups dialog box. The name that you entered is displayed in the Name box.
5. The Include User Names box lists all users who have been defined in the database and displays the group assignment for each user. To add a user who is in another group to the current group, highlight the use name in the Include Users box and click the **Add User** button
Note: The group that you create must have at least one user.
6. In the Scope box, you can select a previously defined scope. If scopes have not yet been defined, you can do this at a later time.
Important: If this is the Administrator group, ensure that the setting in the Scope box is set to **(none)**. The Administrator group cannot be scoped.

7. The Include Library box displays the MASTER library. To set access to this library, proceed as follows:
 - a. In the Include Library box, double-click **MASTER** or highlight **MASTER** and choose **Add Library**.
 - b. **Result:** The Library Access Mode box opens.
 - c. Select the group's access to the library:
 - * **Read Only** — Users in the group have access to all saved objects that belong to the library. They cannot publish personal objects to the library.
 - * **Read/Write** — Users in the group have access to all saved objects that belong to the library. They can publish personal objects to the library.

Important: If this is the Administrator group, grant read/write access to the MASTER library.
 - d. Click **OK**.

Result: You are returned to the Maintain Groups dialog box.
8. If other libraries have been defined, you can set the group's access to these libraries. To do this, select the library in the Include Library box and proceed as described in Step 7. If libraries have not been defined, you can do this at another time.

Important: If this is the Administrator group, grant read/write access to each library.
9. In the Access Mode box, select the group's access mode. The access mode determines which applications the group members can use and their mode of access. You can choose one of the following options:
 - **Administrator** — Users have access to Analyzer via the Windows client or the Web interface. Each user has an individual personal database. Users also have access to Database Manager and OSA Web Administrator. The group should not be scoped and should have read/write access to all libraries.
 - **Standard** — Users have access to Analyzer and Application Manager via the Windows client or the Web interface. Each user has an individual personal database. Scoping and public library privileges apply.
 - **Read Only Web** — Users have read-only access to Analyzer via the Web interface. Users do not have personal databases. Scoping and library

privileges apply, but access to public library documents and objects is read-only.

For more information about access modes, see [“Group access modes” on page 7-3](#).

Important: Only one group can be designated as the Administrator group.

10. Choose **Save** to save the group definition.

Note: If any user in the User Names box has the “unassigned” status, you will not be allow to save the group. You must assign each user to a group.

Maintaining groups

You can create new groups. You can also use the Catalog, Maintain Groups option in Database Manager to perform the following maintenance functions for groups:

- Add users to a group.
- Change a group’s access rights to the MASTER public library. Also set or change access rights to other public libraries.
- Set or change a group’s association with a scope.
- Change the access type (Administrator, Standard, or Read Only Web) for a group. Note that there can be only one Administrator group.
- Rename a group.
- Delete a group.

Example: Maintaining Groups

The following example shows the Maintain Groups dialog box with a sample entry for the EAST group. Users in this group have Standard access mode to the database

and no scope, with read-only access to the MASTER library, and read/write access to the EAST library.

Maintain Groups

Name: EAST

Save Close

Include Users

JEDWARDS	SOUTH
JJONES	DBA
MFCARPENTER	WEST
MSMITH	EAST

Add User Remove User

Create... Rename... Delete Help

Scope: (none)

Include Libraries

EAST	RW
MASTER	RO
NORTH	
SOUTH	

Add Library Remove Library

Access Mode

Administrator

Standard User

Read Only Web

Modified: 10:44AM 04/23/2002

Related information

For more information about maintaining groups, search for the following topics in the Database Manager [Database Maintenance] Help system:

- “Adding Users to Groups”
- “Creating Groups”
- “Creating the Administrator Group”
- “Deleting Groups”
- “Removing Users from Groups”
- “Renaming Groups”

Working with Users

About users

The user name identifies an Analyzer user as follows:

- It controls the user's access to the database. If user validation is enabled, users with invalid user names will be prohibited from opening the database.
- If the user belongs to a group with Standard or Administrator access mode, the name keeps track of personal objects, such as reports, graphs, forecasts, custom aggregates, and custom measures, that the user creates and saves within an application

Analyzer stores information about each user in the User catalog (UR.CATALOG). The information includes a user name and description, and an affiliation with a defined group.

Procedure: Creating a user

To create a new user, perform the following steps:

1. Open Database Manager.
2. From the Database Manager Catalogs menu, choose **Maintain Users**.
Result: The Maintain Users dialog box appears.
3. Choose **Create**.
Result: The Create User dialog box opens.
4. Enter a user name (up to 32 characters without spaces or leading numbers) and choose **OK**.
Note: You are returned to the Maintain User dialog box.
5. Type in a full user name or choose **Import User** to import the user name from the host system or from Oracle Applications.

Note: If you import a user from the host system or from Oracle Applications, the user's name will no longer be available on an import list. This also applies if you enter the name manually.

Important: If your site will be using the single sign-on feature to access Analyzer through Oracle Applications, you *must* import user names from Oracle Applications. For more information, see [“Importing User Names from Oracle Applications” on page 7-11](#).

6. You can enter optional descriptive text in the Information box.
7. If you have previously defined groups, you must associate the user with a group by selecting the group in the Group box. If you have imported the user, you select the group when you choose the user name to import.

Important: A user can be a member of only one group.

8. To turn User Validation off or on for the database, check or disable the **Enable User Validation** box. See [“Full security” on page 7-17](#) for information about this important security setting.

You only have to set this parameter once. The User Validation setting applies to all previous and future users that you enter.

Important: In order to enable user validation, you must have a group that has the Administrator access mode.

9. Choose **Save** to save your work.

Note: If groups have been defined, you cannot save until you associate the user with a group.

Maintaining users

You can create new users. You can also use the Catalog, Maintain Users option in Database Manager to perform the following maintenance functions for users:

- Assign a user to a different group.
- Change free text information for a user.
- Delete a user.
- Change the security parameter for user validation.

Example: Maintaining users

The following example shows the Maintain Users dialog box with a sample entry for user JJONES. This user is associated with the SALES group. User Validation is enabled for the database.

The screenshot shows the 'Maintain Users' dialog box. The 'Name' field is set to 'JJONES'. The 'Full Name' field contains 'Jennifer Jones'. The 'Information' field contains 'Sales Manager, Eastern Region'. The 'Group' dropdown is set to 'SALES'. The 'Enable User Validation' checkbox is checked. The right side of the dialog features buttons for 'Save', 'Close', 'Create...', 'Delete', 'Import Users...', and 'Help'. A 'Modified:' field is located at the bottom right.

Related information

For more information, search for the following topics in the Database Manager [Database Maintenance] Help system:

- “Defining User Identifications”
- “Deleting User Identifications”
- “Modifying User Identifications”

Importing User Names from Oracle Applications

When to import user names from Oracle Applications

If Web users will access Analyzer through the Oracle Applications single sign-on feature, you must import the names of these users from Oracle Applications. If you attempt to create these users manually or import them from the host system, the users will not be able to access Analyzer through single sign-on.

Note: If user validation is enabled for the database, users who have been imported from Oracle Applications will only be able to access Analyzer through the Web interface. They will not be recognized if they attempt to access Analyzer through the Windows client interface.

DBA users in single sign-on environment in which user validation is enabled

If Web users will access Analyzer through the Oracle Applications single sign-on feature and user validation is enabled for the database, an individual who is a DBA should maintain a separate DBA user account that is associated with his or her name on the host system. This will allow the administrator to access Application Manager and Database Manager to perform DBA tasks that are available *only* through the client environment, such as setting up users, groups, libraries, and scopes, and customizing the database.

Prerequisites for importing user names from Oracle Applications

In order to import user names from Oracle Applications, you must have performed several preliminary set up tasks relating to single sign-on. See the topic [“Implementing Single Sign-on” on page 8-3](#) for more information.

Procedure: Importing user names from Oracle Applications

To import user names from Oracle Applications, perform the following steps.

1. Open Database Manager.
2. From the Catalogs menu, choose **Maintain Users**.
Result: The Maintain Users dialog box is displayed.
3. In the Maintain User dialog box, choose **Import User**.
Result: The Import User dialog box is displayed.

4. Choose the **Oracle Applications** tab.
5. Enter the following information in the Oracle Applications tab:
 - User ID — Your ID for the Oracle Apps public schema or the Oracle Apps schema.
 - Password — Your password for the Oracle Apps public schema or the Oracle Apps schema.
 - Alias — Connection alias for the Oracle Applications instance (matching entry in TSNAMES.ORA on the host).
 - FNDNM — Name of the public schema. This information is only necessary if security for the Apps schema has been raised.
6. Choose **OK**.

Result: The list of Oracle Application users is displayed.
7. Select one or more users and choose **OK**.

Working with Libraries

About libraries

A *library* is a collection of saved objects such as reports, graphs, forecasts, folders, custom aggregates, custom measures, and saved selections.

Libraries exist as a property that is associated with saved objects in the integrated catalogs. For example, in the Custom Aggregate catalog, the `::LIBRARY::` property identifies libraries to which each custom aggregate belongs. Analyzer appends a prefix to the `::LIBRARY::` property, which identifies the database configuration as remote (MF). For example, a custom aggregate in the Sales library would be associated with `MF.SALES`.

Types of libraries

Analyzer supports three types of libraries: private libraries, a MASTER public library, and other public libraries which are defined by an administrator.

Private libraries

Private libraries contain objects that have been saved by individual users. Each user who belongs to a group that accesses Analyzer in Standard or Administrator mode has read/write access to a private library in the user's personal database.

MASTER public library

Every Analyzer database includes a MASTER public library. By default, all groups have automatic read-only access to the MASTER library. This enables users to work with documents that have been published (copied) to the library.

As an administrator you can specify whether a group's access to the MASTER library is read-only (users in the group can view and use objects in the library, but cannot publish to the library) or read/write (users in the group can view objects in the library and can publish to the library).

Important: Grant the group that has the Administrator access mode read/write access to the MASTER library. This will enable users in this group to publish data cube definitions (used in scoping and slicing), as well as other saved objects.

Other public libraries

As an administrator, you can create additional public libraries and set group access to them. Access can be read-only (users in the group can view and use objects in the library, but cannot publish to the library) or read/write (users in the group can view objects in the library and can publish to the library).

For example, you might create an EASTERN SALES library and give the Eastern Sales group read/write or read only access to this library.

Important: Grant the group that has the Administrator access mode read/write access to *all* libraries. This will enable administrators to publish data cube definitions (used in scoping and slicing), as well as other saved objects.

Important: Align libraries with scopes

If you intend to apply scoping to user groups, Oracle Corporation recommends that you create a unique library for each scoped group. This will enable users to access custom measures and custom aggregates that are unique to the group's scope.

As of Analyzer Version 6.32, access to custom aggregates and custom measures is controlled through library access, rather than via scoping. A user is granted access to a published custom aggregate or custom measure based on whether or not he or she is a member of a group that has access to the library to which the object has

been published. (In Version 6.31 and all prior versions, a user was granted access to custom aggregates and custom measures based on whether or not these objects had been added to the user's scope.)

You must ensure that you align libraries and scopes so that users access only those custom aggregates and custom measures for which they have the requisite base data. For more information, see [“Note: Publishing for scoped users” on page 7-16](#).

Procedure: Creating a library

To create a new library, perform the following steps:

1. From the Database Manager Catalogs menu, choose **Maintain Libraries**.
Result: The Maintain Libraries dialog box is displayed.
2. In the Name box, choose **Create**.
Result: The Create Library dialog box opens.
3. Enter a name for the library (up to 16 characters without leading numbers or spaces) and choose **OK**.
Result: You are returned to the Maintain Libraries dialog box.
4. Optionally, type a full description and a short, descriptive alias in the Description box and the Alias box, respectively.
5. Choose **Save** to save your work.

Granting library access to groups

Once you have defined a library, you can associate the library with one or more groups. This gives each user who is assigned to the group access to the objects in the library. You can grant read-only or read/write access. For more information, see [“Procedure: Creating a group” on page 7-4](#).

Maintaining libraries

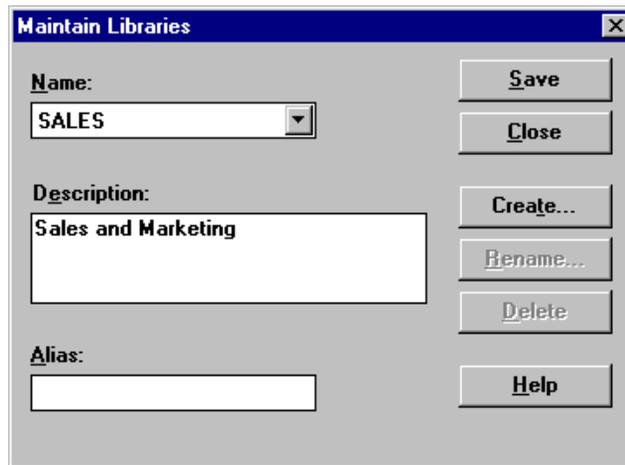
You can create new libraries. You can also use the Catalog, Maintain Libraries option in Database Manager to perform the following maintenance functions for libraries:

- Rename a library.
- Delete a library.

- Change the description for a library.
- Change the alias for a library.

Example: Libraries

The following example shows the Maintain Libraries dialog box in Database Manager with entries for the SALES library.



Publishing objects to public libraries

As an administrator who has access to all public libraries, you can publish objects to appropriate libraries. You can publish reports, graphs, folders, forecasts, custom aggregates, custom measures, data cubes, saved selections, and slices. Users will have access to these objects based on the library permissions that have been set for their group.

Users who have individual personal databases can share their saved objects by publishing them to a public library to which they have read/write access.

The following guidelines apply to publishing:

- The user must have read/write access to the public database at the operating system level.
- In order for Window client users to publish saved objects to a shared file on the LAN, that file must be flagged as read/write and shareable.

- If you create objects to publish to a public library as administrator, remember that your personal definition overrides the public definition. To test objects that you have published to the public database, use another ID and set up a personal database that does not have saved definitions.

Note: Publishing for scoped users

If you intend to apply scopes to user groups and you want scoped users to have access to custom aggregates and custom measures, you must publish these objects only to libraries to which the users have access. Custom aggregates and custom measures are not automatically included in a scope's data cube.

If you do not set up scopes to align with libraries, problems will occur. For example, assume that you have a user base that is divided between Eastern US and Western US. You set up an appropriate scope for each group and give both groups the ability to publish to the same library. If an eastern user created a custom aggregate, "Major Eastern Cities," with the members "New York," "Boston," and "Atlanta" and published the aggregate to the common public library, then all users, including those in the Western group would see that custom aggregate appear on their geography pick list. If a western user then tried to include the custom aggregate in a report, he or she would receive an error message stating that the data for the custom aggregate was not available to them.

If a scoped group has been granted read/write access to a public library, users in the group can publish custom aggregates and custom measures to the library; however a custom aggregate that has been saved as a script in a scoped dimension will be published as a list.

Related information

For more information about creating and maintaining libraries, search for the following topics in the Database Manager Help system:

- "Creating Libraries"
- "Deleting Libraries"
- "Renaming Libraries"
- "Setting Library Access"

For information about publishing objects in Analyzer, see the *Oracle Sales Analyzer Users Guide*.

Implementing Security

About security options

You can set up an Analyzer database to run under full security, standard security, or unidentified guest access.

Full security

Under full security, each user is validated when he or she logs into Express Server (or Oracle Applications). If the user is in the Analyzer User catalog (UR.CATALOG), the user's group assignment determines access mode, data scope, and library permissions. If the user is not recognized, then the system refuses access.

You can implement full security for an installation that supports Windows client users, Web users, or a combination of Windows client and Web users. You can set up groups that access the system in Administrator, Standard, or Read Only Web mode.

Implementing full security

To implement full security for your database, proceed as follows:

- Set up users, groups, scopes, and libraries. When you set up users, enable the **Enable User Validation** checkbox. For information about setting user validation, see [“Procedure: Creating a user” on page 7-8](#).
- Use Express Instance Manager to set the RequiredSecurityLevel parameter for Express Server to HOST or DOMAIN. For more information, see the *Express Installation and Configuration Guide*.
- Use Express Instance Manager to set the RequiredPassword Encryption parameter for Express Server to NO. For more information, see the *Express Installation and Configuration Guide*.
- If any users will access Analyzer through the Web interface, use Express Instance Manager to set the WebAuthenticationType parameter for Express Web Agent to BASIC or FORM. Depending on your Web listener, you might have to use the FORM setting. For more information, refer to the documentation for your Web listener.

Standard security

Under standard security, each user is validated when he or she logs into Express Server (or Oracle Applications). If the user is in the Analyzer User Catalog (UR.CATALOG), the user's group assignment determines access mode, data scope, and library permissions. If the user is not in the catalog, he or she has access to the entire database, with read-only access to the MASTER library.

You can implement standard security for a installation that supports Windows client users, Web users, or a combination of Windows client and Web users. You can set up groups that access the system in Administrator, Standard, or Read Only Web mode.

Implementing standard security

To implement standard security for your database, set the following parameters:

- Set up users, groups, scopes, and libraries. When you set up users, disable the **Enable User Validation** checkbox. For information about setting user validation see [“Procedure: Creating a user” on page 7-8](#).
- Use Express Instance Manager to set the RequiredSecurityLevel parameter for Express Server to HOST or DOMAIN. For more information, see the *Express Installation and Configuration Guide*.
- Use Express Instance Manager to set the RequiredPassword Encryption parameter for Express Server to NO. For more information, see the *Express Installation and Configuration Guide*.
- If any users will access Analyzer through the Web interface, use Express Instance Manager to set the WebAuthenticationType parameter for Express Web Agent to BASIC or FORM. Depending on your Web listener, you might have to use the FORM setting. For more information, refer to the documentation for your Web listener.

Unidentified guest access

Unidentified guest access allows any user who knows the URL access to the entire Analyzer database in guest mode.

You can implement unidentified guest access for an installation that supports Web users only. All users access Analyzer in Read Only Web access mode. User names, scopes and libraries do not apply.

Implementing unidentified guest access

To implement unidentified guest access, set the following parameters:

- Create a single GUEST user group and give the group Read Only Web access. Create a single user for the GUEST group and enter the Express Server OESGUEST user ID as the user name. Enable or disable the **Enable User Validation** checkbox as appropriate. (Do not forget to create a group that has Administrator access and put yourself and any other administrative users in this group.)
- Use Express Instance Manager to set the RequiredSecurityLevel parameter for Express Server to NONE. For more information, see the *Express Installation and Configuration Guide*.
- Use Express Instance Manager to set the WebAuthenticationType parameter for Express Web Agent to NONE.

Setting Up Database Access for Web Users

Chapter summary

This chapter explains how to set up end user access to a remote Analyzer database in the Web environment.

List of topics

This chapter includes the following topics:

- [About Web Access](#)
- [Defining Virtual Directories in the Web Listener](#)
- [Implementing Single Sign-on](#)
- [Setting Up the Analyzer Home Page](#)
- [Customizing the Web Application](#)
- [Providing Web Access to Users](#)

About Web Access

Summary of functions

Web users can work with documents, custom aggregates, custom measures, saved selections, and folders in a similar manner to Windows client users. Additionally, the Web component provides access to the Analysis Library, a series of documents which enable users to conduct pre-defined business analyses.

The extent of Web users' access depends on you how you set up security, groups, scopes, and libraries. You can set up Web users so that they can create and save Analyzer objects, or you can limit Web users to working with existing objects. You can also set up Web users to access the entire database or a scoped portion of the data.

Preliminary requirements

The administrative functions described in this chapter are specific to configuring Web access. Before you perform these tasks, you must establish database security and use Database Manager in the Windows client environment to set up users, groups, libraries, and data scopes. These tasks are described in [Chapter 7](#).

Server requirements

In addition to the standard Analyzer server components, the following components are required for Web access:

- Express Server, including Oracle Web Agent
- A supported Web listener, such as Apache
- If you will implement single sign-on through Oracle Self-Service Applications, Oracle Applications must be installed

For detailed information, refer to the *Oracle Express Web Products Installation Guide* and the release notes for Oracle Web Agent.

Browser requirements

Each user must have a supported browser installed on the PC. For information about Web browser requirements, refer to the Oracle Web Agent Release Notes.

Defining Virtual Directories in the Web Listener

About virtual directories

You must define two virtual directories in your Web listener and map these to corresponding physical directories:

- Directory for Analyzer Web Administrator — Define the virtual directory `osadba` and map it to the appropriate physical directory, for example, `f:\orahome\olap\osa6xx\dba`.
- Directory for Analyzer Home page — Define the virtual directory `osaweb` and map it to the appropriate physical directory, for example, `f:\orahome\olap\osa6xx\osaweb`.

Mapping virtual directories

The file, `virtdirs.txt`, identifies the physical directories to which the virtual directories should be mapped. This file is located in the Oracle home directory on the server where the Web server is installed.

Implementing Single Sign-on

About single sign-on

Single sign-on is a CGI program that allows authenticated Oracle Self Service Application users to access Analyzer without additional security challenges.

If your site will be implementing single sign-on for Web users, you must perform several preliminary installation and setup tasks.

Requirements for implementing single sign on

Install the AOL patch

If you are using Oracle Applications, Version 11.52 or an earlier version, you must install a special AOL patch for Sales Analyzer. The patch updates Oracle Applications with Analyzer single sign-on metadata (roles, menus, functions, and so forth). The AOL patch is available on Metalink.

Install single sign-on

You must install single sign-on on the same machine where the Web server is installed. For more information, see the *Oracle Sales Analyzer Installation and Upgrade Guide*.

Set properties in Oracle Express Instance Manager

You must set the following properties in Oracle Express Instance Manager to NONE:

- RequiredSecurityLevel — Set this parameter on the Security (General) tab.
- RequiredAuthenticationLevel — Set this parameter on the Security (More) tab.
- WebAuthenticationType — Set this parameter on the Web Agent tab.

Set up users and responsibilities in Oracle Applications

You must define each user in Oracle Applications and assign user responsibilities. You can associate a user with one or both of these responsibilities:

- OSA DBA (Administrator) — Enables the user to access Oracle Web Administrator to set up and modify the Analyzer Home page.
- OSA User — Enables the user to access the Analyzer Home page and work with reports, graphs, and other objects.

Important: If user validation is enabled for the database, an individual who is a DBA should have a *separate* user account that is associated with his or her operating system name. This should be different than the single sign-on account. The DBA will use this account to perform tasks that are available only through Database Manager, a Windows client application. Database Manager includes functions for setting up users, groups, libraries, and scopes, and customizing dimension names. The OSA DBA responsibility enables a user to perform administrative functions that are related to the Web, it does not allow an administrator to perform DBA tasks that are available *only* through the Windows client.

Set up profiles in Oracle Applications

You must enter an Analyzer access path profile for each user responsibility. The profile consists of the path to the Analyzer Home page for that user type. You must define two profiles: one for the Analyzer Web Administrator page and one for the Analyzer Home page.

Use the name of the server and the port, if applicable, when specifying the path, as shown in the following examples:

- `//myservername:80/osadba/index.html`
- `//myservername:80/osaweb/index.html`

Modify the RDC file for the database

Analyzer automatically creates a file with the extension RDC when you build a remote database. You must insert a line in the file that specifies the directory in which to store users' personal data. Each user's data will be stored underneath this directory in a separate subdirectory. If the subdirectory does not exist, it will be created when the user logs in.

The following example illustrates the [Paths] section of an RDC file. The section includes an inserted line, shown in bold, that specifies the directory for users' personal data.

```
[Paths]
code=c:\orahome\olap\oesnzn\common\
data=d:\db\coffee\
extend=e:\db\coffee\
export=d:\export\
personal=d:\users\
```

For information about modifying the RDC file, see [“Remote Personal Databases” on page 4-4](#).

For a RAM database, configure Relational Access Manager

If your site uses Relational Access Manager to access data that is stored in an RDBMS, you must specify a default relational ID and password in your Express database. The password will be used during Analyzer start-up to access the relational database. For details, refer to topics related to defining maintenance procedures in the Relational Access Manager Help system.

Ongoing task: Import users from Oracle Applications

As you use Database Manager to create new users, you must import the user names from Oracle Applications. For details, see [“Importing User Names from Oracle Applications”](#) on page 7-11.

Setting Up the Analyzer Home Page

Using Analyzer Web Administrator

You use Analyzer Web Administrator to set up the Analyzer Home page. If you want groups of users to see unique home pages, you can define multiple home pages. For example, you might define home pages which allow users access to different databases.

Procedure: Setting up an Analyzer Home page

To set up the Analyzer Home page, perform the following steps:

1. Direct your Web browser to the virtual directory (`osadba`) that you defined for the Analyzer Web Administrator page when you assigned virtual directories in the Web listener.

Example: `http://myserver.mycompany.com/osadba/`

Result: The initial page for Analyzer Web Administrator appears.

2. If prompted, enter a valid user ID and password for Windows NT or Unix.
3. Scroll or jump to a setup task and make a selection or enter the required information. The tasks are as follows:
 - **Specify the Analyzer Home Page Path** — Indicates the URL for the Analyzer Home page.
 - **Enable Local Install or JInitiator** — Specifies how users will access Java classes.
 - **Identify the List of Available Databases** — Lists each database that will be available to Web users. Also specifies Analysis Library selections for the database.

- **Specify Text and Graphics for the Welcome Screen** — Provides optional custom text and graphics for the Welcome screen.
 - **Specify Graphics for the Analyzer Home Page** — Provides optional custom graphics for the Analyzer Home page.
4. Repeat Step 4 for each task that you want to perform.
 5. When your work is complete, do one of the following:
 - To save the settings, scroll to the bottom of the page and click **Submit**.
 - To save the settings *and* exit the application, scroll to the end of the page and click **Submit and Close Session**.

Specifying the Analyzer Home page path

About the Analyzer Home page path

The Analyzer Home page path indicates the URL for the Analyzer Home page. You must specify a fully qualified path on the NT or Unix server. The path must be associated with a virtual directory that is defined in your Web listener.

For example, if you defined a virtual directory for

`f:/orahome/olap/osa6xx/osaweb/` in your Web listener, then you could enter `f:/orahome/olap/osa6xx/osaweb/index.html` as the Analyzer Home page path.

Procedure: Specifying the Analyzer Home page path

To specify the Analyzer Home page path, perform the following steps:

1. On the Analyzer Web Administrator page, scroll or jump to the **Specify the Analyzer Home Page Path** section.
2. In the **Fully qualified path** box, enter the fully qualified path for the Analyzer Home page.
3. When your work is complete, do one of the following:
 - To save the settings, scroll to the bottom of the page and click **Submit**.
 - To save the settings *and* exit the application, scroll to the end of the page and click **Submit and Close Session**.

Specifying the installation type

About installation types

The installation type determines how users will access Analyzer's Java foundation classes when they start Analyzer on their PC. You can select an option whereby Java classes are stored remotely and are downloaded to the local PC and installed by the browser during initial login. You can select another option whereby Java classes are downloaded each time a user logs in. Alternatively, you can specify that users run the application under Oracle JInitiator. JInitiator is implemented as a plug-in (for Netscape Navigator) or as an Active X Object (for Microsoft Internet Explorer) and replaces the default JVM provided by the browser.

Note: If visually impaired individuals will be using a screen reader such as JAWS to access Analyzer in the Web environment, select the JInitiator (**Local cache install using Oracle JInitiator**) option.

Procedure: Specifying an installation type

To indicate how you want users' to access Java foundation classes for Analyzer, perform the following steps:

1. On the Analyzer Web Administrator page, scroll or jump to the **Enable Local Install or JInitiator** section.
2. Select one of the following options:
 - **Local install for Netscape Communicator only** — Netscape users will install Java foundation classes on the PC the first time that they log into Analyzer. Internet Explorer users will download Java foundation classes to the PC each time that they log into Analyzer.
 - **Local install for Internet Explorer only** — Internet Explorer users will install Java foundation classes on the PC the first time that they log into Analyzer. Netscape users will download Java foundation classes to the PC each time that they log into Analyzer.
 - **Local install for both Internet Explorer and Netscape Communicator** — Netscape and Internet Explorer users will install Java foundation classes on the PC the first time that they log into Analyzer.
 - **Remote install for both Internet Explorer and Netscape Communicator** — Netscape and Internet Explorer users will download Java foundation classes to the PC each time that they log into Analyzer.

- **Local cache install using Oracle JInitiator** — Users will run Analyzer under the installed version of Oracle JInitiator. The Mime Type, Class ID, Code Base, and Executable Path boxes display the properties for the currently installed version of JInitiator.
3. When your work is complete, do one of the following:
 - To save the settings, scroll to the bottom of the page and click **Submit**.
 - To save the settings *and* exit the application, scroll to the end of the page and click **Submit and Close Session**.

Identifying the list of available databases

About available databases

For each database that you want to make available to Web users, you must specify a fully qualified path on the NT or Unix server. You must also indicate if the database is relational. You can also set up Analysis library documents for the database.

If this is the first time that you are using Analyzer Web Administrator, no databases are specified.

Procedure: Specifying available databases

To specify available databases, perform the following steps:

1. On the Analyzer Web Administrator page, scroll or jump to the **Identify the List of Available Databases** section.
2. In the Description column, enter a description for a database. This is the description that will be displayed for the database on the Analyzer Home page.
3. In the Location column, enter the fully qualified path for the RDC file associated with the database.

For information about the RDC file, see [Appendix C](#).

4. If the database is relational, scroll to the right until the Relational column appears and check the box.
5. To set up Analysis Library documents for the database, scroll to the right until the Analysis Library column appears, and click **Add**.

Result: The Configure Web documents for database section is displayed.

6. Set up the Analysis Library as described in the topic [“Setting up the Analysis Library” on page 8-11](#).
7. Repeat Steps 2 through 6 for each database that you want to set up.
8. When your work is complete, do one of the following:
 - To save the settings, scroll to the bottom of the page and click **Submit**.
 - To save the settings *and* exit the application, scroll to the end of the page and click **Submit and Close Session**.

Example: Available database settings

The following illustration shows the Available Databases section of the Analyzer Web Administrator page. The example has entries for a single database, Saturn Electronics. You can see the columns for Description, Location, and Relational Database. The Analysis Library column is not visible.



Identify the List of Available Databases

Enter the description and the fully qualified path of the Remote Database Configuration (RDC) file for each database you want to make available to the user. The description that you supply will be used as the database name in the list of databases on the Analyzer home page. If a database is relational, click on the box in the Relational column. If you wish to generate the Analysis Library Reports, click on the Add button in the Analysis Library column.

Description	Location	Relational?
Saturn Electronics	e:/saturn/6392b2/tools/saturn1.rdc	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

Setting up the Analysis Library

About Analysis Library setup

The Analysis Library consists of documents which enable users to conduct pre-defined business analyses. As DBA, you select the documents that all users will see and specify the dimensions for which specific documents will be available.

Procedure: Specifying default settings for Analysis Library documents

To specify default settings for Analysis Library documents, perform the following steps:

1. On the Analyzer Web Administrator page, scroll or jump to **Identify the List of Available Databases**.

Result: The Available Databases section is displayed, showing the databases for which Web access has been enabled.

Important: If no databases are listed, you must set up the databases as described in the topic [“Identifying the list of available databases” on page 8-9](#).

2. Identify the database for which you want to set up the Analysis Library and scroll to the right until you see the Analysis Library column.

3. Click **Add**.

Result: The Configure Web documents for database page is displayed. If this is a relational database, you might be prompted to log into the relational database before you can access this area.

4. Click or scroll to **Select Documents to Generate**.

5. To select a document, check its Available box.

Note: To deselect a document, clear its Available box.

6. For each selected document, choose the dimensions for which the document will be available. You can make unique selections for each document. Scroll to the right to ensure that you can see all of the available dimensions.

Note: When documents are generated, there will be a document of each type for each dimension that you select. For example, if you select Geography and Product as dimensions for the Comparison document, the Comparison folder in the navigation list on the Analyzer Home page will include a Geography Comparison document and a Product Comparison document.

7. Scroll or jump to **Set System Defaults** and proceed as follows.
 - a. In the Folder View Preferences section, indicate whether you want the folders for Analysis Library documents to display before or after other public and personal document folders in the navigation list.
 - b. In the Time Dimension section, select a Time hierarchy for documents that will use a fixed time period. Also select levels for Year, Quarter, and Month. If your Time hierarchy is not structured precisely in terms of Year, Quarter, and Month, select three levels in descending order.
 - c. In the Measure Dimension section, select a base measure and a comparison measure. The base measure will appear as the default measure in documents that include a single measure. The comparison measure will appear in documents that display comparisons. For example, you might specify Sales as the base measure and Quota as the comparison measure.
 - d. Two documents, Account Review and Quota Review, are only available for a single dimension. If you intend to make these documents available to your users, in the Reviews Section, select the dimension for reviews.
8. To save your settings, scroll to the bottom of the page and click **Configure**.

Overview of Analysis Library documents

The Analysis Library includes 19 documents. From among these, you select the documents that you want to provide to your user community. Following is a list of document types:

- **Comparison documents** — Comparison documents compare products, geographies, and so forth, to highlight patterns, exceptions, and comparative performance. Comparison documents include Comparison, Comparison to Aggregate, Year-to-Year Comparison, and Mix Comparison. The Comparison documents that you select will be displayed in the Comparison folder in the navigation list.
- **Distribution documents** — Distribution documents demonstrate how distribution affects performance. Distribution documents include Increasing/Decreasing Sales and New Sales. The Distribution documents that you select will be displayed in the Distribution folder in the navigation list.
- **Exception documents** — Exception documents highlight dimension values such as products or geographies in which activity differs substantially from the average. Exception documents include 80/20 Rule, Sales Above/Below Last Year, Sales Above/Below Threshold. The Exception documents that you select will be displayed in the Exception folder in the navigation list.

- **Growth documents** — There is a single Growth document which shows the difference in a dimension value's performance between two time periods. If you select the Growth document, it will be displayed in the Growth folder in the navigation list.
- **Quota documents** — Quota documents compare the actual performance of dimension values such as products and geographies to a target. Quota documents include Above/Below Quota, and Quota Performance. The Quota documents that you select will be displayed in the Quota folder in the navigation list.
- **Ranking documents** — Ranking documents rank dimension values such as products and geographies in terms of a specific measure. Ranking documents include Ranking and Top/Bottom. The Ranking documents that you select will be displayed in the Ranking folder in the navigation list.
- **Review documents** — Review documents provide a graphical snapshot of a specific account at a specific time to identify business trends. Review documents include Account Review and Quota Review. The Review documents that you select will be displayed in the Review folder in the navigation list.
- **Trend documents** — Trend documents present trends over time for dimension values such as products or geographies. Trend documents include Cumulative Sales Trend, Moving Average/Total, and Trend. The Trend documents that you select will be displayed in the Trend folder in the navigation list.

Note: Dimension selection restrictions for documents

In the majority of cases, you can specify that a particular document be made available for one or more of the dimensions in your database. However there are some restrictions:

- For the Mix Comparison document, the analysis dimension is always a Product dimension or a dimension that represents Product (for example, Item). The database must also have a Geography dimension or a dimension that represents Geography (for example, Customer).
- For the Year-to-Year Comparison document and the Cumulative Sales Trend document, the analysis dimension is always a Time dimension. For all other documents, the analysis dimension cannot be a Time dimension.
- In order for a dimension other than Time to be available, the dimension must have a hierarchy. If a dimension does not have a hierarchy, it will display as unavailable (N/A) for all documents.

Example: Analysis Library document selections

In the following illustration, the user selects two Comparison documents (Comparison and Comparison to Aggregate) for the Product dimension and Exception documents (80/20 Rule and Sales Above/Below Last Year) for the Product and Geography dimensions.

Document Description	Available?	Time	Geography	Product	Channel
Comparison Folder					
Comparison	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Comparison to Aggregate	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Year to Year Comparison	<input type="checkbox"/>	<input type="checkbox"/>	N/A	N/A	N/A
Mix Comparison	<input type="checkbox"/>	N/A	N/A	<input type="checkbox"/>	N/A
Distribution Folder					
Increasing/Decreasing Sales	<input type="checkbox"/>	N/A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
New Sales	<input type="checkbox"/>	N/A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Exception Folder					
80/20 Rule	<input checked="" type="checkbox"/>	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sales Above/Below Last Year	<input checked="" type="checkbox"/>	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Specifying text and graphics for the Welcome screen

About Welcome screen text and graphics

The Welcome screen is displayed when a user logs into the Analyzer Home page. The Welcome screen includes default text and a default image. You can replace the default text with custom text. You can also replace the default image with a custom graphic. The custom graphic must be a GIF or JPEG file. The file path must be defined as a virtual path in your Web listener.

Procedure: Specifying custom text and graphics for the Welcome screen

To specify unique text and graphics for the Welcome screen, perform the following steps:

1. On the Analyzer Web Administrator page, scroll or jump to the **Specify Text and Graphics for the Welcome Screen** section.
2. To specify custom text, type the text in the **Enter welcome text** box.
3. To specify a custom graphic, in the **Image virtual path** box enter the virtual path for the JPEG or GIF file that you want to use.
Important: The virtual path must be defined in your Web listener.
4. To save the settings, scroll to the bottom of the page and click **Submit**.

Specifying graphics for the Analyzer Home page

About Home page graphics

By default, the Analyzer Home page displays an “Oracle Sales Analyzer” banner graphic and no background graphic. You can specify a graphic for a custom banner, a custom background, or for both a custom banner and a custom background.

Requirements for custom graphics

The file path for a custom graphic must be defined as a virtual path in your Web listener. The banner graphic must be a GIF or JPEG file that is 1600 pixels wide by 90 pixels high.

Procedure: Specifying custom graphics for the Home page

To specify unique banner and background graphics, perform the following steps:

1. On the Analyzer Web Administrator page, scroll or jump to the **Specify Graphics for the Analyzer Home Page** section.
2. To specify a file for the background graphic, in the **Background graphic virtual path** box enter the virtual path for the JPEG or GIF file that you want to use.
Important: The virtual path must be defined in your Web listener.

3. To specify a file for the banner graphic, in the **Banner graphic virtual path** box enter the virtual path for the JPEG or GIF file that you want to use.

Important: The virtual path must be defined in your Web listener. The dimensions of the graphic must be 1600 x 90 pixels.

4. To save the settings, scroll to the bottom of the page and click **Submit**.

Customizing the Web Application

Extending Analyzer Web functions

Analyzer includes several hooks that enable you to extend the basic functionality of the Web application. For example you might use the Analyzer Web API to open a specific Sales Analyzer document via a URL which can be embedded in an e-mail or another web application. You can also enable drilling to relational detail within an Analyzer Web document. For detailed information about implementing these types of custom functions, refer to the Technical Note, *Hooks in the OSA Web Interface*.

Implementing Analyzer Web access in the Analytic Solutions portal environment

You can run Analyzer within the Analytic Solutions portal environment. In order to do this, you must install AS Portlets as a server application. You must also install supporting software. For information about installation and setup tasks related to the portal environment, see the *Oracle Sales Analyzer Installation Guide*.

Rebranding Analyzer

If your site will be integrating Analyzer into another application, you might want to change the generic text and graphics that are associated with the introductory screens to reflect the environment of the host application. Also, if you are implementing a multi-language installation, you might want to display generic text in a specific language.

You can use the Analyzer Web configuration file to implement these customizations. In a standard Analyzer installation, the configuration file is `osaweb.cfg`, located in the `/osaweb/` directory on the server where the Web server is installed. Following installation, the configuration file has the following defaults:

```
[BANNER]
TITLE=(default)
IMAGE=(default)

[SPLASH]
IMAGE=(default)
INIT_MSG=(default)
WELCOME_MSG=(default)
```

Specifying custom text

To specify custom text, replace the default value for a text property with your text. You can provide custom text for the following:

- `TITLE` property in the `BANNER` section — Specifies the text that will be displayed when users select a database and start Analyzer.
- `INIT_MSG` property in the `SPLASH` section — Specifies the text that will be briefly displayed while Analyzer is initializing.
- `WELCOME_MSG` property in the `SPLASH` section — Specifies the text that will be initially displayed in the right frame on the Analyzer Home page.

Specifying custom graphics

To specify a custom graphic, replace the default value for an `IMAGE` property with the name and virtual path of your GIF or JPEG file. The file path for the graphic must be defined as a virtual path in your Web listener.

You can provide custom graphics for the following:

- `IMAGE` property in the `BANNER` section — Specifies the graphic that will be displayed in the banner area when users select a database and start Analyzer. The dimensions of the graphic must be 1600 pixels wide by 90 pixels high.
- `IMAGE` property in the `SPLASH` section — Specifies the graphic that will be initially displayed in the right frame on the Analyzer Home page.

Note: If you used the Analyzer Web Administrator page to change custom graphic defaults for the Home page or Welcome screen, as described in [“Setting Up the Analyzer Home Page” on page 8-6](#), the settings on the Administrator page take precedence over those in the configuration file.

Providing tokens for languages other than English

To provide custom tokens for languages other than English, add a line for the language to the appropriate sections of the configuration file. Enter the property that you want to specify and append the appropriate language code. Then enter the value for the property.

For example, if you want to specify a welcome message and image to appear in French, the [SPLASH] section of the configuration file might resemble the following:

```
[SPLASH]
IMAGE_FR=/osaweb/limage.gif
INIT_MSG=(default)
WELCOME_MSG_FR=bonjour
```

Note: If the user's browser language setting does not match a token, then English will be used by default.

Providing Web Access to Users

Distributing the Analyzer URL

Distribute the URL for the Analyzer Home page to your user community. This is the /osaweb/ directory that you specified as the virtual directory for the Analyzer Home page in your Web Listener.

For example, the URL might resemble the following:

```
http://myserver.mycompany.com/osaweb
```

Informing users about browser settings

You should instruct users how to set up their browsers with the following defaults:

- Accept cookies
- Compare document to cache every time

Users who will access Analyzer in a language other than English should have their browsers set to that language.

Providing voice access to Web functions

Users who have installed a screen reader, such as JAWS, on their PC can have voice access to Analyzer in the Web environment. Before you can set up voice access, you must perform the following procedures in Analyzer:

- In the OSAWEB.CFG file, set the value for the `ENABLE` parameter for JAWS to `YES` (the default is `NO`). For information about the OSAWEB.CFG, file, see [Appendix D](#).
- In the Analyzer Web Administrator page, choose **Jinitiator** as the Installation Type. Jinitiator is the only installation type that supports verbal communication. For more information, see [“Specifying the installation type” on page 8-8](#).

You must also configure the screen reader to work with the application.

Alert Netscape users about local install prompts

If you have enabled local install, then users with Netscape browsers might need guidance on how to proceed the first time that they enter the URL for the Analyzer Home page.

What happens	Action
The Oracle Sales Analyzer Local Install page appears.	Instruct users to choose Click Here to Continue the Download and then restart the browser. Note: This page will not appear again unless Netscape is reinstalled or Analyzer is reinstalled.
A Java security page appears.	Instruct users to check Remember this decision and then choose Grant .

Scoping a Database

Chapter summary

This chapter describes how to scope a database in order to restrict user access to data.

List of topics

This chapter includes the following topics:

- [Overview of Scoping](#)
- [Defining Data Cubes](#)
- [Copying Data Cubes](#)
- [Defining Scopes](#)
- [Enabling Scoping](#)
- [Associating Scopes with User Groups](#)
- [Refreshing Scopes for a Shared Shell](#)
- [Modifying Scopes](#)
- [Removing Scopes](#)

Overview of Scoping

Purpose of scoping

You scope a remote database to restrict the access of a group of users to specified dimension values. For example, you can limit access to a specific geographic area such as the Eastern or Western region.

Process for scoping a database

The following list describes the general process that you follow to scope a database:

1. You define data cubes that specify the limits of the base data that specific groups of users will access. You can use Analyzer or Database Manager to define data cubes.
2. If you used the Analyzer client to define data cubes, you must then publish the data cubes to the Master public library.

If you used Database Manager to define data cubes, you do not have to publish the cubes. They are automatically saved in the Master library.

3. You define scopes that reference the data cubes. You use Database Manager to define scopes.

Note: As an alternative to defining a data cube and referencing the cube in a scope, you can use a single ad hoc process in Database Manager to select and save the values for scope. This process automatically creates both the scope and the data cube in the Master library.

4. You enable scoping for the database.
5. You associate scopes with user groups.

Custom measures and custom aggregates will be ignored in scopes

If you want scoped users to have access to custom aggregates and custom measures, you must copy these objects to appropriate public libraries. Only custom aggregates and custom measures that are stored in libraries to which a user has access will be visible to that user. If you attempt to include a custom aggregate or custom measure in a scope's data cube, a warning message will appear.

Defining Data Cubes

About data cubes

A *data cube* is a selection of data that can be limited across any of the base dimensions, including the Measure dimension.

A group of users can access only one scope. When you define a data cube that will be the basis of a scope, ensure that it contains all of the data that the group will need.

Procedure: Using Analyzer to define a data cube

To use the Analyzer client to define a data cube, perform the following steps:

1. From the Analyzer Tools menu, choose **Maintain Data Cubes**.

Result: The Maintain Data Cubes dialog box is displayed.

2. Choose **Create**.

Result: The Create Data Cube dialog box appears.

3. Type a name for the data cube and choose **OK**.

Result: You are returned to the Maintain Data Cubes dialog box. The Name box displays the name of the cube. The Contents box displays the number of selections for each dimension.

4. Select dimension values for the data cube as follows:

- a. In the Contents box, select a dimension and click the **Selector** button.

Result: The Selector dialog box appears. The Dimension box displays the dimension you chose.

- b. Choose values for the current dimension. You can select a hierarchy within which to limit the values. You can select individual values from a list. You can also use a variety of tools to select and sort values.

Important: Do not use data-based selections (i.e., selections made using Top/Bottom or Exception criteria in the Selector) when you define the data cube. Also, take care when using the Attribute tool. The *Attribute Values* list box will be populated with all attribute values that are contained in the database, even if the values belong to dimension values that fall outside the scope.

- c. To select values for another dimension, select the dimension in the Dimension box and repeat Step b.
 - d. Choose **OK**.
Result: You are returned to the Maintain Data Cubes dialog box. The Contents box now reflects the choices that you made.
5. Choose **Save** to save the data cube.

Procedure: Using Database Manager to define a data cube

To use Database Manager to define a data cube, perform the following steps:

1. From the Database Manager Catalogs menu, choose **Maintain Data Cubes**.

Result: The Maintain Data Cube dialog box is displayed.

2. Choose **Create**.

Result: The Create Data Cube dialog box appears.

3. Type a name for the data cube and choose **OK**.

Result: You are returned to the Maintain Data Cube dialog box. The Name box displays the name of the cube. The Contents box displays the number of selections for each dimension.

4. Select dimension values for the data cube as follows:

- a. In the Contents box, select a dimension and click the **Selector** button.

Result: The Selector dialog box appears. The Dimension box displays the dimension you chose.

- b. Choose the values for the current dimensions. You can select a hierarchy within which to limit the values. You can select individual values from a list. You can also use a variety of tools to select and sort values.

Important: Do not use data-based selections (i.e., selections made using Top/Bottom or Exception criteria in the Selector) when you define the data cube. Also, take care when using the Attribute tool. The *Attribute Values* list box will be populated with all attribute values that are contained in the database, even if the values belong to dimension values that fall outside the scope.

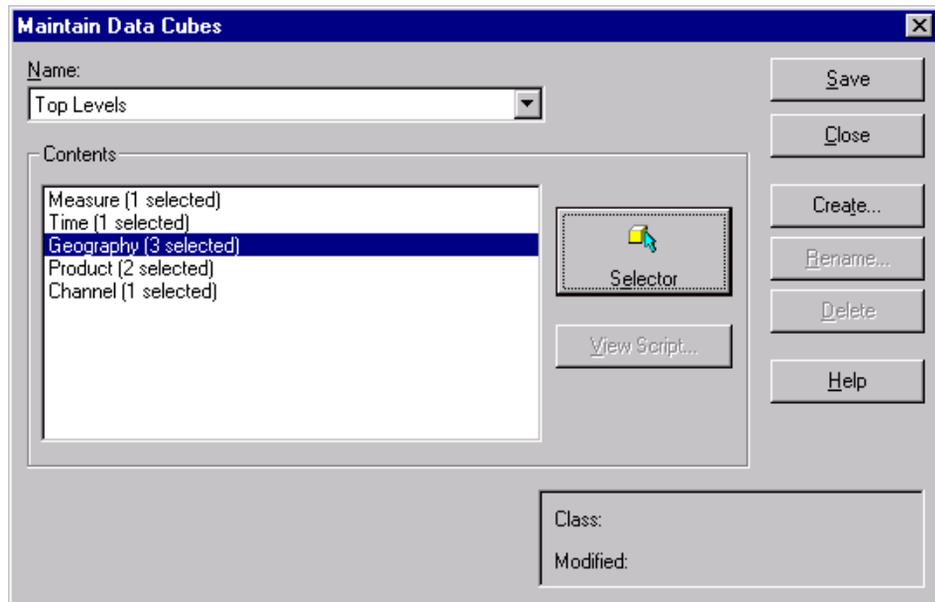
- c. To select values for another dimension, select the dimension in the Dimension box and repeat Step b.
- d. Choose **OK** to confirm your selections.

Result: You are returned to the Maintain Data Cube dialog box. The Contents box now reflects the choices that you made.

5. Choose **Save** to save the data cube.

Example: Data cube definition

The following example shows the Maintain Data Cubes dialog box in Analyzer. In this example, a data cube named “Top levels” has been limited across five dimensions.



What happens when you save a data cube?

If you used the Analyzer client to define your data cube, the data cube definition is saved in the Data Cube catalog in your personal database. If you used Database Manager to define and save your data cube, the data cube definition is saved in the Data Cube catalog in the public database in the Master library.

Related information

For more information about data cubes, refer to the Analyzer Help system, or search for the following topic in the Database Manager [Database Maintenance] Help system:

“Maintain Data Cube Dialog Box”

Copying Data Cubes

About copying a data cube

If you used the Analyzer client to create a data cube, you must copy it to the Master library to make it available for scoping.

You can use the Analyzer client to copy the data cube. Alternatively, you can use the Web interface.

Procedure: Using the Analyzer client to copy a data cube

To use the Analyzer client interface to copy a data cube to the Master library, perform the following steps:

1. From the Analyzer Tools menu, choose **Maintain Libraries**.
Result: The Maintain Libraries dialog box is displayed.
2. In the **Catalog** box, choose **Data Cube Catalog**.
3. In one Platform box, select **PC**. In the Library box, select **Private**.
Result: The Entries box displays the data cubes in your personal database.
4. In the Entries box, select a data cube.
5. In the other Platform box, select **MF**. In the Library box, select **Master**.
6. Choose the **Copy** button that points to the Master library.
7. Choose **Close** to exit.

Related information

For more information about copying to and from libraries, search for the following topics in the Analyzer Help system:

- “Copying Objects from Libraries”
- “Copying Objects to Libraries”

Defining Scopes

About defining a scope

Defining a scope involves populating the Scope catalog (SP.CATALOG) with the name of the scope and its referenced data cube. You use Database Manager to define, modify, and delete scopes.

Special requirement for scoping

If the MCALCINT property in the Source catalog (SOCAT) is set to YES, you must perform a preliminary procedure in Express before you define the first scope.

Procedure: Special preparation for scoping

To prepare for scoping if the MCALCINT property in SOCAT is set to YES, perform the following steps before defining the first scope:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. If you have any permanent custom aggregates or custom measures in your server database, perform the following steps once before defining the first scope:
 - a. Determine the names of custom aggregates in your database that are permanent so that you can have a record of them.

```
->report down pb.ag.entry -  
Continue>pb.ag.catalog -  
Continue>(pb.ag.prop 'PERMANENT')
```


Methods for defining a scope

You can base a scope on a data cube that has been saved in a public library. Alternatively, you can use a single ad hoc process in Database Manager to select and save the values for a scope. This process automatically creates both the scope and the data cube in the Master public library.

Procedure: Defining a scope based on a data cube

To define a scope based on a data cube, perform the following steps:

1. From the Database Manager Catalogs menu, choose **Maintain Scopes**.

Result: The Maintain Scopes dialog box is displayed.

2. Choose **Create**.

Result: The Create New Scope dialog box is displayed.

3. Type a name for the scope and choose **OK**.

You are returned to the Maintain Scopes dialog box. The Names box displays the name that you entered.

4. In the Data From box, select **Data Cube**.

Note: The Data Cube option is only enabled if there is at least one data cube in the Master library.

5. In the Data Cube box, select a data cube from the list.

Note: The number of dimension values in the selections that make up the data cube are displayed in the Selections box, although the box is not enabled.

Choose **Save** to associate the data cube with the scope.

6. Choose **Close** to exit the Maintain Scopes dialog box.

Defining a scope based on an ad hoc selection of values

To define a scope based on an ad hoc selection of values, perform the following steps:

1. From the Database Manager Catalogs menu, choose **Maintain Scopes**.

Result: The Maintain Scopes dialog box is displayed.

2. Choose **Create**.

Result: The Create Scope dialog box is displayed.

3. Type a name for the scope and choose **OK**.

You are returned to the Maintain Scopes dialog box. The Names box displays the name that you entered.

4. In the Include Data From box, choose **Selections**.

5. Select dimension values for the scope as follows:

- a. In the Selections box, choose a dimension and click the **Selector** button.

Result: The Selector dialog box appears. The Dimension box displays the dimension you chose.

- b. Choose the values that will comprise the scope for the dimension that appears in the Name box. You can use a hierarchy to limit your value selections. You can select individual values from a list. You can also use a variety of tools to select and sort values.

Important: Do not use data-based selections (i.e., selections made using Top/Bottom or Exception criteria in the Selector) when you select values. Also, take care when using the Attribute tool. The *Attribute Values* list box will be populated with all attribute values that are contained in the database, even if the values belong to dimension values that fall outside the scope.

- c. To select values for another dimension, select the dimension in the Dimension box and repeat Step b.

- d. Choose **OK** to confirm your selections.

Result: You are returned to the Maintain Scopes dialog box. The number next to the dimension names in the Selections box now reflects the choices that you made.

6. If you used a tool to select dimension values or if you chose a saved selection of values, you can choose **View Script** to see the selection script.

7. Choose **Save** or **Apply** to save the scope definition.

Result: You are prompted to save the data cube definition.

8. Choose **OK** to continue.

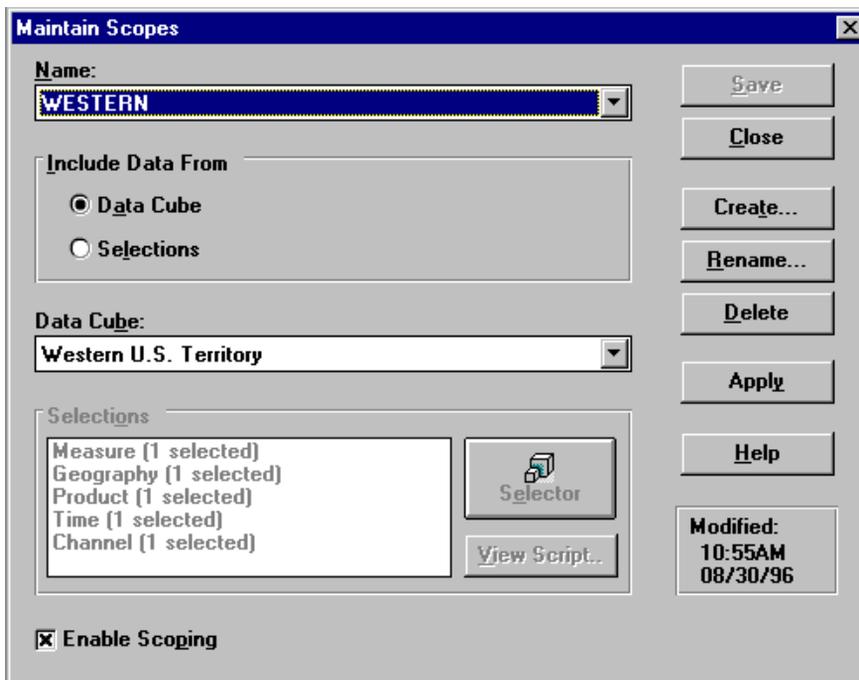
9. In the Create Data Cube box, type a name for the data cube and choose **OK**.

You are returned to the Maintain Scopes dialog box. The selection in the Include Data From box is now **Data Cube**. The name in the Data Cube box is the name that you entered.

10. Do one of the following:
 - Choose **Save** to save the data cube.
 - Choose **Apply** to save the data cube and close the Maintain Scopes dialog box.
11. If you chose **Save** in the previous step, choose **Close** when you are ready to exit the Maintain Scopes dialog box.

Example: Defining a scope based on a data cube

The following example shows the Maintain Scopes dialog box. The Name box contains the name of the scope. The Data Cube box contains the name of the data cube referenced by the scope. Enable Scoping is selected.



Related information

For more information about defining a scope, search for the following topics in the Database Manager [Database Maintenance] Help system:

- “Defining Scopes”
- “Enabling Scoping”
- “Modifying Scopes”
- “Setting Scopes for Users”

Enabling Scoping

About enabling scoping

By default, scoping is not enabled for a new Analyzer installation. Until you have enabled scoping, scopes that you have defined are inactive.

You enable scoping for the database by setting a master switch in Database Manager.

Procedure: Enabling scoping

To enable scoping, perform the following steps:

1. From the Database Manager Catalogs menu, choose **Maintain Scopes**.
Result: The Maintain Scopes dialog box is displayed.
2. Select the **Enable Scoping** check box.
Result: Scoping is enabled for the database.

Associating Scopes with User Groups

Relationship between scopes and groups

Associating a scope with a group of users limits the users' view of the data to the values on which the scope is based. Oracle Corporation recommends that you set up scopes to align with public libraries. This will ensure that users have access to only those custom aggregates and custom measures for which they have the requisite base data.

In Analyzer Version 6.31 and all prior versions, a user was granted access to custom aggregates and custom measures based on whether or not these objects had been added to the user's scope. As of Analyzer Version 6.32, access to custom aggregates and custom measures is controlled through library access, rather than via scoping. A user is granted access to a published custom aggregate or custom measure based on whether or not he or she has read-only or read/write access to the library to which the object has been published. For more information, see [“Note: Publishing for scoped users” on page 7-16](#).

Associating scopes with user groups

Use the Maintain Groups function in Database Manager to associate scopes with groups. For more information, see [“Working with Groups” on page 7-2](#).

Refreshing Scopes for a Shared Shell

Using multiple scopes with a shared shell

You might maintain a read-only shared shell database on the LAN, and use more than one scope. If so, you must refresh a different shell for each scope.

Procedure: Refreshing a shared shell

To refresh multiple scopes if users access a shared shell, perform the following steps:

1. Open the database with read/write access mode to the shell once for each scope. Use a user name from each group in order to get the properly scoped DB files.

Important: Ensure that each time you open the database, you use a different personal database for each user name; otherwise, errors will occur.

2. Copy the shells to the LAN and flag them as read-only and shareable.

Important: Ensure that you copy the files to different subdirectories or that the files have unique names. Because the scopes are all from the same “root” database, if they all have the same name and you copy them to the same place on the LAN, one will overwrite another.

Related information

For more information about refreshing a shell database, see [“Maintaining a Shell Database” on page 5-12](#).

Modifying Scopes

Methods for modifying a scope

You can change a defined scope as follows:

- You can change the value of the data cube on which the scope is based.
- You can associate the scope with a different data cube.

Related information

For more information about modifying a scope, search for the following topics in the Database Manager [Database Maintenance] Help system:

“Maintain Scopes Dialog Box”
“Modifying Scopes”

Removing Scopes

Guidelines for removing a scope

When you no longer need a scope, you can remove it from the database.

Important: If you are removing the only scope for this database, you must also disable scoping.

Procedure: Deleting a scope

To delete a scope, perform the following steps:

1. From the Database Manager Catalogs menu, choose **Maintain Scopes**.
Result: The Maintain Scopes dialog box is displayed.
2. In the Name box, select the scope and choose **Delete**.
3. If you are deleting the only scope for this database, clear the **Enable Scoping** check box to disable scoping for the database.

Chapter summary

This chapter describes methods and procedures for building slice databases, in order to provide portability of or faster access to specified data.

List of topics

This chapter includes the following topics:

- [General Instructions for Building a Slice](#)
- [Defining a Slice](#)
- [Creating a Slice Directly](#)
- [Creating a Slice from Files](#)
- [General Instructions for Updating a Slice](#)
- [Updating a Slice Directly](#)
- [Updating a Slice from Files](#)
- [Upgrading a Slice](#)

General Instructions for Building a Slice

Definition: Slice

A slice is a subset from a remote database that can contain selected dimension values, document definitions, and data cubes. It is often used to provide portability or faster access to a frequently used portion of data. A slice can reside on a PC or it can be shared across a LAN server.

Process for building a slice

The following procedure describes the process that you use to build a slice:

1. Define the slice by specifying its data and how that data will be dimensioned.
2. Create the slice, using the slice definition that you have defined in Step 1.

Note: You cannot create a slice from another slice.

Tools for building a slice

You can use Analyzer or Database Manager as a tool to define the data cubes that make up a slice, as well as to define the slice itself. Then, to *create* the slice, you can use either Application Manager or Express commands and Application Manager, depending on the method you choose.

Methods for creating a slice

The following table describes the two methods that can be used to create a slice.

Method	Description	Advantages
Creating a slice directly	Using Application Manager, you can pipe the data directly from the source database into a slice database. Note: To create a slice directly from a remote database, you must access the database in thick client mode.	You can easily create and maintain a slice on a LAN server for users to share. Or, users can create and maintain their own slices. This method has a Windows interface and does not require access to the command line of the operating system.
Creating a slice from file	Using Express commands, you can distribute slice EIF files to your users via diskettes or a LAN server. Then, you or your users can use Application Manager to create a slice from the files.	You can create a large slice in batch mode. Users do not need remote communications and are not involved in downloading the slice.

Defining a Slice

Overview of defining a slice

Defining a slice can involve the following tasks:

- Optionally, estimating the size of the slice
- Identifying data and definitions for the slice
- Defining the slice using Analyzer or Database Manager
- Optionally, adding saved selections to the slice

Estimating the size of the slice

The size of a slice depends on the following:

- The number of dimension values included in the slice
- The type of data represented in the slice

The following formula can be used to estimate the *maximum* size of a slice.

maximum slice size = no. of measures x no. of dim
 values that the measures are
 dimensioned by x bytes per measure
 x density factor

Factor	Description
No. of dim. values	The product of the number of dimension values selected for each dimension.
Bytes per measure	The weighted average of the number of bytes required to store data for measures. For most data, use four bytes per measure.
Density factor	A value for the estimated density of your data. Use a value between 1 percent and 100 percent. The density factor varies greatly from database to database.

For example, assume that you have defined a slice with 10 measures (all dimensioned by Geography, Time, and Product), 100 geography values, 20 time periods, and 25 products, and that your data requires four bytes per measure. The *maximum* size of your slice would be 2,000,000 bytes (10 X 100 X 20 X 25 X 4 X 100). This example assumes a data density factor of 100 percent. Depending on the density of your data, a slice might be considerably smaller than the example.

Distributing large slices

In general, Oracle Corporation does not recommend that slices greater than 500MB be used with Analyzer. If slices are greater than 500MB, the performance of Analyzer is likely to degrade significantly — more so if the database grows larger than 750MB.

The maximum size of a slice that you can create and download to the PC or LAN is also limited by the resources of your remote system, as well as by the disk space that is available to store the slice on the PC or LAN. If you wish to create a slice that is larger than you can reliably create or download on your system, you can create and download several smaller slices with a limited number of time periods, and then combine them into a single slice on the PC or LAN using Application Manager's Update Slice command.

Identifying data and definitions for the slice

A slice contains data and the dimensions needed to index that data. You identify the data that will be copied into the slice by selecting measures and dimension values for those measures. If you want to include a custom aggregate in the slice, you must include all of its members.

Selected measures and dimensions must make up one or more *data cubes*, which are described in [Chapter 9](#). You may have to use more than one data cube in the definition of a slice if you want to include more than one hierarchy, or if your slice is too large for one EIF file. If you use multiple data cubes, you must be aware of the following:

- Parent-child relationships are not preserved between dimension values included in different data cubes. As a result, selections made in the Selector with the Family tool might not include all family members. Similarly, drilling on a dimension value may not expand or collapse the hierarchy to display or hide the children.
- If you select different dimension values from data cubes that do not intersect, zeros rather than NAs will appear on reports and graphs. Using the Global Computing database as an example, if one data cube contains the geographic region “Asia Pacific” and the product family “CD-ROM” and another cube contains the geographic region “Europe” but does not contain the product family “CD-ROM,” when the user selects the geographic regions “Asia Pacific” and “Europe” and the product “CD-ROM,” the value for the measure will be 0 for the combination of “Europe” and “CD-ROM.”

Note: Use of non-contiguous time periods

Oracle Corporation does not recommend the use of two or more non-contiguous time periods in a dimension representing periods of time. For example, do not create a slice in which the Time dimension consists of the first month and the last month of the year without including the intervening months as well. The use of non-contiguous time periods may cause errors when Express applications attempt to access data.

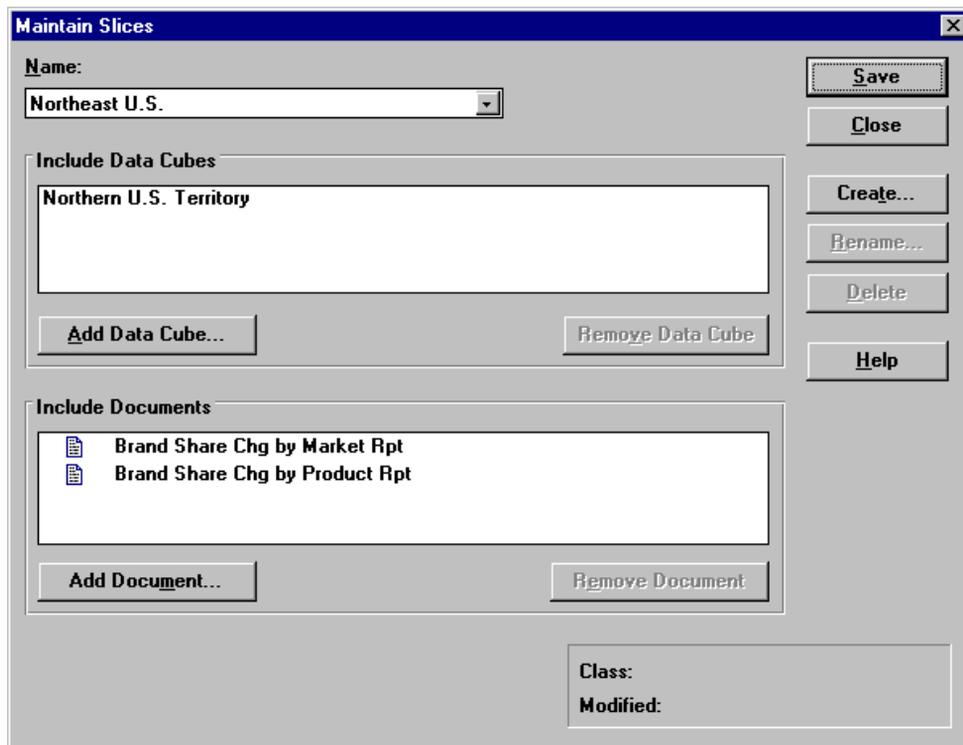
Procedure: Defining a slice

To define a slice, perform the following steps:

1. To create a new data cube for the slice,
 - a. From the Analyzer Tools menu, choose **Maintain Data Cubes**.
Result: The Maintain Data Cubes dialog box is displayed.
 - b. Complete the dialog box as described in [Chapter 9](#), save the data cube, and close the box.
2. From the Analyzer Tools menu, choose **Maintain Slices**.
Result: The Maintain Slices dialog box is displayed.
3. Complete the dialog box by selecting one or more data cubes and documents (reports and graphs) to include in the slice.
4. Save the slice definition and close the Maintain Slices dialog box.
5. If you are preparing to create slices in batch mode, or if users will share the slice definition, choose **Maintain Libraries** to copy the modified Data Cube and Slice catalogs from your private library to a public library, as described in [Chapter 7](#).
Result: The slice definition will be available for users to create a slice directly in Application Manager.

Example: Maintain Slices dialog box

The following is an example of the Maintain Slices dialog box with a sample slice, Northeast U.S. The slice includes a single data cube and multiple documents.



Adding saved selections to a slice

If you wish to include specific saved selections in the slice, open the database from Database Manager. From the Express Command Window, use Express to edit the CAT.ENTRIES property of SC.CATALOG on your PC. Identify each catalog entry with the catalog name, a colon, and the entry dimension value, for example, SL.CATALOG:DBA25. List each catalog entry on a separate line. Refer to [Chapter 2](#) for more information about using Express.

Related information

For more information about defining a slice, search for the following topics in the Analyzer Help system or the Database Manager [Database Maintenance] Help system:

“Maintain Data Cubes Dialog Box”

“Maintain Libraries Dialog Box”

“Maintain Slices Dialog Box”

Creating a Slice Directly

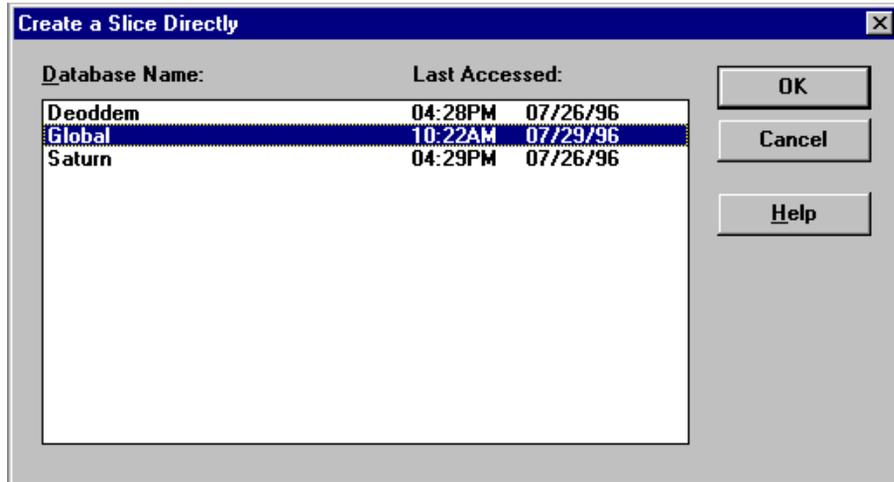
About creating a slice directly

After you have defined a slice, users with thick client access can create the slice using the **Create a Slice Directly** option on the Application Manager Tools menu. This option downloads data directly from the source database into a slice in the personal area. This method does not require you to perform any additional procedures once you have defined slices and made the definitions accessible to users. (To make a slice definition accessible to a user, the definition must be copied to a public library to which the user has access.)

Note: You cannot create a slice directly from a remote database that you are accessing in thin client mode. To create a slice directly from a remote database, use a database setup that specifies thick client mode.

Example: Create a Slice Directly dialog box

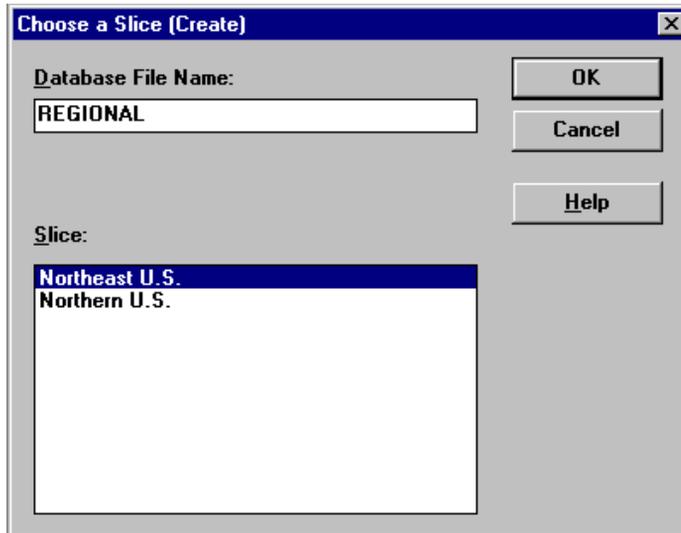
The following is an example of the Create a Slice Directly dialog box, with the Global database selected as the source for the slice.



After the user has selected a source database, the Choose a Slice (Create) dialog box is displayed, listing the available slice definitions. The user can then select the appropriate slice definition and type a database name for the slice.

Example: Choose a Slice (Create) dialog box

The following is an example of the Choose a Slice (Create) dialog box with the slice definition Northeast U.S. and the database name REGIONAL.



Related information

For more information about creating a slice directly, search for the following topic in the Application Manager Help system:

“Creating Slices Directly from Source Databases”

Creating a Slice from Files

Overview of creating a slice from files

Using Express commands, you can create EIF files on the remote system or LAN, and then distribute them to users. Users can then build a slice from the EIF files in Application Manager, using the **Create a Slice from File** option on the Tools menu.

Overview of the SC.CREATE.DRV utility

You create slice files by running the SC.CREATE.DRV utility, as described separately in this topic. The following table lists the slice files that are created or overwritten in the default directory when you run SC.CREATE.DRV.

Description	File Name
Table of Contents	TOCTXT
Binary Catalog	CATEIF
Binary Data	DAT01EIF, DAT02EIF, DAT03EIF . . .

Before running SC.CREATE.DRV, ensure that you have read/write access to the current directory where your database resides. Also, ensure that you have adequate disk space to create the slice files.

The syntax for SC.CREATE.DRV is as follows.

```
CALL SC.CREATE.DRV ('SLICE-NAME')
```

Here, "SLICE-NAME" is the name of a slice entered in the PB.SC.ENTRY dimension. The name of the slice (PB.SC.ENTRY) must be in uppercase.

Procedure: Creating slice files using the SC.CREATE.DRV utility

To create slice files, perform the following steps:

1. From Database Manager, open the remote database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. Run the SC.CREATE.DRV utility, using the name of the slice as it appears in PB.SC.ENTRY (e.g., DBA25).

Example:

```
->call sc.create.driv('DBA25')
```

4. Choose **Close** to close the Express Command Window and return to Database Manager.

Distributing slice files

Slice files must reside in the Analyzer installation directory. If you created slice files on a remote system, distribute the files by copying them to the local drive.

Before distributing slice files, you must rename each file to the database name, followed by the appropriate file name extension. For example, you would rename slice files for a database named SATURN to SATURN.TOC, SATURN.CAT, and SATURN.D01, etc.

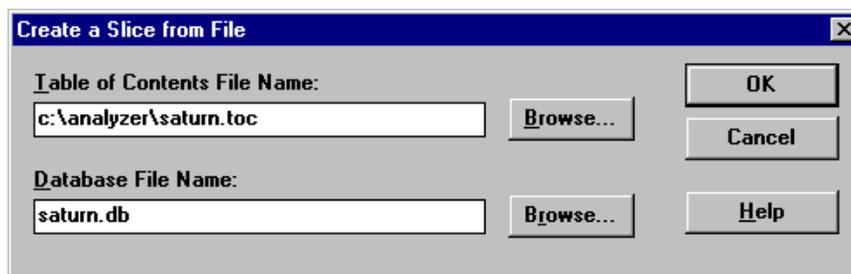
Before distributing slice files, rename them as shown in the following table.

Slice File	Renamed Slice File
TOCTXT	DATABASE . TOC
CATEIF	DATABASE . CAT
DAT01EIF, DAT02EIF, DAT03EIF . . .	DATABASE . D01, DATABASE . D02, DATABASE . D03 . . .

Converting slice files to a slice

Once you have distributed the files, users can create a slice by choosing **Create a Slice from File** from the Application Manager Tools menu.

The following is an example of the Create a Slice from File dialog box, with the table of contents and database file names selected.



Related information

For more information about creating a slice from files, search for the following topic in the Application Manager Help system:

“Creating Slices from Files”

General Instructions for Updating a Slice

Overview of updating a slice

You can update a slice to include data for additional time periods. For example, a slice might be current up to the week ending 4/5/98; if the last data update to your source database added the weeks ending 4/12/98 and 4/19/98, you can create an update slice that contains data for only these two additional time periods.

Note: You cannot simply update a slice if non-time dimensions such as markets or products have changed. Such changes are interspersed throughout the dimensions and create problems in aligning the new dimension values with the old ones in the slice. In this case, you must build a *new* slice containing all the data that you want to access. You must also build a new slice if you add a measure. Since slice files tend to be large, you should also provide a new slice for a user who has missed several updates, rather than supplying all of the incremental slice files. When you build a new slice, users must create a new slice instead of updating the old one.

The process for updating a slice

The following procedure describes the process that you use to update a slice:

1. Creating new data cube and slice definitions that pertain specifically to updating the slice.

Note: When creating new saved selections for a slice, the Time dimension should be saved as a script; other dimension selections should be saved as a selection list.

2. Updating the slice, using the slice definition that you have defined in Step 1.

Tools for updating a slice

You can use Analyzer or Database Manager as a tool to redefine the data cubes that make up a slice, as well as to redefine the slice itself. For more information, refer to the topic [“Defining a Slice” on page 10-3](#).

To *update* the slice, you can use either Application Manager or Express commands and Application Manager, depending on the method you choose.

Recommendation for updating slices

Before you distribute updated slices to your user community, you should set the MCALCINT property in the Source catalog (SOCAT) to NO.

Methods for updating a slice

The following table describes the two methods that can be used to update a slice.

Method	Description	Advantages
Updating a slice directly	Using Application Manager, you can pipe the data directly from the source database into a slice database. Note: To update a slice directly from a remote database, you must access the database in thick client mode.	You can easily update a slice on a LAN server for users to share. Or, users can maintain their own slices. This method has a Windows interface and does not require access to the command line of the operating system.
Updating a slice from file	Using Express commands, you can distribute slice EIF files to your users via diskettes or a LAN server. Then, you or your users can use Application Manager to update a slice from the files.	You can update a large slice in batch mode. Users do not need remote communications and are not involved in downloading the slice.

Updating a Slice Directly

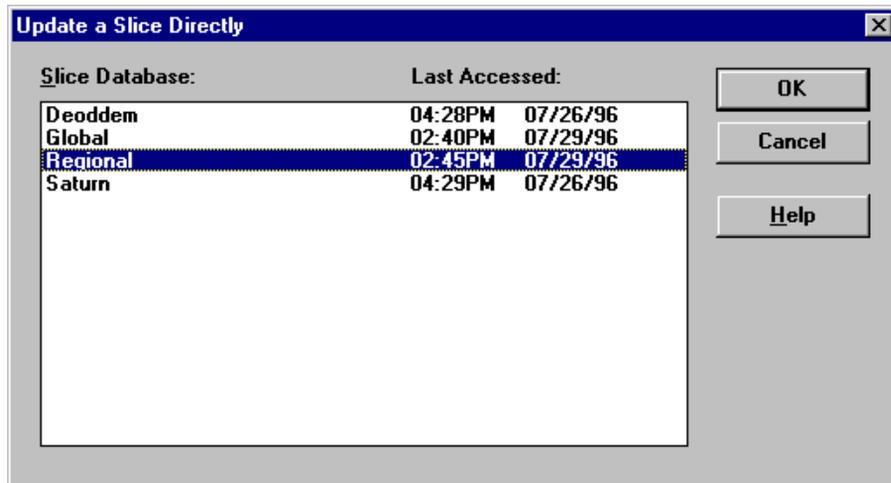
Overview of updating a slice directly

Before users can update a slice directly, you must copy the new data cube or cubes to the appropriate public libraries, just as you did when creating the slice. Then, users can update a slice directly by choosing the **Update a Slice Directly** function from the Application Manager Tools menu.

After users have selected the name of the slice to be updated in the Update a Slice dialog box, the Choose a Slice (Update) dialog box is displayed. Users can then update the slice using the definition that is already associated with the slice, or using a new slice definition.

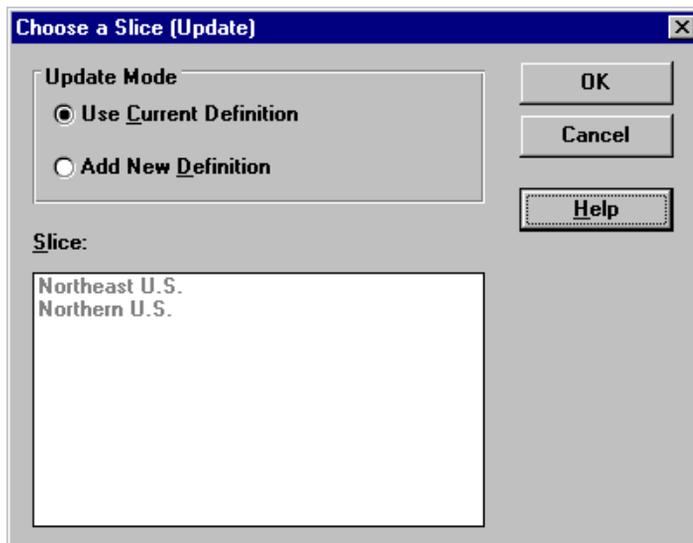
Example: Update a Slice Directly dialog box

The following is an example of the Update a Slice Directly dialog box, with the Regional database selected as the slice to be updated.



Example: Choose a Slice (Update) dialog box

The following is an example of the Choose a Slice (Update) dialog box. In the example, the slice is being updated with the current definition.



Related information

For more information about updating a slice directly, search for the following topic in the Application Manager Help system.

“Updating Slices Directly from Source Databases”

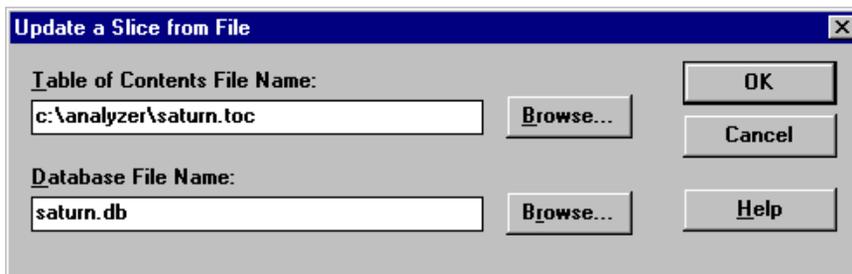
Updating a Slice from Files

Overview of updating a slice from files

Using Express, you can create EIF files on the remote system or LAN and distribute them to users, as described in the topic [“Creating a Slice from Files” on page 10-10](#). Users can then update a slice from the EIF files in Application Manager, using the **Update a Slice from File** option on the Tools menu.

Example: Update a Slice from File dialog box

The following is an example of the Update a Slice from File dialog box, with the table of contents and database file names selected.



Related information

For more information about updating a slice from files, search for the following topic in the Application Manager Help system:

“Updating Slices from Files”

Upgrading a Slice

Introduction

A slice that was created with a recent release of Analyzer can be upgraded on a PC for use with the current Analyzer version.

Procedure: Upgrading a slice

To upgrade a slice, perform the following steps:

1. Copy or restore the slice and personal databases to the appropriate directories for use with the current Analyzer release.
2. From the Application Manager File menu, choose **Database Setup**.

Result: The Database Setup dialog box is displayed.

3. Define a database setup for the slice, as described in [Chapter 5](#).

Note: Ensure that the access mode in the configuration file designates a slice database with read/write access.

4. Open the database from Analyzer, using the database setup that you have defined in Step 3.
5. Close the database.
6. If users will share the slice, modify the access mode in the configuration file to designate a slice database with read-only access.

Related information

For more information about editing configuration files, search for the following topic in the Application Manager Help system:

“Editing Configuration Files”

Customizing a Database

Chapter summary

This chapter describes how to modify an Analyzer database to tailor it to your users' requirements for data analysis.

List of topics

This chapter includes the following topics:

- [Changing Dimension Value Names](#)
- [Changing Level Names](#)
- [Providing Permanent Custom Aggregates and Measures](#)
- [Hiding Measures](#)
- [Modifying Aggregation Methods](#)
- [Configuring Databases for Forecasting](#)
- [Adjusting for Fluctuations in Time](#)
- [Defining the Latest Time Period](#)
- [Modifying Translation Tables](#)

Changing Dimension Value Names

How Analyzer provides names for dimension values

When displaying data, Analyzer uses the values of variables dimensioned by the base dimensions to provide descriptive names for dimension values. The name variables are also dimensioned by a language dimension (named LANG), which allows you to customize your database for a multilingual user base.

Short, medium, and long names

Each base dimension has three sets of names that can be used in reports and graphs. When defining a report or graph, users can select the type of name they want to use.

The following table describes short, medium, and long names.

Type of Name	Length	Description
Short	Up to 16 characters	Short names appear most frequently in graphs.
Medium	Up to 30 characters	Medium names are not used in the default definition of reports or graphs, but you can choose to use them instead of the default long or short names.
Long	Up to 70 characters	Long names appear in reports and Selector lists.

Name variables for Analyzer

The following table lists the name variables that Analyzer uses for common base dimensions.

Note: The dimension abbreviations in the table are “P” for Product, “G” for Geography, and “T” for Time. Note that the dimension abbreviations in your database might be different.

Dimension	Long Names	Medium Names	Short Names
PROD	NAME.P	MNAME.P	SNAME.P
GEOG	NAME.G	MNAME.G	SNAME.G
TIME	NAME.T	MNAME.T	SNAME.T
other (e.g., OUTLETS)	NAME.x (e.g., NAME.O)	MNAME.x (e.g., MNAME.O)	SNAME.x (e.g., SNAME.O)

Changing names for dimension values

In certain databases, the names of dimension values may not be meaningful to end users. If you are not satisfied with the current dimension value names, you can change them by populating similar “DBA” variables, and then running a program to copy them to NAME.x, MNAME.x, and SNAME.x.

If a dimension in your database has a large number of values, you can write an Express program that populates the variables, as an alternative to entering the individual names manually. An Express program is also useful if you want to construct names using a different method than that used by Analyzer.

The DB.MERGE.DBANAME utility (which resides in the DSBUILD database) moves customized names from the “DBA” variables to the corresponding standard Analyzer name variables.

The following procedure describes how to use DB.MERGE.DBANAME to copy modified dimension value names to the server database.

Procedure: Changing names for dimension values

To copy modify names for dimension values to the server database, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. Write an Express program to enter values for custom names into the following variables, where x is the one-letter abbreviation for the dimension.

Example:

```
DBANAME.x    Long name
DBAMNAME.x  Medium name
DBASNAME.x  Short name
```

Note: Valid characters for use in dimension value names are A-Z, a-z, 0-9, period (.) and underscore (_). Do not use special characters, start the name with a number, or include spaces. Also, do not use an Express reserved word. (You can obtain a list of reserved words by typing SHOW RESERVED in Express.)

4. Issue an UPDATE command to save your program in the catalog database, and then run your program to verify that the contents of the “DBA” variables are satisfactory.
5. Detach the server database and reattach it first with read/write access. Attach the DSBUILD database last with read-only access.

Example:

```
->database detach dsaturn
->database attach dsaturn first rw
->database attach DSBUILD last
```

6. Copy the DBA variables into the standard name variables for Analyzer.

Example:

```
->call db.merge.dbaname('prod')
```

7. Type update to save your changes to the server database.
8. Detach the server database and reattach it with read-only access.
9. Choose **Close** to close the Express Command Window and return to Database Manager.
10. Exit Database Manager and then reopen it to perform a master startup.

Related information

For information about customizing dimension value names in the source data, refer to the documents listed in the following table.

IF you use . . .	THEN refer to . . .
Relational Access Manager,	the <i>Oracle Express Relational Access Manager User's Guide</i> .
the data loader,	the <i>Oracle Sales Analyzer Data Loader Guide</i> .

Changing Level Names

How Analyzer stores dimension level names

Analyzer stores the names of dimension levels as they appear to users in variables named NAME.xLV, where *x* is the one-letter abbreviation for the dimension. These variables have two dimensions:

- A level dimension, named xLV, where *x* is the one-letter abbreviation for the dimension.
- A language dimension, named LANG, which allows a separate entry for each language required by your user community.

For example, if the abbreviation for Time in your database is T, the levels for the Time dimension are stored in NAME.TLV, which is dimensioned by TLV and LANG. Analyzer uses TLV internally when it needs to use time levels; it only uses NAME.TLV for display.

Procedure: Changing level names

To change the names for dimension levels as they appear to users, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. Detach the server database and reattach it first with read/write access.

Example:

```
->database detach dsaturn  
->database attach dsaturn rw first
```

4. Enter values in the NAME.xLV variable for the dimension whose level names you wish to change. The following example changes the name for a Time level

using a QDR. The example assumes that T is the abbreviation for the Time dimension.

Example:

```
->name.tlv(tlv 'QUADWEEK' lang 'ENU') = 'Month'
```

5. Use a qualified data reference (QDR), to enter the new level name in CH.HIERLVL.CAT.
6. Type `update` to save your changes to the server database.
7. Detach the server database and reattach it with read-only access.
8. Choose **Close** to close the Express Command Window and return to Database Manager.
9. Exit Database Manager and then reopen it to perform a master startup.

Translating level names for custom aggregates and custom measures

In the Selector, the level name for a custom aggregate level for all dimensions defaults to “Custom Aggregate,” regardless of the language that was specified when the database was built. Similarly, the level name for a custom measure defaults to “Custom Measure.” You can write an Express program to make these descriptions language-specific.

Important: If you write a program to translate level names for custom aggregates and custom measures, you must rerun it *after* each data update and *before* users access the new server database. Otherwise the new server database will not retain the translated names.

Procedure: Translating level names for custom aggregates and custom measures

To create an Express program that translates level names for custom measures and custom aggregates, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Verify that the server database is attached read/write.

```
->dtb list
```

If the database is attached read-only, detach the database and reattach it read/write using the following commands:

Example:

```
->database detach dsaturn  
->database attach dsaturn rw first
```

4. List all dimensions in the database and their abbreviations.

```
->limit dmentry to all  
->rpr dncat(dmpprop 'ABBREV')
```

5. For each of the dimensions, type the following:

```
NAME.xLV (xLV 'CUSTAGG' = 'language-specific description')
```

Substitute the appropriate abbreviation for *x* into the command (in two places) and substitute your language-specific description. For the measure dimension, the description should be a translation of the phrase “Custom Measure.” For all other dimensions, the description should be a translation of the phrase “Custom Aggregate.”

6. Type `update` to save your changes to the server database.
7. Detach the database and re-attach it read only. Use the following commands:

Example:

```
->dtb detach dsaturn  
->dtb attach dsaturn ro
```

8. Choose **Close** to close the Express Command Window and return to Database Manager.
9. Exit Database Manager and then reopen it to perform a master startup.

Related information

For information about specifying level names in the source data, refer to the following documents.

IF you use . . .	THEN refer to . . .
Relational Access Manager,	the <i>Oracle Express Relational Access Manager User's Guide</i>
the data loader,	the <i>Oracle Sales Analyzer Data Loader Guide</i>

Providing Permanent Custom Aggregates and Measures

Custom aggregates and measures as library objects

When you copy custom aggregates to a public library, Analyzer adds them as values in the appropriate base dimension in the server database, and populates the associated variables whenever a user starts up the database. Similarly, Analyzer defines custom measures as objects in the server database and performs other tasks necessary to support the measures. Since the server database is attached read-only, these changes exist only at runtime. Every time a user opens a database, custom aggregates and custom measures are installed. Thus, as the number of custom aggregates and custom measures increases, starting up the database takes a little longer.

Custom aggregates and measures as permanent objects

If you have many public custom aggregates and custom measures, you may want to write them permanently to the server database. This should improve startup performance significantly for all of your users, since the objects are no longer installed at startup.

Overview of the DS.WRITE Program

You use the DS.WRITE program to write saved objects directly to the server database. DS.WRITE has one required argument, which has a value of AG for custom aggregates or DA for custom measures. DS.WRITE permanently writes custom aggregates and custom measures to the database based on their 'PERMANENT' property catalog setting. If you later decide to remove all objects

from the server database, you must rerun DS.WRITE with the optional RESET argument.

DS.WRITE has the following syntax.

```
CALL DS.WRITE ('object-type' [ 'RESET' ])
```

where “object-type” can be AG for custom aggregates or DA for custom measures.

Important: An error in the data load will result if permanently written objects are left in the server database. Therefore, you must run the DS.WRITE program with the RESET argument before each data load or build. Then re-run it with appropriate arguments *after* the build to re-write objects to the database. You might want to add the DS.WRITE commands into pre- and post- load programs.

Procedure: Writing custom aggregates and measures to the server database

To write custom aggregates and custom measures to the server database, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.

Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Attach the public database.

Example:

```
->database detach dsaturn  
->database attach msaturn rw first
```

4. Optionally, you can view descriptions of custom aggregates and custom measures in your database by issuing the following commands.

```
->limit pb.ag.prop to 'NAME.LONG'  
->report down pb.ag.entry w 30 pb.ag.catalog
```

5. Record the values that match the names of the objects that you wish to use.

6. To permanently write a custom aggregate or custom measure to the database, perform the following steps:
 - a. For each custom aggregate that you want to write to the server database, set the PERMANENT property of PB.AG.CATALOG to YES. The following example uses a qualified data reference (QDR) to set the PERMANENT property to YES for a custom aggregate named MYAGG.

Example:

```
->pb.ag.catalog -  
Continue>(pb.ag.prop 'PERMANENT', -  
Continue>pb.ag.entry 'MYAGG') = 'YES'
```

- b. For each custom measure that you want to write to the server database, set the PERMANENT property of PB.DA.CATALOG to YES. The following example uses a qualified data reference (QDR) to set the PERMANENT property to YES for a custom measure named DBA43.

Example:

```
->pb.da.catalog -  
Continue>(pb.da.prop 'PERMANENT', -  
Continue>pb.da.entry 'DBA43')='YES'
```

7. Type `update` to save your changes.
8. Write the custom aggregates and custom measures that you have identified in Step 5 to the server database.
 - a. To write custom aggregates to the server database, issue the following command.

```
->call ds.write('AG')
```
 - b. To write custom measures to the server database, issue the following command.

```
->call ds.write('DA')
```
9. Choose **Close** to close the Express Command Window and return to Database Manager.
10. Exit Database Manager and then reopen it to perform a master startup.

Procedure: Removing permanent custom aggregates and custom measures

The DS.WRITE program with the RESET argument removes *all* permanent custom aggregates and measures from the server database. To retain specified custom aggregates and measures, and then add them back to the server database, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. Remove all custom aggregates and custom measures from the server database.
 - a. To remove custom aggregates, issue the following command.

```
->call ds.write('AG' 'RESET')
```
 - b. To remove custom measures, issue the following command.

```
->call ds.write('DA' 'RESET')
```
4. Type `update` to save your changes.
5. Add the custom aggregates and custom measures that you want to keep back into the server database. Follow the instructions in Step 6 in the topic [“Procedure: Writing custom aggregates and measures to the server database” on page 11-9](#).
6. Choose **Close** to close the Express Command Window and return to Database Manager.
7. Exit Database Manager and then reopen it to perform a master startup.

Hiding Measures

Overview

You can prevent a measure from appearing to end users. You might want to do this if you have a custom measure that is based on one or more other measures that you do not want users to see.

Important: You must rerun the procedure to hide measures *after* each data update and *before* users access the new server database.

Procedure: Hiding measures

Use the following procedure to hide measures:

1. From Database Manager, open the database.

Result: Database Manager performs a master startup.

2. From the Database Manager Help menu, choose **Express Command Window**.

Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Verify that the server database is attached read/write.

```
->dtb list
```

If the database is attached read-only, detach the database and reattach it read/write using the following commands:

Example:

```
->database detach dsaturn  
->database attach dsaturn rw first
```

4. List the measures in the database.

```
->RPR DOWN MEAS W 30 NAME.M
```

5. Limit the measures to those that you want to hide.

```
->LIMIT MEAS to MEAS1NAME,MEAS2NAME  
->LIMIT DAENTRY TO CHARLIST(MEAS)
```

6. To prevent a measure from displaying to end users, issue the following commands:

```
->DACAT(DAPROP 'HIDE') = 'YES'  
->DACAT(DAPROP 'TIME MODIFIED') = DS.TIMESTAMP
```

7. Update SOCAT to force a refresh of the public catalogs in the next master startup:

```
->MAINTAIN MEAS DELETE CHARLIST(MEAS)  
->SOCAT(SOPROP 'TIME MODIFIED')=DS.TIMESTAMP
```

8. Choose **Close** to close the Express Command Window and return to Database Manager.
9. Exit Database Manager and then reopen it to perform a master startup.

Modifying Aggregation Methods

How Analyzer calculates custom aggregates

The Data catalog includes properties that define an *aggregation method* for each base dimension. Analyzer uses these aggregation methods in the MCALC function to calculate custom aggregates for each measure.

Analyzer provides default aggregation methods, which you can override by modifying values of the Dx.MCALC.DBA properties in the public database Data catalog (PB.DA.CATALOG). To find out the name of the base dimension associated with D1, D2, and so forth, you can look at the Dx.DIM.ID properties in PB.DA.CATALOG.

```
->limit pb.da.entry to all
->rpr group pb.da.entry w20 down pb.da.prop w50 pb.da.catalog
```

Overview of aggregation methods

Each aggregation method is an expression of the form:

```
method OF members [WEIGHTBY weight]
```

The following table describes the two arguments in this expression.

Argument	Description
"method"	This is the aggregation method. The "method" argument accepts one of the following keywords: AVERAGE, AVERAGEWEIGHTBY (expression), FIRST, LAST, LARGEST, TOTAL, and NONADD.
"members"	This is the members list for the base dimension whose values are being aggregated.

As an example, you might modify the D1.MCALC.DBA, D2.MCALC.DBA, and D3.MCALC.DBA properties in PB.DA.CATALOG to contain the following expressions.

```
D1.MCALC.DBA: LAST OF MEMBERS.T
D2.MCALC.DBA: AVERAGE OF MEMBERS.G WEIGHTBY F.MEASNAME
D3.MCALC.DBA: TOTAL OF MEMBERS.P
```

Note: You can provide more than one aggregation method, listing them one per line. Or you can use the Express `joinlines` command to create a multi-line expression. Analyzer attempts to use the first method. If this method fails, Analyzer uses each method successively until it succeeds in performing the aggregation. Multiple aggregation methods are particularly useful when calculating a weighted average; you can specify an alternative measure on which to base the weights when the first one is not available in the database.

The “method” argument

The method argument accepts one of the keywords that are described in the following table.

Keyword	Description
AVERAGE	Analyzer can calculate either a simple average or, using the WEIGHTBY argument, a weighted average. It uses the following formula, in which the <i>sign</i> is identified in the members list. $\text{Sum of (weight * value * sign) / Sum of (weight * sign)}$
FIRST	This keyword returns the first value in the members list. Use this aggregation method only for the Time dimension.
LAST	The LAST keyword returns the last value in the members list. Use this aggregation method only for the Time dimension.
LARGEST	The LARGEST keyword returns the largest value.
TOTAL	Calculates a simple sum. Positive and negative signs in the members lists are applied to the values.
NONADD	Returns NA in all cases.

Using the AVERAGE keyword

Consider the following example, in which a custom aggregate for the Geography dimension contains BOSTON, NEWYORK, and NORTHEAST in its members list.

Dimension Value in Members List	Sign in Members List	Weight	Data Value
BOSTON	- (Negative)	1	15
NEWYORK	- (Negative)	3	75
NORTHEAST	+ (Positive)	6	325

Note that BOSTON and NEWYORK are subsets of NORTHEAST, that is, their values contributed to the aggregate NORTHEAST value.

Analyzer calculates the weighted average as follows.

$$\frac{((1*(15)*(-1)) + (3*75*(-1)) + (6*325*(+1)))}{((1*(-1)) + (3*(-1)) + (6*(+1)))}$$

or

$$(-15 -225 + 1950) / (-1 -3 + 6) = 855$$

The weights default to one (1) for a simple average, which can cause problems in custom aggregates with negative signs in the members list. When a custom aggregate contains as many negative members as positive members, the result is always zero because of the attempt to divide by zero. Likewise, if there are more negative members than positive ones, the result will be negative instead of positive. Simple averages are seldom used in practice because weighted averages almost always give more meaningful results.

If you get a value of zero or a negative number for a weighted average, check the members lists to make sure that all negative values are subsets of positive values. An example is given in this section under “Members Lists.”

If a value has a weight of NA, that value is omitted from the calculation. If all weights are NA, then zero is returned.

The “members” argument

The “members” argument is a text variable dimensioned by one of the base dimensions. For all of the regular dimension values, the members variable contains a value of NA. For a custom aggregate, it contains a list of dimension values to be aggregated, which are stored as text with one value per line. A custom aggregate cannot use another custom aggregate in its members list.

As shown in the following table, Analyzer defines a members list for each base dimension using the dimension’s unique abbreviation as a suffix to the variable name MEMBERS.x.

Note: The dimension abbreviations in the table are “P” for Product, “G” for Geography, and “T” for Time. Note that the dimension abbreviations in your database might be different.

Dimension	Members List
PROD	MEMBERS.P
GEOG	MEMBERS.G
TIME	MEMBERS.T
other	MEMBERS.x

Using positive and negative values in a members list

A value in a members list can be either positive (+) or negative (-). When signs are used, they are used on all members of a custom aggregate. The sign precedes the value in the members lists. A member without a sign has a positive value.

A member with a negative sign must be part of a value at a higher level in the dimension (i.e., an embedded total), which is also a member of this custom aggregate. For example, you might want to define an aggregate that removes the Midwest region from the total USA. The value for Midwest is contained in the value for the total USA. You would define the custom aggregate so that total USA is a positive member and Midwest is a negative member.

By verifying that a negative member is a subset of a positive member, you can avoid making a common error: using a negative to make a comparison, such as +NY -BOSTON to try to find the difference between these two cities. You would only get valid results when the aggregation method is TOTAL. Since custom aggregates are used with all measures, an improperly defined custom aggregate will produce meaningless results for at least some measures.

Procedure: Modifying aggregation methods

Important: The recommended procedure for modifying aggregation methods is to do so in a build or update, as described in your documentation for Relational Access Manager or the data loader. The following is an alternative procedure only.

1. From Database Manager, open the database.

Result: Database Manager performs a master startup.

2. From the Database Manager Help menu, choose **Express Command Window**.

Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Ensure that DSBUILD is attached last.

```
->database attach dsbuild last
```

4. Detach the server database; then reattach it first with read/write access.

```
->database detach dsaturn
->database attach dsaturn rw first
```

5. Modify the Data catalog (DACAT) by performing the following steps:

- a. Enter your custom aggregation methods. The following example uses a QDR to set the aggregation method of the first dimension (TIME, with the abbreviation T) of a measure (MEASNAME).

Example:

```
->dacat(daprop 'D1.MCALC.DBA', -
Continue>daentry 'MEASNAME') -
Continue>= 'LAST OF MEMBERS.T'
```

- b. Set the MODIFIER and TIME.MODIFIED properties.

Example:

```
->dacat(daprop 'MODIFIER', -
Continue>daentry 'MEASNAME') = 'DBA'
->dacat(daprop 'TIME.MODIFIED'),
Continue>daentry 'MEASNAME') = ds.timestamp
```

6. For a database built with the data loader, set the Source catalog entry and then redefine the formula by running DB.FORMULA.DRV.

```
->limit DB.EXTSOURCE to 'EXTRACDR'
->call db.formula.drv
```

7. For a database built with Relational Access Manager, attach EXTRACSQ and READWH last and then run SQ.PARSE.WEIGHT.

```
->database attach EXTRACSQ last
->call sq.parse.weight
```

8. If the database has already been released to your users, then you must force a refresh by ensuring that the server database is attached read/write and then updating the time stamp in SOCAT.

```
->socat(soprop 'TIME.MODIFIED') = ds.timestamp
->update
```

9. Type `update` to save your changes to the server database.
10. Choose **Close** to close the Express Command Window and return to Database Manager.
11. If you are modifying aggregation methods before you have released the updated database to your users, you can end the procedure. Your changes will be reflected in PB.DA.CATALOG. If you wish, you can synchronize DACAT and PB.DA.CATALOG by performing Step 12.

If you are modifying aggregation methods after you have released the updated database to your users, continue with Step 12.

12. Choose **Close** to close the Express Command Window and return to Database Manager.
13. Exit Database Manager and then reopen it to perform a master startup.

Result: DACAT (in the server database) and PB.DA.CATALOG (in the public database) will be synchronized. When users open the database from their PCs, a refresh will occur and the changes will be integrated in DA.CATALOG (in the personal database) as well.

Related information

For information about specifying aggregation methods in the source data, refer to the documents listed in the following table.

IF you use	THEN refer to
Relational Access Manager,	the <i>Oracle Express Relational Access Manager User's Guide</i> .
the data loader,	the <i>Oracle Sales Analyzer Data Loader Guide</i> .

Configuring Databases for Forecasting

General forecast requirements

An Analyzer forecast is the estimate of a measure's future value based on its historic values within a supporting time range. If your users will be creating forecasts, the database must include the future time periods to be forecast. You must also ensure that null values will be displayed as NA rather than as zero.

Making forecasting adjustments for a data loader database

Adding time periods

If you use the data loader program to build an Analyzer database from flat files, you must add the future time periods to the Time parentage file. The future time periods should not contain any data.

Adding and adjusting control file keywords

You must also add the JLATEST keyword to the control file and set its value to the last time period that contains valid data for the lowest time level. When you specify a value for JLATEST, the LATEST.x parameter automatically sets logical values for all higher time levels.

In addition, if you have specified values for the SYNONYM and RETAIN TIME keywords, you must adjust them so that the correct time values will be used when you read in data and roll off time periods.

Ensuring that null data is displayed as NA

To ensure that null data is displayed as NA rather than as zero, you must set the SCREENBY property for each measure in the Measure file to DATA.

Making forecasting adjustments for a Relational Access Manager database

Adding time periods

If you use Relational Access Manager to access data that is stored in an RDBMS, then you must add future time periods to the lookup tables in the RDBMS. The future time periods should not contain any data.

It is not necessary to set a special parameter for the latest time period that contains data; Relational Access Manager uses the date of the last build or update to set this property.

Ensuring that null data is displayed as NA

To ensure that null data is displayed as NA rather than as zero, you must set the Handle Null Data Option property in the Relational Access Administrator Project to NA (the default). You set this option in the Data Model Tab.

Related information

For detailed information, refer to the following:

Technical Note: Preparing Analyzer Databases to Be Used for Forecasting

Oracle Express Relational Access Manager User's Guide

Oracle Sales Analyzer Data Loader Guide

Relational Access Administrator Help system

Adjusting for Fluctuations in Time

How Analyzer manages periodicity

Analyzer can detect and handle a wide range of time periods, including daily data and 5-day weeks (or any other definition of “week” that you wish to use). Analyzer handles most issues concerning the periodicity of your data automatically.

Consistent time periods

Analyzer requires consistency among the time periods in order to sort them correctly. It groups similar time periods into *chunks* so that it can perform lag-type analyses. For example, Analyzer groups all 1-week periods into one chunk, all 4-week periods into a second chunk, all 12-week periods into a third chunk, and so forth, and sorts each chunk chronologically. This allows Analyzer to identify a prior 4-week period correctly, for example, instead of finding a 1-week period because it is the previous dimension value. Remember that the Time dimension often has a 4-week total immediately after the four 1-week periods that contribute to the 4-week's aggregate value.

Adjusting time periods (data loader)

Because of fluctuations in the calendar, you may have to make adjustments to account for irregularities in time periods. For example, if you aggregate your data into 4-week periods, you must define a 5-week period and thus a 53-week year about once every 7 years. However, you will want this 5-week period to be in the same chunk as the 4-week periods, and not be relegated to a separate chunk by itself.

You must allow for this type of fluctuation, either in the form of gaps or overlaps in your data, while building the server database. For example, in the data loader control file, you can set the ALLOWGAPSINCHUNK parameter to YES. (By default, it is set to NO.)

Controlling prior time periods (data loader)

Similarly, you can control the way Analyzer identifies a prior period. By default, the ending date of the prior period must immediately precede the beginning date of the current period. However, if you wish the time periods to be able to overlap, you can set the ALLOWGAPSINPRIOR keyword to YES. (By default, it is set to NO.)

Defining the Latest Time Period

Introduction

Analyzer allows the user to select values based on time periods. For example, the user can use the Range tool in the Selector to select time values based on a specific number of the most recent time periods or a beginning time period through the current period. If the current month is June, 2002, and the user selects the “latest four months,” the user expects to select the values of March through June, 2002.

How Analyzer identifies the current time period

Analyzer stores the current time period for each level in the Time dimension in the variable LATEST.x, where x is the abbreviation for the Time dimension. By default, this variable stores as the current time period the last time period for each level in the Time dimension.

Databases that support forecasting typically have future time periods in the Time dimension. In this case, if the user selects the “latest four months,” the selected time values will be based on a future time period rather than the current period. To provide the user with expected results, you must ensure that the object LATEST.x is populated with the current time period (that is, the most recent time period for which data exists).

Setting the latest time period property

If you use the data loader to build and update your database, you can use the JLATEST keyword in the control file to set the latest time period. For more information, see the *Oracle Sales Analyzer Data Loader Guide*.

If you use Relational Access Manager (or, if you use the data loader but do not set the latest time period in the control file), then LATEST.x will be set to the last time period in the database by the build or update. In such case, before making the database available to your users, you can reset the property with the current time period.

Procedure: Resetting the current time period

To reset the current time period, perform the following steps:

Note: This procedure assumes that the abbreviation for the Time dimension in your database is T. If this is not the case, substitute the actual abbreviation for T.

1. From Database Manager, open the database.
2. From the Database Manager Help menu, choose **Express Command Window**.

Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Detach the server database and reattach it first with read/write access. Attach the DSCODE database last with read-only access.

Example:

```
->database detach dsaturn  
->database attach dsaturn first rw  
->database attach DSCODE last
```

4. To set the latest values for all levels in the Time dimension, call the function DS.SETLATEST. The syntax is DS.SETLATEST ('date'), where the input parameter (date) is the date to use as the setting for LATEST.T. In the following example, the input parameter is May 18, 2002.

Example:

```
->call ds.setlatest('2000518')
```

Result: For each level in the Time dimension, LATEST.T is set to a dimension value, as follows.

Note: If two or more dimension values in the same level have the same date, then LATEST.T is set to the first of the two dimension values.

IF . . .	THEN LATEST.T is set to . . .
there is a dimension value whose date is equal to the input parameter (for example, March 18, 1998),	that dimension value. For example, LATEST.T might be set to T15 (the week ending May18, 2002).
there is no dimension value whose date is equal to the input parameter,	the dimension value whose date is closest to and greater than the input parameter. For example, LATEST.T might be set to T30 (May 02), because its ending date is May 31, 2002.
there is no dimension value whose date is equal to or greater than the input parameter,	the current setting. (From the database build, this is the last value in the Time dimension.)

5. Type `update` to save your changes to the server database.
6. Detach the server database and reattach it with read-only access.
7. Choose **Close** to close the Express Command Window and return to Database Manager.
8. Exit Database Manager and then reopen it to perform a master startup.

Modifying Translation Tables

Overview of character translation

Before characters can be displayed to the Analyzer user, they are matched against the character set that applies to the user's platform. Because character sets often vary between platforms and applications, it is essential that characters are translated properly between character sets.

When translation is unsuccessful

Unsuccessful translation is common when translating between single-byte (256 possible characters) and double-byte (32,768 possible characters) character sets. However, problems can also arise when different platforms or applications pass information using different single-byte character sets. Most single-byte character sets use the standard ASCII characters for the numbers 0-127. However, two different single-byte character sets may use different extended characters (128-255) in order to support specific characters such as the French "é."

Translation of a particular character between two character sets will most likely be successful if that character exists in both sets. However, translation will be unsuccessful if that character exists only in one of the sets.

How Analyzer translates characters

The following table describes how characters are translated by Analyzer and Express, and whether you can modify character translation.

Translation	Methods Used	Modifying Translation
Between Analyzer and Personal Express	Analyzer uses the ANSI-standard character set. Personal Express uses whatever DOS/OEM character set (or <i>code page</i>) has been specified on the user's PC.	Translation is not modifiable. Translation of characters between Analyzer and Personal Express should pose no problem as long as valid characters have been used.
Between Personal Express and Express Server on a remote system	In thick client mode, Personal Express and Express Server exchange data via Express Communications Architecture (or <i>XCA</i>). Express Server uses translation tables to translate characters to and from the remote system's character set.	Translation is modifiable. Modifications may be necessary to allow invalid characters to be translated to and from the remote platform successfully.

Example: Unsuccessful translation in Analyzer

For example, consider a custom aggregate that uses an extended character in its name. Aggregate information is stored in the user's local personal database; however, the aggregate dimension value is stored in the server database on the remote system. The user might be able to view the name of the aggregate properly in the Custom Aggregates dialog box, since this name is read from local catalogs and thus relies on translation between Analyzer and Personal Express. However, the aggregate dimension value may not appear properly in a report or graph, since this value is read from the server database on the remote system, and thus relies on translation between Personal Express and Express Server. By modifying translation tables in Express Server, you can define new tables that will provide successful character translation.

Procedure: Modifying translation tables

To modify translation tables in Express Server, perform the following steps. Perform this procedure on the remote system *only*:

1. Make a connection to initiate a session of Express.

Note: In order to modify translation tables, you must have write access to the remote Express database.

2. Define two 256-byte text variables to be used as translation tables by performing the following steps:

- a. Define a variable for translations inbound to Express Server. In the following example, the variable is named INTBL.

Example:

```
->define intbl text  
->ld 256-byte translation table (in)
```

- b. Define another variable for translations outbound from Express Server. In the following example, the variable is named OUTTBL.

Example:

```
->define outtbl text  
->ld 256-byte translation table (out)
```

3. Populate the two translation table variables that you have created in Step 2. For a description and example of this step, refer to [“Populating translation table variables” on page 11-27](#).
4. Add a value to the system dimension `_XLTID` for your combined inbound and outbound translation tables. (In this example, the value is named MYXLATE.)
5. Use a QDR to set the value of the system variable `_XLTABLE` that corresponds to the dimension value that you created (in this example, MYXLATE). The value will include the INTBL and OUTTBL values, respectively.

Example:

```
->_XLTABLE(_XLTID -  
Continue>'MYXLATE')=joinlines(INTBL,OUTTBL)
```

- Set the value of the system option `_XLTEIF` to the name of the dimension value that you created for your combined inbound and outbound translation tables (in this example, `MYXLATE`).

Example:

```
->_XLTEIF='MYXLATE'
```

- Type `update` to save your changes to the server database.

Populating translation table variables

Step 3 of the procedure for modifying translation tables involves populating two translation table variables, so that the number of codes for each is 256. The first slot in each table corresponds to the first character of the standard character set, the second slot to the second character, and so on through all 256 characters in the standard order. As you populate each variable, identify the positions of the characters that you would like to replace, and replace the codes with the appropriate ones.

Example: Populating translation table variables

The following example shows the command that you might use to populate the variable `INTBL`. This example uses hexadecimal escape sequences, which are described in the glossary entry “hexadecimal escape” in the *Oracle Express Language Reference Manual*.

```
INTBL = joinchars( -
'\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f' -
'\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f' -
'\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f' -
'\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f' -
'\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f' -
'\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f' -
'\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f' -
'\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f' -
'\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f' -
'\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f' -
'\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf' -
'\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf' -
'\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf' -
'\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf' -
'\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef' -
'\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff' -
)
```

Controlling the Display of Data

Chapter summary

This chapter explains how you can control the format and presentation of data that is displayed by Analyzer.

List of topics

This chapter includes the following topics:

- [Attributes for Displaying Measure Values](#)
- [Formatting Measure Values](#)
- [Installing Measure Value Formats](#)
- [Changing the Order of Dimension Values](#)
- [Setting Dimension Value Limits](#)
- [Specifying Default Documents](#)

Attributes for Displaying Measure Values

Introduction

You can change the default values for several attributes that control the display of measures. In doing so, you can control how Analyzer will display values for a specific measure. For example, you can prevent leading zeroes from displaying, or you can display a negative number in parentheses preceded by a dollar sign.

Methods for changing default attributes

Depending on the particular attribute, you will use one of the following methods to change a default value:

- Using the Regional Settings Properties in the Windows Control panel
or
- Installing the desired format in the Data catalog in the public database (PB.DA.CATALOG)

List of measure display attributes

The following table lists the measure display attributes that are under your control, where each default comes from, and the method that you can use to change it.

Display Attribute	Default	How to Change
Decimal separator	Determined by Regional Settings	Access the Regional Settings Properties dialog box from the Windows Control Panel.
Thousands separator	Determined by Regional Settings	Access the Regional Settings Properties dialog box from the Windows Control Panel.
Currency symbol	FORMAT property in the Data catalog. If no symbol is specified, determined by Regional Settings	Access the Regional Settings Properties dialog box from the Windows Control Panel or install the format for a measure in the Data catalog. The Data catalog takes precedence over Regional Settings.
Display or suppress leading zeros	Display leading zeros	Install the format for a measure in the Data catalog.

Display Attribute	Default	How to Change
Number of decimal digits	Two	Install the format for a measure in the Data catalog.
Negative Numbers	Parentheses, e.g., (100)	Install the format for a measure in the Data catalog.
Positive Numbers	No special character	Install the format for a measure in the Data catalog

Formatting Measure Values

Preparing a format statement

Before installing new formats in the Data catalog, you must prepare a format statement for each measure value that you wish to reformat. Later, you will use this format statement to edit the FORMAT property in the Data catalog. A format statement consists of four clauses separated by semicolons as follows:

number format; positive format; negative format; special characters

The number format, positive format, and negative format clauses are required; special characters are optional.

Example: Format statement

For example, the statement `0.0; 1$; -1$; $=$` is interpreted as follows:

- Number format: 0.0
- Positive format: 1\$
- Negative format: -1\$
- Special characters: dollar sign (\$)

Preparing the number format clause

The number format clause controls the display of leading zeros and trailing decimal places. It can contain the following formatting instructions:

- A zero (0) to display a leading zero, or a pound sign (#) to suppress it.
- A decimal point and from one to six zeros to identify the number of decimal places you wish to display, if any.

Example: Number format clause

The following table provides several examples of number format clauses and shows their effect on two data values, 5.6735 and .541. Data values are rounded to the number of decimal places allowed by the format.

Number Format	“5.6735” Data Value	“.541” Data Value
0	6	1
#.0	5.7	.5
0.0	5.7	0.5
#.000	5.674	.541
0.000	5.674	0.541

Preparing the positive format clause

The positive format clause affects only positive numbers, and enables you to include a positive sign (+) and a symbol placeholder (\$) whose value will be assigned in the special characters clause. The number “1” in this clause represents the data value.

You can specify the +, \$, and 1 to appear in any order; “1” is the only character that is required.

Example: Positive format clause

The following table provides several examples of positive format clauses and their effect on a data value of 5. Special character formats are discussed in this chapter.

Positive Format	Special Characters	“5” Data Value
1	unspecified	5
+1\$	\$=%	+5%
\$1	\$=\$	\$5
\$1+	\$=£	£5+

Preparing the negative format clause

The negative format clause affects only negative numbers. It enables you to include a negative sign or parentheses and a symbol placeholder whose value will be assigned in the special characters clause.

You can specify the +, \$, and 1 to appear in any order; “1” is the only character that is required.

Example: Negative format clause

The following table provides several examples of negative format clauses and their effect on a data value of -5.

Negative Format	Special Characters	“-5” Data Value
(1)	unspecified	(5)
-1\$	\$=%	-5%
(\$1)	\$=\$	(\$5)
\$(1)	\$=£	£(5)

Preparing the special character clause

A format statement can include an optional clause defining special characters such as a currency symbol or a percent symbol. This will override the default values for currency that are defined in the Regional Settings Properties dialog box in the Windows Control Panel.

You should always define the currency symbol explicitly in the format statement where appropriate; for example, you would do this to prevent a user’s Windows settings from changing U.S. dollars to U.K. pounds.

You can define special characters as follows, where the symbol following the equal sign is the special character that will be used to format the data:

- For the currency symbol, enter \$=\$
- For the percent symbol, enter \$=%

Example: Special character clause

The following table provides some examples of format statements with special character clauses and their effect on a data value of 5892.

Format Statement	“5892” Data Value
0.00;1;-1	5,892.00
0.00;1\$;-1\$;\$=%	5.89%
0.00;\$1;\$(1);\$=\$	\$5,892.00

Installing Measure Value Formats

Overview of installation

The format for particular data values for a measure is controlled by the `FORMAT` property of the Data catalog (`PB.DA.CATALOG`). It is possible to edit this property directly; however, for base measures, this catalog will be overwritten at each update. If you do not want to redo your format changes, you can populate a variable named `DBAFORMAT.M` with the formats you want applied to specific measures, and then run `DB.MERGE.DBANAME` to install the new formats in the Data catalog. This is the same program that you use to install custom names for dimension values.

`DBAFORMAT.M` is dimensioned by the `MEAS` dimension, which has all measures in your database as entries. For each measure to which you want to apply a special format, enter the formatting instructions in the form of a format statement in `DBAFORMAT.M`.

Procedure: Installing formats for measure values

To install formats for measure values in the Data catalog, perform the following steps.

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Detach the server database and reattach it first with read/write access. Attach the DSBUILD database last with read-only access.

Example:

```
->database detach dsaturn
->database attach dsaturn rw first
->database attach DSBUILD last
```

4. Enter formats into DBAFORMAT.M, using format statements with QDRs.

Example:

```
->dbaformat.m(meas 'SALES')='0.0;1$;-1$;${}='
```

5. Install the formats in the Data catalog in the public database.

```
->call db.merge.dbaname('MEAS')
```

6. Update the Source catalog and save your changes to the server database.

```
->socat(soprop 'TIME.MODIFIED')=ds.timestamp
->update
```

7. Detach the server database and reattach it with read-only access.
8. Choose **Close** to close the Express Command Window and return to Database Manager.

Working with custom measures

Simply changing DBAFORMAT.M does not work for custom measures, because custom measures are added when the database starts up. (The process of detaching and reattaching the server database removes these measures from the MEAS dimension, making it impossible to change DBAFORMAT.M.) In order to change the format of a custom measure, you must change the FORMAT property of PB.DA.CATALOG, as well as FORMAT.M in the custom measure programs. Since custom measures are not stored in DACAT, they will not be overwritten in PB.DA.CATALOG after an update.

Important: For permanently installed custom measures, you must also set FORMAT.M manually in the server database.

Procedure: Installing formats for custom measure values

To install formats for custom measures, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.

3. Detach the public database and reattach it first read/write.

```
->database detach msaturn  
->database attach msaturn rw first
```

4. Enter formats in PB.DA.CATALOG using assignment statements with QDRs, where MEASNAME is the name of the measure.

Example:

```
->pb.da.catalog(pb.da.prop 'FORMAT' pb.da.entry -  
Continue>'MEASNAME')='0.0;1$;-1$;$=%'
```

5. Edit the MD.measname and PD.measname programs, where “measname” represents the name of your measure. In each program, change the FORMAT.M line to reflect the changes you made in PB.DA.CATALOG.

Example:

```
FORMAT.M(MEAS 'MEASNAME')='0.0;1$;-1$;$=%'
```

6. Update the Source catalog and save your changes to the server database.

```
->socat(soprop 'TIME.MODIFIED')=ds.timestamp  
->update
```

7. Choose **Close** to close the Express Command Window and return to Database Manager.

Related information

If you use Relational Access Manager to build and update your database, refer to the *Oracle Express Relational Access Manager User's Guide* for information about specifying data value formats. If you use the data loader program, refer to the *Oracle Sales Analyzer Data Loader Guide*.

Changing the Order of Dimension Values

Overview

The order in which dimension values are displayed in the Selector and in reports is assigned by the build process. However, you can change the specified order by modifying any of three valuesets in the server database.

Applicable valuesets

The following table describes the valuesets that you can modify to change the order of dimension values.

Valuset	Description
UNIVERSEVS. <i>x</i> where <i>x</i> is the one-letter abbreviation for the dimension.	If the user does not specify using a hierarchy, this is the order in which dimension values are displayed.
ORDERVS. <i>xn</i> where <i>x</i> is the one-letter abbreviation for the dimension and <i>n</i> is the number of the hierarchy. For example, the fiscal-year and year-to-date hierarchies in the Time dimension might be represented by ORDERVS.T1 and ORDERVS.T2, respectively.	If the user specifies a hierarchy but does not turn on drilling, this is the order in which dimension values are displayed If the SortType defined in the dimension file in the build process is set to Level, dimension values will be displayed by level (e.g., all Years, followed by all Quarters, and then all Months). If the SortType is set to Hierarchy, dimension values will be displayed by order in the hierarchy (e.g., Year 1, followed by Quarter 1, and then Months 1-3).
HORDERVS. <i>xn</i> where <i>x</i> is the one-letter abbreviation for the dimension and <i>n</i> is the number of the hierarchy.	If the user specifies a hierarchy and turns on drilling, this is the reverse order in which dimension values are displayed. Dimension values will always be displayed by order in the hierarchy (e.g., Year 1, followed by Quarter 1, and then Months 1-3), regardless of the SortType defined in the dimension file in the build process.

Procedure: Changing the order of dimension values

To change the order in which dimension values are displayed, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. Detach the server database and reattach it first with read/write access.
->database detach dsaturn
->database attach dsaturn rw first
4. Limit the appropriate dimension to the valueset that you wish to modify.
Example:
->limit time to universevs.t
5. Use the SORT or LIMIT command to change the order of dimension values.
Example:
->sort time a date.t
6. Limit the valueset to the dimension.
Example:
->limit universevs.t to time
7. Type `update` to save your changes.
8. Detach the server database and reattach it with read-only access.
9. Choose **Close** to close the Express Command Window and return to Database Manager.

Related information

If you use Relational Access Manager to build and update your database, refer to the *Oracle Express Relational Access Manager User's Guide* for information about specifying the order of dimension values. If you use the data loader program, refer to the *Oracle Sales Analyzer Data Loader Guide*.

Setting Dimension Value Limits

Overview

The Maximum Values catalog (MX.CATALOG) contains default limits that affect user actions in the Selector. Using Database Manager, you can change the default limits on the following:

- The number of dimension values allowed in reports and graphs and the number of members allowed in custom aggregates. The catalog also specifies a *limit type* for each limit. The limit type determines the error condition that results when a limit is reached or exceeded.
- The number of dimension values for which drag-and-drop sorting is allowed in the Selector. If a user exceeds the limit, drag-and-drop sorting will not appear in the Sort Selection dialog box as a sort method in the *Based On* box.
- The number of selections allowed using the Family tool in the Selector. In the Select by Family dialog box, if the user attempts to exceed the limit, an error message prompts the user to select fewer dimension values before running the Family tool.

Default and application limits

The following table provides a summary of default and application limits.

Value	Default Limit	Application Limit
Reports		
Across	250 data cells	5000 data cells
Down	250 data cells	5000 data cells
Page	250 data cells	5000 data cells
Graphs		
Across	250 data cells	500 data cells (pie graphs: 10 data cells)
Legend	250 data cells	500 data cells (pie graphs: 100 data cells)
Page	250 data cells	5000 data cells
Custom Aggregates	100 members	1200 members
Drag-and-Drop Sorting	1500 values	99,999 values
Family Tool	100 values	1000 values

Important: As the number of dimension values selected approaches the application limit, a significant degradation of performance may occur. For example, a report with 5000 dimension values across and 5000 values down will produce 25,000,000 data values. Such a report may put your system at its limits and may not run to completion. Users should not exceed the application limit (for example, 5000 data cells per dimension in a report).

How limits are applied on the Web

Dimension value limits that you set for report, graphs, and the Family Tool do not apply on the Web. Limits that you set for custom aggregate members do apply.

Limit types

The following table describes the three limit types, which determine the error condition that will result when a user reaches or exceeds a limit for a report, graph, or custom aggregate.

Limit Type	Description
NO LIMIT	Causes Analyzer to ignore the limit set by the DBA. The maximum limit is then the application limit. Note: Oracle Corporation recommends that you do <i>not</i> use this setting.
WARNING	Displays a message indicating that the limit set by the DBA has been reached or exceeded. A confirmation box is displayed; the user must choose Yes to continue or No to return to the Selector dialog box. Note: This is the default limit type.
ERROR	Displays a message indicating that the maximum value has been reached or exceeded. Prevents users from continuing until a smaller number of dimension values or custom aggregate members is selected. Note: This limit type also applies automatically if a user attempts to reach or exceed the application limit.

Setting limits and limit types

You can set value limits and limit types for a Windows client by choosing **Governors** from the Settings menu in Database Manager [Analyzer]. The Governors dialog box contains the following tabs:

- **Report tab** — Allows you to set limits and limit types for the Across, Down, and Page dimensions in a report created in a Windows client.
- **Graph tab** — Allows you to set limits and limit types for the Across, Legend, and Page dimensions in a graph created in a Windows client.
- **Custom Aggregates tab** — Allows you to set limits and limit types for members in a custom aggregate created in a Windows client.
- **Selector tab** — Allows you to set limits on the number of dimension values that are allowed when using drag-and-drop sorting and the Family tool in the Selector.

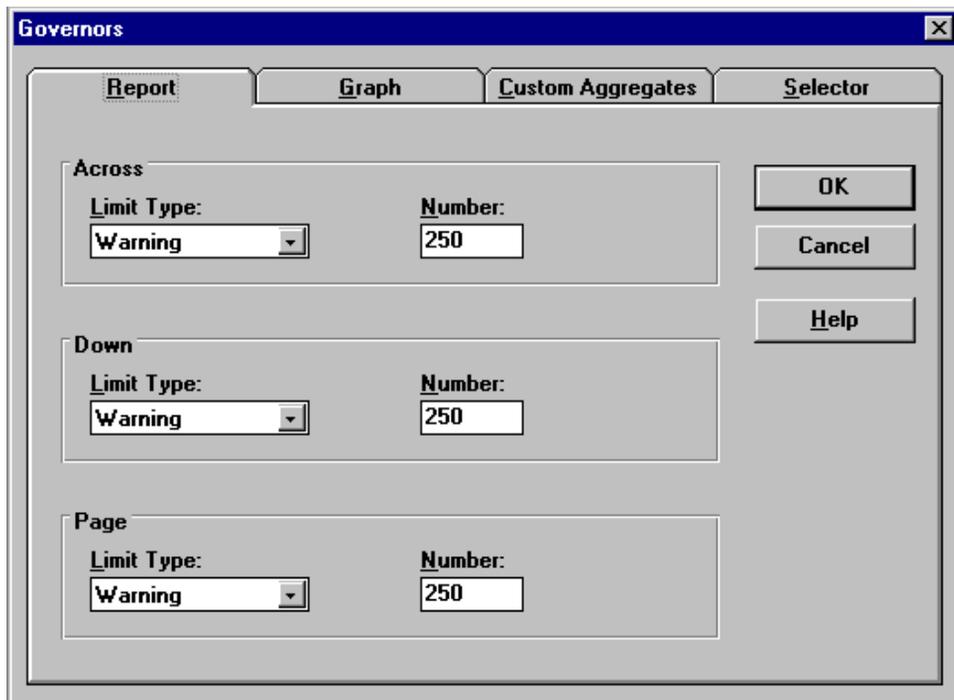
Procedure: Setting limits and limit types

To set value limits and limit types, perform the following steps:

1. From the Database Manager [Analyzer] Settings menu, choose **Governors**.
Result: The Governors dialog box is displayed
2. Choose the appropriate tab or tabs and make the desired changes.
3. When your work is complete, choose **OK**.

Example: Governors dialog box

The following example shows the Governors dialog box with the default limits and limit types for reports.



Related information

For more information about setting value limits, search for the following topics in the Database Manager [Analyzer] Help system:

- “Setting Data Cell Limits for Graphs”
- “Setting Data Cell Limits for Reports”
- “Setting Member Limits for Custom Aggregates”
- “Setting Limits for Actions in the Selector”

Specifying Default Documents

About default documents

To create a new report or graph in the Analyzer user interface, a user chooses **New**. If the user also chooses the **Use Default** option, Analyzer displays a default report or graph as a template for the new document. The default report or graph includes a specified layout, dimension selections, and formatting, which the user can then modify for the new document.

System defaults for documents

Analyzer uses a system default for reports and graphs, which displays the first three values from each dimension in the database. However, you can replace the system default by specifying a new default that displays the dimension values of your choice.

Considerations for default documents and relational data

If users access data stored in an RDBMS, you should specify default documents that include dimension values that are stored in the database, rather than summarized at runtime. Specifying a default report or graph that uses stored values will reduce the time required to display the document.

Procedure: Specifying a default document

To specify a default report or graph, perform the following steps:

1. From the Analyzer File menu, choose **New**.
2. Create the report, or graph that you wish to use as the new default.
3. When your work is complete, from the File menu, choose **Save As**.
Result: The Save As dialog box is displayed.
4. In the Save As dialog box, select the **Use as Default** check box, and then choose **OK**.
5. To provide the new default report to users, copy it to the MASTER library or the public library, as described in [Chapter 7](#).

Related information

For more information about specifying a default report and graph, search for the following topics in the Analyzer Help system:

“Specifying Default Reports”

“Specifying Default Graphs”

Customizing the Windows Client Interface

Chapter summary

This chapter describes how you can modify the Analyzer Windows client interface to suit the needs of your users.

List of topics

This chapter includes the following topics:

- [Customizing Menus and the Toolbar](#)
- [Managing Customizations](#)
- [Installing Custom Help](#)
- [Installing a Custom Application](#)
- [Using Languages Other Than English](#)

Customizing Menus and the Toolbar

Types of customizations

The following table describes the types of customizations that you can make to Analyzer menus and the toolbar, using the Customize dialog box in the Analyzer Windows client.

Type of Customization	Notes
<p>Creating custom menu items and toolbar buttons to perform specified actions such as the following:</p> <ul style="list-style-type: none"> ▪ Initializing applications ▪ Executing Express commands ▪ Opening reports and graphs 	<p>You can add as many as 10 new items to the bottom of a standard menu, or you can create a custom User menu with as many as 20 items.</p> <p>When you create a custom menu item, Analyzer creates a corresponding custom toolbar button automatically. The new button is displayed in the Customize dialog box, in the Available box of the Toolbar tab. The category for the toolbar button is the same as that for the menu item.</p>
<p>Adding, moving, and removing toolbar buttons, by dragging them on and off of the toolbar</p>	<p>You can define bitmaps for new buttons, or use default bitmaps; Analyzer automatically adds these to the Available Buttons box in the Customize dialog box.</p>
<p>Hiding menus and menu items that you do not wish to make available to users</p> <p>Redisplaying a menu or menu item that you have hidden previously</p>	<p>The following menus or menu items cannot be hidden:</p> <ul style="list-style-type: none"> ▪ File menu ▪ File/Open ▪ File/Close ▪ File/Exit ▪ Help menu ▪ Help/About Analyzer
<p>Deleting menu items and toolbar buttons that you have created</p>	<p>When you delete a custom menu item, Analyzer automatically removes the associated button from the toolbar or from the Available box of the Toolbar tab in the Customize dialog box.</p> <p>You cannot delete menu items and toolbar buttons that issue system commands.</p>

Type of Customization	Notes
Editing the captions for menu items or the ToolTips	The caption appears when the mouse pauses over specific toolbar buttons.
Editing the long description for menu items and toolbar buttons	The description appears on the status bar when the user selects a menu item or toolbar button.

Note: When you customize menus and the toolbar, Analyzer Help topics do not reflect these changes.

The effect of hiding menus and items on the toolbar

Hiding certain menu items also removes associated buttons from the toolbar, as shown in the following table.

Menu/Menu Item Hidden	Effect on Toolbar
File/Save	Removes the New, Save, and Save As buttons from the toolbar.
File/Print	Removes the Print button from toolbar.
File/Print Setup	Removes the Print buttons from the Print Preview and Print dialog boxes.
File/Print Multiple	Removes the File/Print Multiple button automatically if File.Print=No.
File/Export to File	Removes the Export button from the toolbar.
Format/Number	Removes the Number button from the Axis Options dialog box.
Format/Alignment	None
Format/Font	Removes the Font button from the Axis Options dialog box.
Format/Patterns	Removes the Patterns button from the Axis Options dialog box.
Format/Page Setup	Removes buttons from Print Preview and Print dialog boxes.
Tools Menu	None
Tools/Selector	Removes the Selector button from the toolbar and disables access to the Selector when double-clicking on tiles.

Procedure: Customizing menus and the toolbar

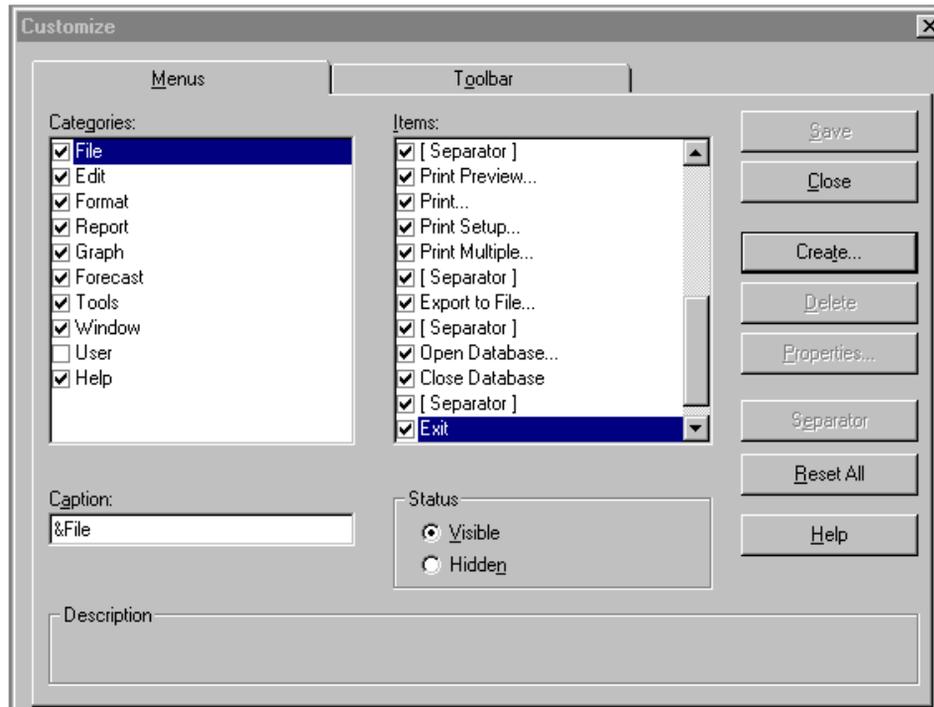
To customize menus and the toolbar, perform the following steps:

1. From the Analyzer Tools menu, choose **Customize**.
Result: The Customize dialog box is displayed, with tabs for customizing menus and the toolbar.
2. Choose the Menus tab or the Toolbar tab. Make the desired customizations.
3. Choose **Save** to save your work on both tabs for that session.
4. Choose **Close** to leave the Customize dialog box.

Example: Customize dialog box with Menus tab

The following example shows the Menus tab with the **File** menu selected. In the example, the check box next to all of these items are checked. This indicates that all of the items on the **File** menu will be displayed. (The Status frame also indicates

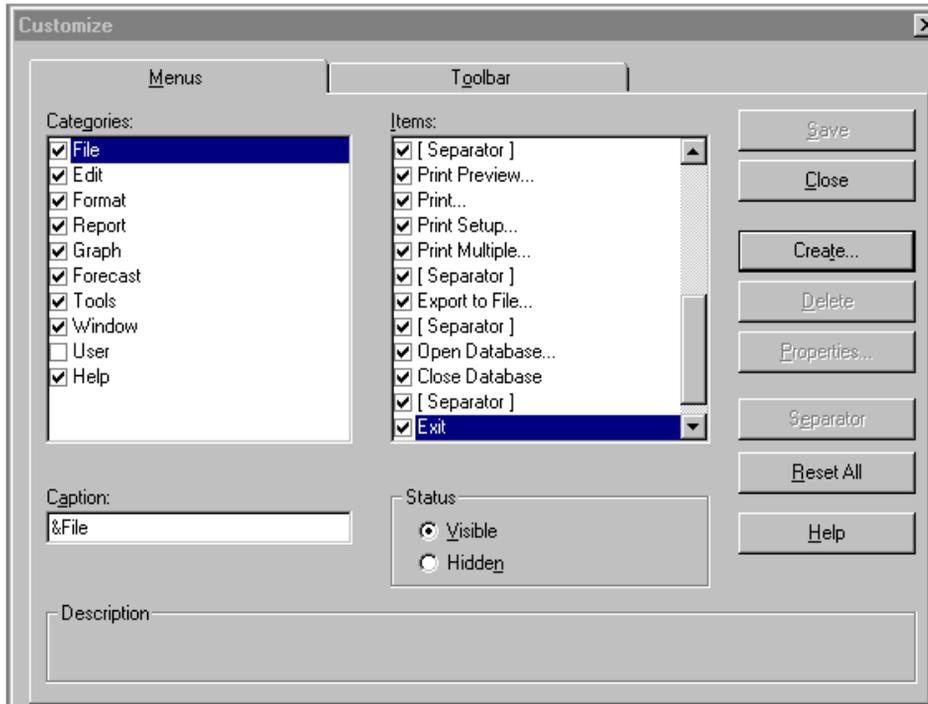
that the selected items will be visible.) Four [Separator] menu items will be displayed as solid separator bars on the File menu.



Example: Customize dialog box with Toolbar tab

The following example shows the Toolbar tab with the **File** menu selected. In this example, a bitmap picture for each available button is shown in the Available

Buttons box; an available button can be dragged from this box directly to the toolbar.



Related information

For detailed information about customizing menus and the toolbar, search for the following topics in the Analyzer Help system:

- “Adding and Removing Toolbar Buttons”
- “Changing Toolbar Button Functions”
- “Creating Menu Commands”
- “Creating Toolbar Buttons”
- “Customizing the Analyzer Toolbar”
- “Deleting Toolbar Buttons”
- “Deleting User-Defined Commands”
- “Hiding Menus and Menu Commands”
- “Modifying Toolbar Button Descriptions”
- “Renaming Toolbar Buttons”

Managing Customizations

Storing and distributing customizations

By default, any customizations that you make to menus and the toolbar are stored in a file called ANALYZER.CST. This file is located in your Analyzer user directory. If you wish to distribute customizations to users on their PCs, you can distribute the file ANALYZER.CST.

Resetting customizations

If you decide that you want to discard all of the customizations that you have made to menus and the toolbar, you can reset Analyzer to the default states that existed at the time of installation.

Before you reset the customizations that you have made, it is a good idea to preserve your customizations by saving the file that stores them. (By default, this file is ANALYZER.CST.) Then, if you later decide to restore your customizations, you can copy this file back to the Analyzer installation directory.

Procedure: Resetting customizations

To reset customizations to installation defaults, perform the following steps:

1. From the Analyzer Tools menu, choose **Customize**.

Result: The Customize dialog box is displayed, with tabs for customizing menus and the toolbar.

2. Choose **Reset All**.

Result: Analyzer displays a confirmation box reminding you that resetting will discard all customizations that you have accumulated.

3. Choose **Yes** in the confirmation box.

4. Choose **Close** to leave the Customize dialog box.

Redisplaying the Customize dialog box

If you hide the Tools menu or the **Customize** menu item, you will be unable to access the Customize dialog box again in the event that you wish to redisplay either item.

You can redisplay the Tools menu and the **Customize** command by changing a setting in the Options section of the ANALYZER.INI file, as follows:

```
ShowCustomize=TRUE.
```

Related information

For more information about storing customizations, search for the following topics in the Analyzer Help system:

“Saving Your Customizations”

“Resetting Menus and Menu Commands”

Installing Custom Help

Providing a custom Help system

You can provide a custom Help system that contains topics of particular interest to your user community. For example, you can provide information about your data or about customizations that you have made to Analyzer. Users can access these topics from the Analyzer Help menu or from another menu or toolbar button.

Each custom Help system must be contained in a separate file with an .HLP extension. You must create the file using a facility such as Microsoft Help Tools or RoboHELP.

Methods for installing a custom Help system

You can install a custom Help system by using either of the two methods described in the following table.

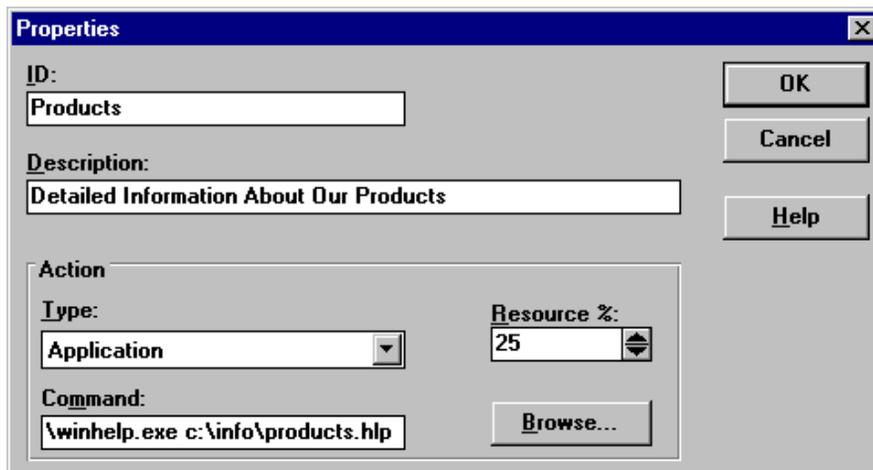
Method	Notes
Using the Customize dialog box in Analyzer	<p>With this method, you can install a Help system for use with Analyzer only.</p> <p>You can set up the system to access the custom Help from the Analyzer menu or from a toolbar button of your choice.</p>
Using a tool such as Windows Notepad to modify the .INI file for the application	<p>With this method, you can install a Help system for use with Analyzer or with any Analyzer component or module (e.g., Application Manager).</p> <p>You can set up the system access the custom Help from the application Help menu only. Custom Help menu items are listed below the standard Help menu item, in the order in which they are entered in the .INI file.</p>

Procedure: Installing custom Help using the Customize dialog box

To install a custom Help system using the Customize dialog box, create a custom menu item or toolbar button that initializes the WINHELP executable and your custom Help (.HLP) file. (See [“Customizing Menus and the Toolbar”](#) on page 13-2.)

Example: Installing custom Help using the Customize dialog box

In the following example, a custom Help system called “Products” will be accessible from a custom menu item or toolbar button.



Procedure: Installing custom Help by modifying an .INI file

To install a custom Help system by modifying an .INI file, perform the following steps:

1. If a section for custom Help does not already exist in the .INI file for the application, add one. Use the following label: [Custom Help]
2. Add a line for each Help file. Use the following format, where *help item* is the name of the item that will appear on the application Help menu, and *path and filename* are the full DOS path and file name for the Help file.

Example: `help item = path\filename`

Note: You can precede the appropriate letter in your Help file name with an ampersand (&) to designate a shortcut key.

Example: Installing custom Help by modifying an .INI file

The following lines add two menu items to the Help menu: “Our Own Intro” and “Our Brands.” In the example, “Our Brands” can be accessed from the Help menu with shortcut keys (ALT, H, B).

```
[Custom Help]
Our Own Intro = G:\HELPDIR\INTRO.HLP
Our &Brands = G:\HELPDIR\BRANDS.HLP
```

Installing a Custom Application

Custom applications accessed from Analyzer

As discussed in the topic [“Customizing Menus and the Toolbar” on page 13-2](#), you can use the Customize dialog box in Analyzer to create custom menu items and toolbar buttons that will initialize applications from Analyzer.

Custom applications accessed from icons

You can also develop and install custom applications that can be accessed from their own application icons or database icons. This process involves the following steps:

1. Using a tool such as Windows Notepad to add the path of the custom application to the APPMGR.INI file.
2. Running the AP.SETUP program. This installs application information in the Applications catalog (AP.CATALOG) and the Source catalog (SOCAT), and updates the database.

Procedure: Adding the path of an application to APPMGR.INI

To add an application to APPMGR.INI, perform the following steps:

1. Add a statement to the Executables section of the APPMGR.INI file. The statement must have the name of the application (as you want users to see it) and the name of the executable, including the full path. Follow the syntax used for the application.
2. Save the file and exit.

Example: Adding the path of an application to APPMGR.INI

The following statement adds a custom application to APPMGR.INI. The executable is MYAPP.EXE, located in the C:\CUSTOM directory.

```
[Executables]
Application Manager=D:\ORAHOME\OLAP\OSAnnn\PROGRAM\
  APPMGR.EXE
Analyzer=D:\ORAHOME\OLAP\OSAnnn\PROGRAM\ANALYZER.EXE
Database Manager=D:\ORAHOME\OLAP\OSAnnn\PROGRAM\
  DEMANAGE.EXE
Custom Application=C:\CUSTOM\MYAPP.EXE
```

Procedure: Running AP.SETUP

To run AP.SETUP, perform the following steps:

1. From Database Manager, open the database.
Result: Database Manager performs a master startup.
2. From the Database Manager Help menu, choose **Express Command Window**.
Result: The Express Command Window is displayed, from which you will execute Express commands.
3. Detach the server database and reattach it first with read/write access.

Example:

```
->database detach dsaturn
->database attach dsaturn first rw
```

4. Run AP.SETUP, providing an argument for each of the following: Name of application; Is application available to users? (YES or NO); Description of application; Mainframe code database; Mainframe startup program; PC code database; PC startup program.

Note: You must provide all seven arguments. To leave a property blank, enter a null value: ' '.

Example:

```
->call ap.setup('CUSTOM APPLICATION' -
Continue>'YES' 'Custom Application' ' ' ' -
Continue>'CUSTCODE' 'CUSTSTART')
```

5. Detach the server database and reattach it last with read-only access.

Example:

```
->database detach dsaturn  
->database attach dsaturn last ro
```

6. Choose **Close** to close the Express Command Window and return to Database Manager.

Using Languages Other Than English

Translating Analyzer components

You can easily translate names, short descriptions, and long descriptions for Analyzer menu commands and toolbar buttons, as described in the topic [“Customizing Menus and the Toolbar” on page 13-2](#).

If you are using a version of Analyzer that has been translated to a language other than English, your language will be used automatically if the language is selected in the Regional Settings Properties dialog box in the Windows Control Panel.

Copying English language versions of program files

You can use a utility such as Windows File Manager to copy English language versions of some program files if the following are true:

- Application Manager and Analyzer have not been translated to your language.
- The language setting in the Regional Settings Properties dialog box in the Windows Control panel is not set to English (American) or English (International).

Procedure: Copying English language versions of program files

To copy English language versions of your program files, perform the following steps:

1. Choose **Regional Settings** from the Windows Control Panel and check the value for the language setting.
2. Copy `DSENU*. *` to `DSxxx*. *`, where `xxx` is the Windows language setting for your current Windows installation.
3. Copy `WSENU*. *` to `WSxxx*. *`.
4. Copy `DWENU*. *` to `DWxxx*. *`.
5. Copy `BTENU*. *` to `BTxxx*. *`.
6. Copy `*.ENU` to `*.xxx`.

Catalog Descriptions

Appendix summary

This appendix lists Analyzer catalogs alphabetically. Catalog properties also appear in alphabetical order within each catalog.

List of topics

This appendix includes the following topics:

- [AG.CATALOG \(Custom Aggregate Catalog\)](#)
- [AP.CATALOG \(Applications Catalog\)](#)
- [CA.CATALOG \(Custom Analysis Catalog\)](#)
- [CO.CATALOG \(Component Catalog\)](#)
- [DA.CATALOG \(Data Catalog\)](#)
- [DACAT \(Data Catalog\)](#)
- [DF.CATALOG \(Data Format Catalog\)](#)
- [DI.CATALOG \(Display Catalog\)](#)
- [DMCAT \(Dimension Catalog\)](#)
- [DV.CATALOG \(Data Cube or Data View Catalog\)](#)
- [FC.CATALOG \(Forecast Catalog\)](#)
- [FD.CATALOG \(Folder Catalog\)](#)
- [GP.CATALOG \(Group Catalog\)](#)
- [GR.CATALOG \(Graph Catalog\)](#)

- INCAT (Infer Catalog)
- MP.CATALOG (Map Catalog)
- MRCAT (Measure Catalog)
- MX.CATALOG (Maximum Values Catalog)
- OP.CATALOG (User Option Catalog)
- SC.CATALOG (Slice Catalog)
- SG.CATALOG (Segment Catalog)
- SL.CATALOG (Saved Selection Catalog)
- SO.CATALOG (Source Catalog)
- SOCAT (Source Catalog)
- SP.CATALOG (Scope Catalog)
- TB.CATALOG (Report or Table Catalog)
- UR.CATALOG (User Catalog)

AG.CATALOG (Custom Aggregate Catalog)

Description

Defines each custom aggregate. Populated when a user defines a custom aggregate. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.AG.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

AG.ENTRY contains an entry for each custom aggregate.

Properties

AG.PROP contains the following properties.

Property	Description
CATEGORY	Database to which this aggregate belongs (e.g., SATURN)
DIMENSION	Dimension of the aggregate's members
DIMVAL	Name of the aggregate in the base dimension
ENGLISH	Description displayed to users when selecting a custom aggregate
IMEMBERS	Integers representing dimension values, listed one per line, which are used to calculate the aggregate; if one or more values will be subtracted, a + or - sign belongs before each member
MAKEPERAGO	YES causes prior period custom aggregates to be created
MAKEYRAGO	YES causes year-ago custom aggregates to be created
MEMBERS	Dimension values, listed one per line, used to calculate the aggregate; if one or more values will be subtracted, a + or - sign belongs before each member
MODIFIER	ID of the person who last modified this entry
NAME.LONG	Long name of the aggregate (up to 70 characters)
NAME.MEDIUM	Medium-length name of the aggregate (up to 30 characters)
NAME.SHORT	Short name of the aggregate (up to 16 characters)
NUMDAYS	Number of days in a time aggregate
NUMPOS	Number of positive members in the aggregate
NUMWEEKS	Number of weeks in a time aggregate
PERAGO	Name of the entry in AG.ENTRY that can represent the prior period
PERMANENT	YES saves the aggregate in the server database
QDRS	Personal Express commands, listed one per line, to set the aggregate's qualified data references and variables
SCRIPT	Personal Express commands, listed one per line, to set the status of the dimension
TIME.CREATED	Date and time this entry was created

Property	Description
TIME.MODIFIED	Date and time this entry was last modified
TYPE	SYSTEM or USER; if SYSTEM, the aggregate was created by Analyzer to infer year-ago or period-ago time periods; if USER, it was created by a user
VALID	YES indicates the aggregate was validated successfully
VIEW	Name of the catalog database
YRAGO	Name of the entry in AG.ENTRY that can represent the year-ago period
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

AP.CATALOG (Applications Catalog)

Description

Stores information that Analyzer needs to start up an application. (For the application to be available, the AVAILABLE property must be set to YES.) Populated by Analyzer when building or updating the catalog database.

Location

Catalog database.

Entries

AP.ENTRY contains an entry for each ANALYZER application.

Properties

AP.PROP contains the following properties.

Property	Description
AVAILABLE	YES indicates the application is available
DESCRIPTION	Description of the application
INITIALIZED	YES indicates the application has been initialized
MF.CODEDB	Code database(s) on host platform, listed one per line
MF.STARTUP	Host platform startup program
MODIFIER	ID of the person who last modified this entry
PC.CODEDB	Code database(s) on PC, listed one per line
PC.STARTUP	PC startup program
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

CA.CATALOG (Custom Analysis Catalog)

Description

Required to support compatibility with previous releases of Analyzer.

Location

Public and personal databases. The catalog in the public database is called PB.CA.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

CO.CATALOG (Component Catalog)

Description

Describes the component dimensions for reports and graphs. Populated by Analyzer when a user defines a report or graph. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.CO.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

CO.ENTRY contains an entry for each component.

Properties

CO.PROP contains the following properties.

Property	Description
EXCEPT.COUNT	The number of exceptions applied
EXCEPT.TEXT	Exceptions that apply to the entire object
FORMAT	Default formatting properties for the object
LABELVAR	Integer identifying the type of labeling to use on a component (1 for long labels, 2 for short labels, 3 for other)
LOCATION	Number identifying the tile location in the display
MODIFIER	ID of the person who last modified this entry
NESTING	YES allows nesting of this component
OBJECTS	Object types this component is eligible for (such as TABLE or GRAPH)
SELECTION	Reference to the Saved Selection catalog (SL.CATALOG)
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

DA.CATALOG (Data Catalog)

Description

Defines all measures (see also DACAT). Populated by Analyzer when building or updating the server database.

Location

Public and personal databases. The catalog in the public database is called PB.DA.CATALOG. The personal database version of the catalog contains information from the Data catalog in the public database as well as user-specific information. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

DA.ENTRY contains an entry for each measure.

Properties

DA.PROP contains the following properties.

Property	Description
ATTR.TYPE	Attribute type
BASE	Three-letter code for the base measure used to infer this measure
BASEDON	Object names of all base measures used to infer this measure
CUM.FORMULA	Formula for measures that are non-additive (e.g., percent or share)
CUSTOM.ADDLCALC	Custom additional calculations
CUSTOM.FIELD1	First data item for creating a custom measure
CUSTOM.FIELD2	Second data item for creating a custom measure
CUSTOM.SUBTYPE	Custom measure subtype
CUSTOM.TYPE	Custom measure type
D1.AGG.METHOD	Aggregation method for dimension 1
D1.DIM.ID	Name of dimension 1

Property	Description
D1.MCALC	Aggregation method for dimension 1, derived from D1.MCALC.SYS
D1.MCALC.DBA	DBA's aggregation method for dimension 1, which overrides D1.MCALC.SYS
D1.MCALC.SYS	Default aggregation method for dimension 1
D2.AGG.METHOD	Aggregation method for dimension 2
D2.DIM.ID	Name of dimension 2
D2.MCALC	Aggregation method for dimension 2, derived from D2.MCALC.SYS
D2.MCALC.DBA	DBA's aggregation method for dimension 2, which overrides D2.MCALC.SYS
D2.MCALC.SYS	Default aggregation method for dimension 2
D3.AGG.METHOD	Aggregation method for dimension 3
D3.DIM.ID	Name of dimension 3
D3.MCALC	Aggregation method for dimension 3, derived from D3.MCALC.SYS
D3.MCALC.DBA	DBA's aggregation method for dimension 3, which overrides D3.MCALC.SYS
D3.MCALC.SYS	Default aggregation method for dimension 3
D4.AGG.METHOD	Aggregation method for dimension 4
D4.DIM.ID	Name of dimension 4
D4.MCALC	Aggregation method for dimension 4, derived from D4.MCALC.SYS
D4.MCALC.DBA	DBA's aggregation method for dimension 4, which overrides D4.MCALC.SYS
D4.MCALC.SYS	Default aggregation method for dimension 4
D5.AGG.METHOD	Aggregation method for dimension 5
D5.DIM.ID	Name of dimension 5
D5.MCALC	Aggregation method for dimension 5, derived from D5.MCALC.SYS
D5.MCALC.DBA	DBA's aggregation method for dimension 5, which overrides D5.MCALC.SYS

Property	Description
D5.MCALC.SYS	Default aggregation method for dimension 5
D6.AGG.METHOD	Aggregation method for dimension 6
D6.DIM.ID	Name of dimension 6
D6.MCALC	Aggregation method for dimension 6, derived from D6.MCALC.SYS
D6.MCALC.DBA	DBA's aggregation method for dimension 6, which overrides D6.MCALC.SYS
D6.MCALC.SYS	Default aggregation method for dimension 6
D7.AGG.METHOD	Aggregation method for dimension 7
D7.DIM.ID	Name of dimension 7
D7.MCALC	Aggregation method for dimension 7, derived from D7.MCALC.SYS
D7.MCALC.DBA	DBA's aggregation method for dimension 7, which overrides D7.MCALC.SYS
D7.MCALC.SYS	Default aggregation method for dimension 7
D8.AGG.METHOD	Aggregation method for dimension 8
D8.DIM.ID	Name of dimension 8
D8.MCALC	Aggregation method for dimension 8, derived from D8.MCALC.SYS
D8.MCALC.DBA	DBA's aggregation method for dimension 8, which overrides D8.MCALC.SYS
D8.MCALC.SYS	Default aggregation method for dimension 8
D9.AGG.METHOD	Aggregation method for dimension 9
D9.DIM.ID	Name of dimension 9
D9.MCALC	Aggregation method for dimension 9, derived from D9.MCALC.SYS
D9.MCALC.DBA	DBA's aggregation method for dimension 9, which overrides D9.MCALC.SYS
D9.MCALC.SYS	Default aggregation method for dimension 9
D10.AGG.METHOD	Aggregation method for dimension 10
D10.DIM.ID	Name of dimension 10

Property	Description
D10.MCALC	Aggregation method for dimension 10, derived from D10.MCALC.SYS
D10.MCALC.DBA	DBA's aggregation method for dimension 10, which overrides D10.MCALC.SYS
D10.MCALC.SYS	Default aggregation method for dimension 10
DATATYPE	Express datatype of measure
DECIMAL	Number of decimal places to display
ENGLISH	Description displayed to users when selecting a measure
EQUATION	Formula for an inferred or custom measure
FORMAT	Display format
HIDE	YES prevents users from seeing the measure in choice lists
ISBY	Base dimensions listed one per line
MODIFIER	ID of the person who last modified this entry
MVAVG.FORMULA	Moving averages formula for measures that are non-additive (e.g., percent or share)
MVTOT.FORMULA	Moving totals formula for measures that are non-additive (e.g., percent or share)
NAME.LONG	Long name of the measure (up to 70 characters)
NAME.MEDIUM	Medium-length name of the measure (up to 30 characters)
NAME.SHORT	Short name of the measure (up to 16 characters)
PERMANENT	YES writes a custom measure to the server database
SOURCE	Name of server database in which the measure resides
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
TIME.SUMMARY.FORMULA	Formula used in the creation of certain custom measures
TYPE	Type of measure, either BASE, DERIVED, CUSTOM, or USER
VALID	YES indicates if the measure was successfully validated
VIEW	Name of the catalog database
WIDTH	Number of characters to use when printing this measure

Property	Description
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

DACAT (Data Catalog)

Description

Defines all base and custom measures. (See also DA.CATALOG.) Populated by Analyzer when building or updating the server database.

Location

Server database.

Entries

DAENTRY contains an entry for each measure.

Properties

DAPROP contains the following properties.

Property	Description
ABBREV	Three-letter code for the measure
ATTR.TYPE	Attribute type
BASE	Three-letter code for the measure used to infer this measure
BASEDON	Object names of all base measures used to infer this measure
CUM.FORMULA	Formula for measures that are non-additive (e.g., percent or share)
D1.AGG.METHOD	Aggregation method for dimension 1
D1.DIM.ID	Name of dimension 1

Property	Description
D1.MCALC	Aggregation method for dimension 1, derived from D1.MCALC.SYS
D1.MCALC.DBA	DBA's aggregation method for dimension 1, which overrides D1.MCALC.SYS
D1.MCALC.SYS	Default aggregation method for dimension 1
D2.AGG.METHOD	Aggregation method for dimension 2
D2.DIM.ID	Name of dimension 2
D2.MCALC	Aggregation method for dimension 2, derived from D2.MCALC.SYS
D2.MCALC.DBA	DBA's aggregation method for dimension 2, which overrides D2.MCALC.SYS
D2.MCALC.SYS	Default aggregation method for dimension 2
D3.AGG.METHOD	Aggregation method for dimension 3
D3.DIM.ID	Name of dimension 3
D3.MCALC	Aggregation method for dimension 3, derived from D3.MCALC.SYS
D3.MCALC.DBA	DBA's aggregation method for dimension 3, which overrides D3.MCALC.SYS
D3.MCALC.SYS	Default aggregation method for dimension 3
D4.AGG.METHOD	Aggregation method for dimension 4
D4.DIM.ID	Name of dimension 4
D4.MCALC	Aggregation method for dimension 4, derived from D4.MCALC.SYS
D4.MCALC.DBA	DBA's aggregation method for dimension 4, which overrides D4.MCALC.SYS
D4.MCALC.SYS	Default aggregation method for dimension 4
D5.AGG.METHOD	Aggregation method for dimension 5
D5.DIM.ID	Name of dimension 5
D5.MCALC	Aggregation method for dimension 5, derived from D5.MCALC.SYS
D5.MCALC.DBA	DBA's aggregation method for dimension 5, which overrides D5.MCALC.SYS

Property	Description
D5.MCALC.SYS	Default aggregation method for dimension 5
D6.AGG.METHOD	Aggregation method for dimension 6
D6.DIM.ID	Name of dimension 6
D6.MCALC	Aggregation method for dimension 6, derived from D6.MCALC.SYS
D6.MCALC.DBA	DBA's aggregation method for dimension 6, which overrides D6.MCALC.SYS
D6.MCALC.SYS	Default aggregation method for dimension 6
D7.AGG.METHOD	Aggregation method for dimension 7
D7.DIM.ID	Name of dimension 7
D7.MCALC	Aggregation method for dimension 7, derived from D7.MCALC.SYS
D7.MCALC.DBA	DBA's aggregation method for dimension 7, which overrides D7.MCALC.SYS
D7.MCALC.SYS	Default aggregation method for dimension 7
D8.AGG.METHOD	Aggregation method for dimension 8
D8.DIM.ID	Name of dimension 8
D8.MCALC	Aggregation method for dimension 8, derived from D8.MCALC.SYS
D8.MCALC.DBA	DBA's aggregation method for dimension 8, which overrides D8.MCALC.SYS
D8.MCALC.SYS	Default aggregation method for dimension 8
D9.AGG.METHOD	Aggregation method for dimension 9
D9.DIM.ID	Name of dimension 9
D9.MCALC	Aggregation method for dimension 9, derived from D9.MCALC.SYS
D9.MCALC.DBA	DBA's aggregation method for dimension 9, which overrides D9.MCALC.SYS
D9.MCALC.SYS	Default aggregation method for dimension 9
D10.AGG.METHOD	Aggregation method for dimension 10
D10.DIM.ID	Name of dimension 10

Property	Description
D10.MCALC	Aggregation method for dimension 10, derived from D10.MCALC.SYS
D10.MCALC.DBA	DBA's aggregation method for dimension 10, which overrides D10.MCALC.SYS
D10.MCALC.SYS	Default aggregation method for dimension 10
DATATYPE	Type of data (e.g., INT or DECIMAL)
DECIMAL	Number of decimal places displayed in reports
ENGLISH	Description displayed to users when selecting a measure
EQUATION	Formula for an inferred or custom measure
FORMAT	Format used to control the way values of the measure are displayed
HIDE	YES prevents users from seeing the measure in choice lists
INFER.CP	YES defines a change measure for absolute change from the prior period
INFER.CY	YES defines a change measure for absolute change from the year-ago period
INFER.PP	YES defines a change measure for percentage change from the prior period
INFER.PY	YES defines a change measure for percentage change from the year-ago period
ISBY	Base dimensions listed one per line
MODIFIER	ID of the person who last modified this entry
MVAVG.FORMULA	Moving averages formula for measures that are non-additive (e.g., percent or share)
MVTOT.FORMULA	Moving totals formula for measures that are non-additive (e.g., percent or share)
NAME.LONG	Long name for the measure (up to 70 characters)
NAME.MEDIUM	Medium-length name for the measure (up to 30 characters)
NAME.SHORT	Short name for the measure (up to 16 characters)
SOURCE	Name of server database in which the measure resides
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

Property	Description
TIME.SUMMARY.FORMULA	Formula used to create some custom measures
TYPE	Type of measure, either BASE, DERIVED, CUSTOM, or USER
VIEW	Name of the catalog database
WIDTH	Width of the column in characters to use when displaying this measure

DF.CATALOG (Data Format Catalog)

Description

Describes data formats for Web reports. Populated by Analyzer when a user defines a data format for a Web report. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.DF.CATALOG.

Entries

DF.ENTRY contains an entry for each data format defined by the user.

Properties

DF.PROP contains the following properties.

Property	Description
BACKCOLOR	Background color of text.
BORDERCOLOR	Stores the colors for the bottom, top, left, and right borders in the BGR (Blue, Green, Red), the format that Web Agent uses

Property	Description
BORDERSTYLE	Stores the styles for the bottom, type, left, and right borders. Valid values are: <ul style="list-style-type: none"> ■ 0 = None ■ 1 = Single ■ 2 = Double ■ 3 = Dotted ■ 4 = Dashed
BORDERTHICKNESS	Stores the border thickness in pixels for bottom, top, left and right single borders
CONDITIONS	A tagged text block that contains the following fields in which <i>x</i> is a value from 1 to the value of NUMCONDITIONS: <ul style="list-style-type: none"> ■ COMPTYPE_{<i>x</i>} — Type of comparison: DIMENSION or VALUE ■ COMPARISON_{<i>x</i>} — Condition for the comparison (EQUALS, GREATERTHAN, LESSTHAN, BETWEEN) ■ VALUE1_{<i>x</i>} — First comparison value ■ VALUE2_{<i>x</i>} — Second comparison value ■ VALUE3_{<i>x</i>} — Third comparison value ■ MEASURE_{<i>x</i>} — Name of the comparison measure (e.g., F.SALES) ■ MARGINTYPE_{<i>x</i>} — Type of margin comparison (NONE, VALUE, PERCENT) ■ MARGINDIFF_{<i>x</i>} — Margin difference
DESCRIPTION	Descriptive name for the format
FONTNAME	Name of the font (e.g., Sans Serif, Arial, etc.).
FONTSIZE	Size of the font
FONTSTYLE	Style of the font (e.g., bold, italic, etc.)
FORECOLOR	Foreground color.
NUMCONDITIONS	Number of conditions
MEASURE	Measure with which the format is associate
ORDER	Not currently used
QUALIFICATIONS1	Stores first set of qualifying dimension values for the measure

Property	Description
QUALIFICATIONS2	When COMPTYPE=MEASURE is specified, stores an initial set of qualifying dimension values for the comparison measure
QUALIFICATIONS3	When a second condition is specified, stores second set of qualifying dimension values for the measure
QUALIFICATIONS4	When COMPTYPE=MEASURE or a second condition is specified, stores second set of qualifying dimension values for the measure
STATUS	Data formats active for the current document (Internal use only)
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
WIDTH	Number of characters to use when printing the measure
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

DI.CATALOG (Display Catalog)

Description

Describes the type of display created by a report or graph. Usually populated by Analyzer when a user creates a report or graph. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.DI.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

DI.ENTRY contains an entry for each report and graph.

Properties

DI.PROP contains the following properties.

Property	Description
CBO.CAPTION	Caption in the status bar; if blank, no status bar is displayed
CBO.CONTENTS	Loading instructions for the status bar
CBO.INDEX	Default text for the Table status bar
CMD.CAPTION	Caption for a command button; if blank, no command button will be displayed
CNTL.ENTRY	Table and graph entries (from TB.ENTRY and GR.ENTRY) to display in the window, listed one per line
CNTL.HEIGHT	Height of view
CNTL.LEFT	Left position of the child view
CNTL.TOP	Top position of the view inside the window
CNTL.TYPE	Type of view, such as TB for tables and GR for graphs
CNTL.WIDTH	Width of view
CONTENT	Contents of the display window
DESCRIPTION	Description of the display
EXPORT.SETUP	Export information
INFOBAR.SHOW	YES displays the information bar at the bottom of the window
INFOBAR.TEXT	Contents of the information bar
KEYWORD	Keyword for searching and sorting
MODIFIER	ID of the person who last changed this entry
PAGE.SETUP	Page setup information
REDISPLAY	YES turns on automatic redisplay
SWAPTILES	YES allows tiles to be swapped
TILES	YES displays tiles
TIME.CREATED	Date and time this entry was created

Property	Description
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
WIN.HEIGHT	Height of display window
WIN.LEFT	Left position within the application window
WIN.STATE	Window state, either NORMAL, MAXIMIZED, or MINIMIZED
WIN.TOP	Top position within the application window
WIN.WIDTH	Width of the display window
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

DMCAT (Dimension Catalog)

Description

Describes the dimensions in a database. Populated by Analyzer when building or updating the server database.

Location

Server database.

Entries

DMENTRY contains an entry for each base dimension (e.g., PROD, GEOG, TIME and MEAS).

Properties

DMPROP contains the following properties.

Property	Description
ABBREV	Unique one-letter character for this dimension
BLDSORT	Build sorting mechanism
DESCRIPTION	Descriptive name of the dimension
DIM.WIDTH	Maximum width, in characters, for dimension values (must be at least 12 characters); if NA, no width is set
EXTEND	Size of subsequent blocks of dimension values to allocation after initial allocation is exhausted
EXTENDINIT	Size of initial block of dimension values to allocate
GENSVN	YES automatically generates unique dimension value names.
INHERIT	YES indicates that attributes will be determined for dimension aggregates
KEY.WIDTH	Maximum width in characters for UKEY values
LVLORD	Sort order of levels, either ASCENDING or DESCENDING
NAME	Object name of the dimension
OFFSET	Reserved
PLURAL.DESC	Plural description of dimension
ROLLING	YES rolls dimension values during an update
SORTTYPE	Final sorting mechanism: HIERARCHY, LEVEL, etc.
START.SELECTON	Default status for this dimension at startup
TIME.CREATED	Date and time entry was created
TIME.MODIFIED	Date and time entry was last modified
TYPE	Dimension type: TIME, GEOG, PROD, MEAS, OTHR, or CNJT

DV.CATALOG (Data Cube or Data View Catalog)

Description

Stores dimension status used to select data for scopes and slices. Populated when a user defines a data cube through Analyzer. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.DV.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

DV.ENTRY contains an entry for each data cube.

Properties

DV.PROP contains the following properties.

Property	Description
DATAVIEW.SIZE	Estimated size of a data cube in megabytes
DESCRIPTION	Description of the data cube
MODIFIER	ID of the person who last modified this entry
PARTNERS	Not currently used
PUBFLAG	Identifies the data cube as an Information Resources, Inc. Publishing dataview definition
SELECTIONS	LIMIT commands to set status for each base dimension
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs

Property	Description
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	MF platform

FC.CATALOG (Forecast Catalog)

Description

Stores specific information about each defined forecast. It is populated when a user saves a forecast. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.FC.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

FC.ENTRY contains an entry for each folder.

Properties

FC.CATALOG contains the following properties.

Property	Description
COMMENT	User entered comment
FCMEAS	Measure containing forecasted data
INTIME1	Beginning period of supporting data.
INTIME2	Ending period of supporting data
MEASURE	Measure on which the forecast is based
METHOD	Forecast method (Trend, Holt-Winters, and so forth)
MODIFIER	The ID of the person who modified the entry last.
NAME.LONG	Long name of measure (maximum 70 characters)

Property	Description
NAME.SHORT	Short name for measure (maximum 16 characters)
OUTTIME1	Beginning period of forecast data
OUTTIME2	Ending period of forecast data
PARAM1	First optional parameter (depends on forecast method)
PARAM2	Second optional parameter (depends on forecast method)
PARAM3	Third optional parameter (depends on forecast method)
PARAM4	Fourth optional parameter (depends on forecast method)
SELECTIONS	References into the selection catalog
TIMELEVEL	Level within TIME dimension on which the forecast is based
TIME.CREATED	The date and time the entry was created.
TIME.MODIFIED	The date and time the entry was last changed
USER.DATA	User overrides to forecast data
VIEW	The name of the VIEW in which the entry was defined
::CLASS::	The class of the library to which the entry belongs
::LIBRARY::	Name of the Library to which the entry belongs
::PASSCOMP::	The property pass level of the entry
::PLATFORM::	The platform of the library to which the entry belongs

FD.CATALOG (Folder Catalog)

Description

Stores specific information about each defined folder. Populated when a user defines a folder. Initially an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.FD.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

FD.ENTRY contains an entry for each folder.

Properties

FD.PROP contains the following properties.

Property	Description
BMP	Bitmap file for folder
DESCRIPTION	Description of the folder as it will appear on user menus
ITEMLIST	List of items in the folder
MODIFIER	ID of the person who last modified this entry
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

GP.CATALOG (Group Catalog)

Description

Describes the groups of users defined for this database. You populate this catalog as you create and maintain user groups.

Location

Catalog database.

Entries

GP.ENTRY contains an entry for each group. You maintain this dimension.

Properties

GP.PROP contains the following properties.

Property	Description
ACCESS	Specifies the access rights for the group. Valid values are: <ul style="list-style-type: none"> ■ Master — Administrator group ■ Personal — Standard read/write access ■ Guest — Read only access
DBA	Not currently used
DESCRIPTION	Description of the group
LIB.RO	Libraries the group can access as read-only, listed one per line, such as MF.MASTER
LIB.RW	Libraries the group can access as read/write, listed one per line, such as MF.MASTER
MODIFIER	ID of the person who last modified this entry
PARTNER.SCOPE	Not currently used
SCOPE	Name of the entry in SP.ENTRY that defines this group's scope
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

GR.CATALOG (Graph Catalog)

Description

Stores specific information about each defined graph. Populated when a user defines a graph. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.GR.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

GR.ENTRY contains an entry for each graph.

Properties

GR.PROP contains the following properties.

Property	Description
DATA.FORMAT	Internal format of data
EXCEPT.TEXT	Exceptions applied to the entire object; not currently used
FORMAT	Default format properties for the object
FORMAT.VERSION	Version number of the format data: 1.0 original binary format (pre release 1.5) 2.0 MIME-encoded binary
LEGEND.SHOW	YES to show legend
LEGEND.TEXT	User changes to legend table
MODIFIER	ID of the person who last modified this entry
PAGECTRL.SHOW	YES shows page controls
SCRIPT.CLEANUP	Script for Analyzer to apply before closing the graph
SCRIPT.INIT	Script for Analyzer to apply before opening the graph
SCRIPT.POSTPAGE	Script for Analyzer to apply after paging
SCRIPT.PREPAGE	Script for Analyzer to apply before paging
SEGMENT.X	Reference in SG.CATALOG for the X edge
SEGMENT.Y	Reference in SG.CATALOG for the Y edge
SEGMENT.Z	Reference in SG.CATALOG for the Z edge
TEXT.VALUES	Values of standard single line text objects on the graph
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
TYPE	Number identifying the graph type
VIEW	Name of the catalog database
WEB.FORMAT	Default format properties for the object, stored as text

Property	Description
XTICKLABELS	User changes to labels on X-axis
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

INCAT (Infer Catalog)

Description

Contains definitions for inferred measures for use with Information Resources, Inc. InfoScan data. Describes the set of measures in your database that will be inferred or calculated. Populated by Analyzer when building or updating the server database.

Location

Server database.

Entries

INENTRY contains an entry for each measure that can be inferred or calculated from other measures in the database.

Properties

INPROP contains the following properties.

Property	Description
CP	YES indicates change from prior period measure to be inferred
CY	YES indicates change from year ago measure to be inferred
FORMAT	VB number format for the value of the measure
FORMULA	Equation (EQ) to calculate the inferred measure
INFERRED	YES indicates the measure should be inferred

Property	Description
NAME.LONG	Long name of the measure (up to 70 characters)
NAME.MEDIUM	Medium name of the measure (up to 30 characters)
NAME.SHORT	Short name of the measure (up to 16 characters)
PP	YES indicates percent change from prior period measure to be inferred; INFERRED must also be YES
PY	YES indicates percent change from year ago measure to be inferred; INFERRED must also be YES
REQUIRED	Pre-existing measures required to infer this measure

MP.CATALOG (Map Catalog)

Description

Reserved for future use.

Location

Public and personal databases. The catalog in the public database is called PB.MP.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

MRCAT (Measure Catalog)

Description

Describes the set of base measures defined for your database. Populated by Analyzer when building or updating the server database.

Location

Server database.

Entries

MRENTY contains an entry for each measure in the server database.

Properties

MRPROP contains the following properties.

Property	Description
ABBREV	Three-letter code for this measure
ATTR.TYPE	Not currently used
BASE	Three-letter code identifying the base measure required for this measure
CUM.FORMULA	Formula for measures that are non-additive, such as percent or share
D1.AGG.METHOD	Aggregation method for dimension 1
D1.DIM.ID	Name of dimension 1
D1.MCALC	Default aggregation method for dimension 1
D2.AGG.METHOD	Aggregation method for dimension 2
D2.DIM.ID	Name of dimension 2
D2.MCALC	Default aggregation method for dimension 2
D3.AGG.METHOD	Aggregation method for dimension 3
D3.DIM.ID	Name of dimension 3
D3.MCALC	Default aggregation method for dimension 3
D4.AGG.METHOD	Aggregation method for dimension 4
D4.DIM.ID	Name of dimension 4
D4.MCALC	Default aggregation method for dimension 4
D5.AGG.METHOD	Aggregation method for dimension 5
D5.DIM.ID	Name of dimension 5
D5.MCALC	Default aggregation method for dimension 5
D6.AGG.METHOD	Aggregation method for dimension 6
D6.DIM.ID	Name of dimension 6
D6.MCALC	Default aggregation method for dimension 6
D7.AGG.METHOD	Aggregation method for dimension 7
D7.DIM.ID	Name of dimension 7

Property	Description
D7.MCALC	Default aggregation method for dimension 7
D8.AGG.METHOD	Aggregation method for dimension 8
D8.DIM.ID	Name of dimension 8
D8.MCALC	Default aggregation method for dimension 8
D9.AGG.METHOD	Aggregation method for dimension 9
D9.DIM.ID	Name of dimension 9
D9.MCALC	Default aggregation method for dimension 9
D10.AGG.METHOD	Aggregation method for dimension 10
D10.DIM.ID	Name of dimension 10
D10.MCALC	Default aggregation method for dimension 10
DATATYPE	Data type of the formula or variable to be generated
DECIMAL	Number of decimal places displayed in reports
FIELD	Position on the tape
FORMAT	Number format for the value of the measure
MULTIPLY	Amount to multiply a source number by before adding the value to the database
MVAVG.FORMULA	Moving averages formula for measures that are non-additive, e.g., percent or share
MVTOT.FORMULA	Moving totals formula for measures that are non-additive, e.g., percent or share
NAME.LONG	Long name of the measure (up to 70 characters)
NAME.MEDIUM	Medium-length name of the measure (up to 30 characters)
NAME.SHORT	Short name of the measure (up to 16 characters)
READ.INPUT	YES indicates that the measure will be read from tape
SCREENBY	A SCREENBY clause for this measure
SOURCE	Name of server database in which this measure resides
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

Property	Description
TIME.SUMMARY.FORMULA	Formula used to create some custom measures
TYPE	Type of measure, either BASE, DERIVED, CUSTOM, or USER
WIDTH	Data width

MX.CATALOG (Maximum Values Catalog)

Description

Populated by Analyzer when building or updating the catalog database. Stores maximum limits for the following:

- Across, Down, and Page dimensions that can be included in a report
- Across, Legend, and Page dimensions that can be included in a graph
- Members that can be included in a custom aggregate
- Dimension values for which drag-and-drop sorting is allowed in the Selector
- Dimension values on which the Family tool can be run in the Selector

Location

Catalog database.

Entries

MX.ENTRY contains an entry for each limit.

Properties

MX.PROP contains the following properties.

Property	Description
AG.MAXMEM.AMOUNT	Maximum number of members allowed in a custom aggregate
AG.MAXMEM.TYPE	Type of message displayed when the user reaches or exceeds the maximum number of members allowed in a custom aggregate. Valid values are ERROR, WARNING, and NO LIMIT.

Property	Description
GR.ACROSS.AMOUNT	Maximum number of Across dimension values allowed in a graph. Valid values range from 1 to 500. The default is 250.
GR.ACROSS.TYPE	Type of message displayed when the user reaches or exceeds the maximum number of Across dimension values allowed in a graph. Valid values are ERROR, WARNING, and NO LIMIT.
GR.LEGEND.AMOUNT	Maximum number of Legend dimension values allowed in a graph. Valid values range from 1 to 500. The default is 250.
GR.LEGEND.TYPE	Type of message displayed when the user exceeds the maximum number of Legend dimension values allowed in a graph. Valid values are ERROR, WARNING, and NO LIMIT.
GR.PAGE.AMOUNT	Maximum number of Page dimension values allowed in a graph. Valid values range from 1 to 5,500. The default is 250.
GR.PAGE.TYPE	Type of message displayed when the user exceeds the maximum number of Page dimension values allowed in a graph. Valid values are ERROR, WARNING, and NO LIMIT.
MODIFIER	ID of the person who last modified this entry.
SL.FAMILY.ITEMS	Maximum number of dimension values on which the Family tool can be run in the Selector. If a user attempts to exceed the limit, an error message prompts the user to select fewer dimension values before running the Family tool. The default is 100.
SL.SORT.ITEMS	Maximum number of dimension values for which drag-and-drop sorting is allowed in the Selector. If a user exceeds the limit, drag-and-drop sorting will not appear in the Sort Selection dialog box as a sort method in the Based on box. The default value is 1500.
TB.ACROSS.AMOUNT	Maximum number of Across dimension values allowed in a report. Valid values range from 1 to 5,000. The default is 250.
TB.ACROSS.TYPE	Type of message displayed when the user reaches or exceeds the maximum number of Across dimension values allowed in a report. Valid values include ERROR, WARNING, and NO LIMIT.
TB.DOWN.AMOUNT	Maximum number of Down dimension values allowed in a report. Valid values range from 1 to 5,000. The default is 250.
TB.DOWN.TYPE	Type of message displayed when the user reaches or exceeds the maximum number of Down dimension values allowed in a report. Valid entries include ERROR, WARNING, and NO LIMIT.

Property	Description
TB.PAGE.AMOUNT	Maximum number of Page dimension values allowed in a report. Valid entries range from 1 to 5,000. The default is 250.
TB.PAGE.TYPE	Type of message displayed when the user reaches or exceeds the maximum number of Page dimension values allowed in a report. Valid entries include ERROR, WARNING, and NO LIMIT.
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

OP.CATALOG (User Option Catalog)

Description

Describes specific user options.

Location

Personal database. In thick client mode, the catalog on the remote system is replicated in the personal database on the PC.

Entries

OP.ENTRY contains an entry for each option.

Properties

OP.PROP contains the following properties.

Property	Description
DIRECTOR.POS	Position of folders in Director screen
DRILL	Indicates whether the Selector can use drilling in each dimension, listing them one per line
LVLW	Multi-line list indicating, for each dimension, how wide the levels should be displayed in the left-hand Selector box

Property	Description
SHWLVL	Multi-line list indicating, for each dimension, whether or not levels should be displayed in the left-hand Selector box
STATTOP	Multi-line list indicating, for each dimension, whether or not it should be displayed at the top-node or full list

SC.CATALOG (Slice Catalog)

Description

Describes the slices defined from this database. You populate some properties when you define a slice; Analyzer populates the others.

Location

Public and personal databases. The catalog in the public database is called PB.SC.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

SC.ENTRY contains an entry for each slice.

Properties

SC.PROP contains the following properties.

Property	Description
ADDL.CAT	Additional objects required for the catalog export file
ADDL.DATA	Additional objects required for the data export file
CAT.ENTRIES	Catalog entries that belong to this slice, listed one per line; each line has the format <code>catalog_name::entry_value</code>
CATEGORY.ID	Category ID for which the slice is defined
CATEGORY.NAME	Name of category for which the slice is defined
DATAVIEWS	Data cube that defines this slice
DESCRIPTION	Description of the slice

Property	Description
ID	Unique name of slice definition (up to 8 characters)
MODIFIER	ID of the person who last modified this entry
PUBFLAG	Identifies slice as Information Resources, Inc. Publishing slice definition
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

SG.CATALOG (Segment Catalog)

Description

Identifies the Across, Down, and Page dimensions of a report, and similar information for the various types of graphs. Populated when a user defines a report or graph. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.SG.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

SG.ENTRY contains an entry for each segment.

Properties

SG.PROP contains the following properties.

Property	Description
COMPONENTS	Entries in the component catalog, listed from outer dimension to inner dimension
EXCEPT.TEXT	Exceptions applied to the entire object; not currently used
FORMAT	Default format properties for the object
MODIFIER	ID of the person who last modified this entry
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

SL.CATALOG (Saved Selection Catalog)

Description

Defines lists, scripts, and datascripts. Populated when a user defines a saved selection. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.SL.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

SL.ENTRY contains an entry for each saved selection.

Properties

SL.PROP contains the following properties.

Property	Description
DEPEND.DIM	List of dimensions affected by this selection
DEPEND.VAR	List of variables affected by this selection
DESCRIPTION	Phrase describing this saved selection
DIMENSION	Dimension limited by this selection
HIERARCHY	Default hierarchy to display in the Selector; for drilling reports and indenting
MODIFIER	ID of the person who last modified this entry
PUBFLAG	Identifies selection as an Information Resources, Inc. Publishing selection
SEL.DESC	Description of the saved selection
SELECTION	List of dimension values or XLIMIT commands
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
TYPE	Type of saved selection: LIST or SCRIPT
VIEW	Name of the catalog database
::CLASS::	Class of the library to which this entry belongs
::LIBRARY::	Name of the library to which this entry belongs
::PASSCOMP::	Required to support previous releases of Analyzer
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC

SO.CATALOG (Source Catalog)

Description

Contains values that control the behavior of Analyzer. (See also SOCAT.) Populated by Analyzer when building or updating the server database.

Location

Catalog database.

Entries

SO.ENTRY contains an entry for each server database (e.g., DSATURN).

Properties

SO.PROP contains the following properties.

Property	Description
APPS	Applications that can run with this database
DATADB	Name of the server database
DATASOURCE	Key word to define the source of the data
DESCRIPTION	Short phrase used to describe the data source
DO.PRIOR.T	YES defines prior period custom aggregates
DO.YRAGO.T	YES defines year-ago period custom aggregates
INC.REFRESH	YES specifies incremental refreshes whenever needed.
INITIALIZED	YES indicates this database has been initialized since the last build or update
LOCATION	The location of the source, either HOST, LAN, or LOCAL
MASTERDB	Name of the catalog database
MAXVAL	Maximum number of members in a custom aggregate; this entry is superseded by AG.MAXMEM.AMOUNT in MX.CATALOG
MCALCINT	YES indicates that MCALC will use integer members list
MODIFIER	ID of the person who last modified this entry
MSGFILE	Name of file to read for messages
MSTART	Program to set status at startup
PREFIX	Two-letter prefix for object names
SCOPING	YES enables scoping
SYSMESSAGE	YES displays startup messages

Property	Description
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
USERVAL	YES enables user validation

SOCAT (Source Catalog)

Description

Contains values that control the behavior of your application. Populated by Analyzer when building or updating the server database. You can change the default values of some of these properties when defining scopes.

Location

Server database.

Entries

SOENTRY contains an entry for the server database.

Properties

SOPROP contains the following properties.

Property	Description
APPS	Applications that can run with this database
DATADB	Name of the server database
DATASOURCE	Key word to define the source of the data
DESCRIPTION	Short phrase used to describe the data source
DO.PRIOR.T	YES defines prior period custom aggregates
DO.YRAGO.T	YES defines year-ago period custom aggregates
INC.REFRESH	YES specifies incremental refreshes whenever needed
INITIALIZED	YES indicates this database has been initialized since the last build or update

Property	Description
LOCATION	The location of the server database, either HOST, LAN, or LOCAL
MASTERDB	Name of the catalog database
MAXVAL	Maximum number of members in a custom aggregate
MCALCINT	YES indicates that MCALC will use integer members lists
MODIFIER	ID of the person who last modified this entry
MSGFILE	Name of file to read for messages
MSTART	Program to set status at startup
PREFIX	Two-letter prefix for object names
SCOPING	YES enables scoping
SYSMESSAGE	YES displays startup messages
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified; after you change any catalog used for scoping, set this property to DS.TIMESTAMP to initiate a data refresh for all workstations
USERVAL	YES enables user validation.

SP.CATALOG (Scope Catalog)

Description

Describes the scopes for this database. You populate this catalog when you define scopes.

Location

Catalog database.

Entries

SP.ENTRY contains an entry for each scope.

Properties

SP.PROP contains the following properties.

Property	Description
AGENTRY	Custom aggregates in scope
DAENTRY	Custom measures in scope
DATAVIEW	Valid entry in DV.CATALOG that identifies the data cube defining this scope
DESCRIPTION	Description of the scope
MODIFIER	ID of the person who last modified this entry
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

TB.CATALOG (Report or Table Catalog)

Description

Stores information about each defined table. Populated when a user defines a report. Initially, an entry resides in the user's personal database, but it can be copied to the public database by a user with read/write access to a public library.

Location

Public and personal databases. The catalog in the public database is called PB.TB.CATALOG. In thick client mode, the catalog in the personal database on the remote system is replicated in the personal database on the PC.

Entries

TB.ENTRY contains an entry for each table.

Properties

TB.PROP contains the following properties.

Property	Description
COLLABELWIDTH	Default column width
EXCEPT.TEXT	Exceptions applied to the entire object; not currently used
FORMAT	Default format properties for the object
ICONS	Controls whether icons are displayed
INDENT	Controls whether labels are indented.
MODIFIER	ID of the person who last modified this entry.
OPTIONS	Table format options, one per line as follows: <ul style="list-style-type: none"> ▪ Performance Flags ▪ Precedence Order — Format order from the Report Options dialog ▪ Show horizontal grid (1=TRUE, 0=FALSE) ▪ Show vertical grid (1=TRUE, 0=FALSE) ▪ Indent unit (twips) — Amount to indent each level ▪ NA Spell ▪ Zero Spell (not currently used)
ROWLABELWIDTH	Default row width.
SCRIPT.CLEANUP	Script for Analyzer to apply before closing the table
SCRIPT.INIT	Script for Analyzer to apply before opening the table
SCRIPT.POSTPAGE	Script for Analyzer to apply after paging
SCRIPT.PREPAGE	Script for Analyzer to apply before paging
SEGMENT.X	Reference in SG.CATALOG for the X edge
SEGMENT.Y	Reference in SG.CATALOG for the Y edge
SEGMENT.Z	Reference in SG.CATALOG for the Z edge
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified
VIEW	Name of the catalog database
WEBCALL	Custom Express program to run instead of displaying report

Property	Description
WEBCALLWINNAME	Browser window name to display the results of WEBCALL. If blank, results will be displayed in the standard data area.
::CLASS::	Class of the library to which this entry belongs.
::LIBRARY::	Name of the library to which this entry belongs.
::PASSCOMP::	Required to support previous releases of Analyzer.
::PLATFORM::	Platform of the library to which the entry belongs, either MF or PC.

UR.CATALOG (User Catalog)

Description

Identifies users defined for this database. You populate this catalog as you add users.

Location

Catalog database.

Entries

UR.ENTRY contains an entry for each user. You maintain this dimension.

Properties

UR.PROP contains the following properties.

Property	Description
CATID	Unique, permanent ID used to generate unique catalog entry names for each user
DESCRIPTION	Descriptive information about the user (optional)
GROUP	Valid entry in GP.ENTRY that identifies the group to which the user belongs
MODIFIER	ID of the person who last modified this entry
NAME	Full name of the user

Property	Description
TEXTID	Text representing the user ID (usually the same as the value of UR.ENTRY, but will differ when numeric IDs are used, as in Oracle Applications)
TIME.CREATED	Date and time this entry was created
TIME.MODIFIED	Date and time this entry was last modified

Initialization Files

Appendix summary

This appendix describes the initialization files that are associated with Analyzer in the Windows client environment.

List of topics

This appendix includes the following topics:

- [Overview of Initialization Files](#)
- [Application Manager \(APPMGR.INI\)](#)
- [Analyzer \(ANALYZER.INI\)](#)
- [Database Manager \(DBMANAGE.INI\)](#)

Overview of Initialization Files

Initialization files associated with Analyzer

Three initialization (or .INI) files hold user- and installation-specific information about Analyzer in the Windows client environment. The following initialization files are located in your Analyzer user directory:

- APPMGR.INI — Initialization file for Application Manager
- ANALYZER.INI — Initialization file for Analyzer
- DBMANAGE.INI — Initialization file for Database Manager

Modifying initialization file settings

Much of the information in the .INI files was gathered when you install the software. However, you will probably change some of the optional settings in APPMGR.INI. Also, if you customize the user interface as described in [Chapter 11](#), you might also change settings in ANALYZER.INI.

You can modify .INI file settings using a text editor, such as Windows Notepad.

Values for initialization file settings

Values that activate and deactivate settings in initialization files can be entered in any case combination as YES or NO, TRUE or FALSE, or ON or OFF.

Application Manager (APPMGR.INI)

Overview of APPMGR.INI

APPMGR.INI is the initialization file for Application Manager. APPMGR.INI contains the following sections:

- Paths — Identifies the full path names of code database files used by Application Manager. These paths are set during software installation.
- Executables — Identifies the full path names of executable files for all installed applications. These paths are set during software installation. Also identifies and specifies the location of any custom Analyzer applications that have been installed.

- Preferences — Lists optional settings, several of which apply to database setup and startup and application startup. Some of these settings have default values that you should review as part of the installation process.
- Install — Identifies the installed Express components.
- Bitmaps — Identifies the full path name of a bitmap used to replace the standard splash screen.
- Quick Start Icons — Identifies the full path names of bitmaps for database icons.
- Communications — This section exists only when communications were specified as part of the software installation.
- Database Types — Identifies database types and their corresponding type files.
- Custom Help — Lists custom Help items as they appear on the Help menu, and identifies the files containing the Help topics.
- Recent Databases — Identifies recently accessed databases.

APPMGR.INI: Paths section

The following table describes the settings for the [Paths] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
Databases	File path	None	Location of the application's code database files.

APPMGR.INI: Executables section

The following table describes the settings for the [Executables] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
<i>Application Name</i>	Path and file name	None	Applications and their executables files that were specified during installation.

APPMGR.INI: Preferences section

The following table describes the settings for the [Preferences] section of the APPMGR.INI file.

Option Name	Value or Type	Default Value	Description
AllowCommuncations	Yes or No	Yes	If Yes, displays and allows the definition of communications setups. If No, communications setups are not available or visible and the Remote Configuration field is not available in the Edit Configuration dialog box.
AllowConfigEdit	Yes or No	Yes	If Yes, allows access to the Edit Configuration File dialog box from the Select Database Configuration dialog box. If No, the Edit Configuration File dialog box is not accessible.
CommTimeout	Number of seconds	60	Number of seconds to wait for Express server to finish startup before displaying a timeout error.
CompanyLogo	Path and file name	None	Bitmap file to use if a corporate logo will be included in the headings for Analyzer documents that are printed in presentation format.
CreateIconDefault	Yes or No	Yes	If Yes, the Create Database Icon check box in the Database Setup dialog box is checked by default when the dialog box is displayed. If No, the check box is not displayed.
DatabaseGroup	Windows program group	Oracle Sales Analyzer	Windows program group in which to place database icons.
DisableDBA	Yes or No	No	If Yes, hides application-specific DBA menu options, currently located under the Tools menu. If No, DBA options are visible.

Option Name	Value or Type	Default Value	Description
HideUID	Yes or No	No	If Yes, conceals the typed user name with X's in the dialog box that prompts for entry of the database user name. If No, displays the typed user name.
HostInfoTimeout	Number of seconds	30	Number of seconds to display Host Startup Information dialog box. If the value is zero (0), the dialog box will not be displayed.
LoginLogout	Yes or No	Yes	If Yes, allows users to enter and save login user IDs and passwords in the Database Setup dialog box. If No, prohibits users from specifying login information in the database setup and requires users to always enter login information at the time of login.
MinimizeOnUse	Yes or No	Yes	If Yes, users can reduce Application Manager to an icon. If No, users cannot reduce Application Manager to an icon.
PCXWindow	Yes or No	Yes	If Yes, shows Express Command Window menu item under the Help menu. If No, conceals the Express Command Window menu item.
PromptforUID	Yes or No	No	If Yes, prompts the user for a database user name when opening a slice database. If No, does not prompt for a user name.
UserRole	User, Developer, etc.	User	User's level. A setting of Developer activates the BADLINE option in Express.

APPMGR.INI: Install section

The following table describes the settings for the [Install] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
<i>Express product</i>	Version	Preset values; do not change	Versions of installed Express products.

APPMGR.INI: Bitmaps section

The following table describes the settings for the [Bitmaps] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
Main	Path and file name	None	Bitmap file to use for Application Manager in place of the standard splash screen logo.
Database icon	Path and file name	None	Database icon to use when the Create Database Icon check box is selected in the Database Setup dialog box.

APPMGR.INI: QuickStartIcons section

The following table describes the settings for the [QuickStartIcons] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
<i>Application Name</i>	Path and file name	None	Database icons to be used for applications specified in the Create Database Icon dialog box.

APPMGR.INI: Communications section

The following table describes the settings for the [Communications] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
ConnectDebug	Yes or No	No	If Yes, the debug parameter is added to the Connect commands during login to the remote server. If No, no log file is created.

APPMGR.INI: Database types section

The following table describes the settings for the [Database Types] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
<i>Type name</i>	File name	None	Database types and their corresponding type files.

APPMGR.INI: Custom help section

The following table describes the settings for the [Custom Help] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
<i>Help item</i>	Path and file name	None	Additional help items to include in the Application Manager Help menu that brings up custom Help files.

APPMGR.INI: Recent Databases section

The following table describes the settings for the [Recent Databases] section of the APPMGR.INI file.

Option Name	Values or Type	Default Value	Description
<i>Database name</i>	Name assigned in Database Setup	None	Recently accessed databases.

Analyzer (ANALYZER.INI)

Overview of ANALYZER.INI

ANALYZER.INI is the initialization file for Analyzer. ANALYZER.INI contains the following sections:

- Paths — Identifies the full path names of files used by Analyzer.
- Bitmaps — Identifies the location of an alternative bitmap file to be displayed while Analyzer starts up.
- Custom Colors — Identifies the colors used for various parts of the document displays.
- Options — Lists values for Analyzer options, including the General options set from the Options dialog box.
- DIRECTOR — Lists values for the options that are set from the Options dialog box (from the Tools menu, choose Analyzer Options).
- Custom Help — Lists custom Help items as they appear on the Help menu, and identifies the files containing the Help topics.
- [Database] — Identifies recently accessed databases and related reports, graphs, and forecasts. There is a Database section for each database.

ANALYZER.INI: Paths section

The following table describes the settings for the [Paths] section of the ANALYZER.INI file.

Option Name	Values or Type	Default Value	Description
Databases	File path	None	Location of the application's database files.

ANALYZER.INI: Bitmaps section

The following table describes the settings for the [Bitmaps] section of the ANALYZER.INI file.

Option Name	Values or Type	Default Value	Description
Main	Path and file name	None	Specifies a bitmap file to use in place of the standard splash screen logo.

ANALYZER.INI: CustomColors section

The following table describes the settings for the [CustomColors] section of the ANALYZER.INI file.

Option Name	Values or Type	Default Value	Description
Color 1, 2, 3 ...16			Colors used for document display

ANALYZER.INI: Options section

The following table describes the settings for the [Options] section of the ANALYZER.INI file.

Option Name	Value or Type	Default Value	Description
AllowDataException	True or False	True	If True or unspecified, allows data-based selections. If False, makes Ranking/Exception reports unavailable. Also makes Top/Bottom and Exception tools and sorting data by data value unavailable in the Selector. Data-based selection is normally available to users accessing data in an RDBMS. However, due to performance considerations, you might want to restrict users' ability to use data-based selection, thus preventing "runaway selections" that might consume excessive system resources.
ConfirmCopy	True or False	True	If True, displays a confirmation box every time the user copies an object to or from a library. The user can set this General option from the Analyzer Tools menu using the Options command.
ShowEdit	True or False	True	If True, displays the edit bar. The user can set this General option from the Analyzer Tools menu using the Options command.
ShowStatus	True or False	True	If True, displays the status bar. The user can set this General option from the Analyzer Tools menu using the Options command.
ShowTool	True or False	True	If True, displays the toolbar. The user can set this General option from the Analyzer Tools menu using the Options command.

Option Name	Value or Type	Default Value	Description
ShowToolText	True or False	True	If True, displays a balloon Tool tip description of a toolbar button when the user passes the mouse pointer over the button. A longer description is also displayed on the status bar. The user can set this General option from the Analyzer Tools menu using the Options command.
ShowCustomize	Yes or No	No	If Yes, makes the Tools menu with the Customize command visible again after this menu or menu item has been hidden in the Customize dialog box. You can use this setting to force the Customize command to be accessible from the Tools menu in the event that the user has hidden this menu or command and wishes to redisplay it.
Auto Expand	True or False	False	Not currently used.
DrillStatus	True or False	False	If True, allows users to set an option (Save Drill Status) that specifies whether to save the drilled status of dimension selections. The user can set this General option from the Analyzer Tools menu using the Options command.

ANALYZER.INI Director section

The following table describes the settings for the [Director] section of the ANALYZER.INI file.

Option Name	Value or Type	Default Value	Description
BigIcons	True or False	False	If True, Director displays larger icons for folders and for any bitmap used as a logo. If False, Director uses regular-sized icons. The user can set this Director option from the Analyzer Tools menu using the Options command.
Logo	Path and file name	None	Bitmap file to be used as a logo. The logo is displayed in Director, along with icons for document types and folders. The user can set this Director option from the Analyzer Tools menu using the Options command.
ManualMode	True or false	False	If False, Auto Arrange is enabled so that Analyzer automatically rearranges the icons in Director when the user changes the window size, adds, items, or moves items. If True, Auto Arrange is disabled.
MinOnUse	True or false	True	If True, Analyzer displays Director at startup. If False, Analyzer displays only the Analyzer window and toolbar at startup. The user can set this Director option from the Analyzer Tools menu using the Options command.

Option Name	Value or Type	Default Value	Description
ShowAtStartup	True or False	False	If True, Director reduces to an icon after the user has double-clicked on a Director icon to access a group of documents. The user can set this Director option from the Analyzer Tools menu using the Options command.
ShowSystemFolders	True or False	True	If True, Director displays icons for each category of document. If False, Director only displays icons for personal folders. The user can set this Director option from the Analyzer Tools menu using the Options command.

ANALYZER.INI Custom Help section

The following table describes the settings for the [Custom Help] section of the ANALYZER.INI file.

Option Name	Value or Type	Default Value	Description
<i>Help Item</i>	Path and file name	None	Help items for custom Help. These items are included in the Analyzer Help menu.

ANALYZER.INI: *Database* section

The following table describes the settings for the [*Database*] sections of the ANALYZER.INI file.

Option Name	Value or Type	Default Value	Description
<i>Database object</i>	Object identifier	None	Recently accessed report, graph, or forecast in this Analyzer database. Up to four objects will display.

Database Manager (DBMANAGE.INI)

Overview of DBMANAGE.INI

DBMANAGE.INI is the initialization file for Database Manager. DBMANAGER.INI contains the following sections:

- **Paths** — Identifies the full path names of files used by Database Manager. These paths are set during software installation.
- **Preferences** — Contains a setting that enables Analyzer application options to be accessed from Database Manager.
- **Custom Help** — Lists custom Help items as they appear on the Help menu, and identifies the files containing the Help topics.

DBMANAGE.INI: [Path] settings

The following table describes the settings for the [Paths] section of the DBMANAGE.INI file.

Option Name	Value or Type	Default Value	Description
Databases	File path	Program directory	Directory in which DVCAT.DB is created during the Database Manager session. You must have read/write access to this directory.
HelpFiles	File path	Program directory	Specifies the directory where the Help files for Database Manager are located.

DBMANAGE.INI: [Preference] Section

The following table describes the settings for the [Preferences] section of the DBMANAGE.INI file.

Option Name	Value or Type	Default Value	Description
DSEnable	Yes or No	Yes	If Yes, Analyzer application options can be accessed from Database Manager; if No, Analyzer application options cannot be accessed from Database Manager.

DBMANAGE.INI: [Custom Help] section

The following table describes the settings for the [Custom Help] section of the DBMANAGE.INI file.

Option Name	Value or Type	Default Value	Description
<i>Help Item</i>	Path and file name	None	Help items for custom Help. These items are included in the Database Manager Help menu.

Appendix summary

This appendix describes the RDC file for a remote Analyzer database.

List of topics

The appendix contains the following topics:

- [Overview of RDC File](#)
- [RDC File Functions](#)
- [Example: RDC File](#)

Overview of RDC File

The RDC file is the Remote Database Configuration file for an Analyzer database. Analyzer automatically creates an RDC file when you build a remote database. The RDC file has the name of the catalog database and the extension RDC (for example SATURN.RDC). The RDC file is usually located in the directory on the server where the database was built.

RDC File Functions

You reference the RDC file when you set up access to a remote database for Windows or Web clients. You can also use the RDC file to accomplish specific tasks, such as specifying a directory for users' personal databases and resetting the default directory for thin client publish to file.

For more information, see the following topics:

[“Default location for remote personal databases” on page 4-4](#)

[“Procedure: Defining a database setup” on page 5-3](#)

[“Changing the Default Directory for Thin Client Publish to File” on page 5-24](#)

[“Identifying the list of available databases” on page 8-9](#)

Example: RDC File

The following is an example of an RDC file with settings for connection to a relational manager via an ODBC driver.

```
[Paths]
code=d:\orahome\olap\oes6xx\common
data=d:\db\coffee\
export=d:\export\
personal=d:\users\

[DBInfo]
DBName=SATINT1
[Relational]
Type=ODBC
ODBC Datasource=Oracle1emmn Tables
```

Web Configuration File

Appendix summary

This appendix describes the Web configuration file (OSAWEB.CFG) for Analyzer.

List of topics

The appendix contains the following topics:

- [Overview of OSAWEB.CFG File](#)
- [OSAWEB.CFG File Sections](#)
- [Example: OSAWEB.CFG File](#)

Overview of OSAWEB.CFG File

OSAWEB.CFG is the Web Configuration file for Analyzer. The file is usually located in the directory on the server where the Web Server is installed.

OSAWEB.CFG File Sections

OSAWEB.CFG includes the following sections:

- **BANNER** — Identifies the title and image for the Analyzer banner in the Web client interface.
- **SPLASH** — Identifies the image, initialization message, and welcome message for Analyzer in the Web client interface.
- **SETTINGS** — Not currently used.
- **JAWS** — Specifies whether JAWS, a screen reader that provides verbal communication for the visually impaired, is enabled in the Web client interface.

Example: OSAWEB.CFG File

The following is an example of an OSAWEB.CFG file. In the example, the parameters for BANNER and SPLASH are each set to the default value. JAWS is not enabled.

```
[BANNER]
TITLE=(default)
IMAGE=(default)

[SPLASH]
IMAGE=(default)
INIT_MSG=(default)
WELCOME_MSG=(default)

[SETTINGS]

[JAWS]
ENABLE=NO
```

Index

A

- access mode, for groups, 7-5
- accessing data
 - using Relational Access Manager, 3-10
 - using scopes, 9-2
 - using slices, 10-2
- administrative tools
 - Analyzer, 2-14
 - Analyzer Web Administrator, 2-10
 - Application Manager, 2-10
 - Batch Manager, 2-5
 - Database Manager, 2-12
 - Express, 2-6
 - for troubleshooting, 2-15, 2-17
 - Relational Access Administrator, 2-4
- Administrator access mode, 7-3
- AG.CATALOG, 4-10, A-2
- aggregates, custom. *See* custom aggregates
- aggregation
 - data, 3-6
 - levels, 3-6
 - methods, 11-13
- AG.VALIDATE program, 9-8
- AllowConfigEdit option, 5-7
- ALLOWGAPSINCHUNK keyword, 11-21
- ALLOWGAPSINPRIOR keyword, 11-21
- Analysis Library
 - described, 8-11
 - documents, 8-12
 - setting up, 8-11
- Analyzer
 - described, 1-2
 - initialization file, B-8 to B-13
 - non-English languages, 8-18, 13-13
 - using, 2-14
- Analyzer Home page
 - accessing, 8-19
 - graphics, 8-15
 - setting the path to, 8-7
- Analyzer Web Administrator
 - Analysis library setup, 8-11
 - Analyzer Home page, 8-6, 8-15
 - identifying databases, 8-9
 - using, 2-10, 8-6
 - Welcome screen, 8-14
- ANALYZER.INI file, B-8 to B-13
- AP.CATALOG, 4-9, 13-11, A-4
- Application Manager
 - creating a slice, 10-3, 10-8 to 10-10, 10-14
 - defining a communications setup, 6-2 to 6-7
 - defining a database setup, 5-3 to 5-5
 - described, 1-2
 - initialization file, B-2 to B-8
 - Open Later option, 5-14
 - specifying login information, 5-14 to 5-19
 - updating a slice, 10-13 to 10-17
 - using, 2-11
- Applications catalog, 4-9, A-4
- applications, installing, 13-11 to 13-13
- APPMGR.INI file, 13-11, B-2 to B-8
- AP.SETUP program, 13-11
- AS portlet environment, 8-16
- attributes
 - described, 3-9
 - levels as, 3-9

B

- backing up work in Express, 2-9
- banner, customizing for the Web
 - environment, 8-16
- base measures, 4-8, 12-6, A-7
- Batch Manager
 - described, 1-3
 - using, 2-5, 3-11
- browser requirements for Web access, 8-3

C

- CA.CATALOG, 4-10, A-5
- catalog database
 - catalogs in, 4-9
 - described, 4-3
- catalogs
 - defined, 4-7
 - descriptions, A-2 to A-44
 - in catalog database, 4-9
 - in server database, 4-8
 - integrated, 4-9
 - modifying in Express, 2-8
 - prefix in public database, 4-11
- character sets. *See* translation tables, modifying
- chunking time periods, 11-20
- client mode. *See* thick client mode, thin client mode
- CO.CATALOG, 4-10, A-5
- code databases, remote, 4-4
- common module, 3-11
- communications log file, 6-4
- communications setups
 - defining, 5-4, 6-2 to 6-7
 - distributing to users, 5-3, 6-2
 - thick client mode and network connection, 6-2
 - thin client mode and RPC connection, 6-2
- Comparison documents, Analysis Library, 8-12
- Component catalog, 4-10, A-5
- COMSETUP.INI file, 5-3, 6-2
- configuration files
 - access modes, 5-10 to 5-11
 - creating, 5-6 to 5-14
 - described, 5-6
 - distributing to users, 5-3, 5-7
 - path names, 5-6
 - sample, 5-9
 - selecting, 5-4
- ConnectDebug option, 6-4
- control file, 3-11, 4-4
- copying saved objects. *See* publishing objects
- Custom Aggregate catalog, 4-10, A-2
- custom aggregates
 - described, 3-2
 - for scoped users, 7-13, 9-2, 9-12
 - publishing, 7-16
 - removing from server database, 11-11, 11-12
 - translating level names, 11-6
 - value limits for, 12-11
 - writing permanently to server database, 11-8
- Custom Analysis catalog, 4-10, A-5
- custom applications, installing. *See* applications, installing
- custom database, remote, 4-4
- custom Help system
 - installing, 13-8 to 13-11
 - listing on Help menu, B-8
- custom measures
 - for scoped users, 7-13, 9-2, 9-12
 - publishing, 7-16
 - removing from server database, 11-11, 11-12
 - translating level names, 11-6
 - writing permanently to server database, 11-8
- Customize dialog box, redisplaying, 13-7, B-11
- customizing
 - generic Web images, 8-16
 - generic Web text, 8-16
 - See also* menus, toolbar

D

- DACAT catalog, A-11
- DA.CATALOG, 4-10, A-7
- data
 - database. *See* server database
 - loader program, 1-3
 - slice. *See* slice database
 - storage in Express, 3-6

- values, *See* measure values
- warehouse, accessing. *See* Relational Access Manager
- Data catalog, 4-10, A-7, A-11
- Data Cube catalog, 4-10, A-21
- data cubes
 - copying, 9-6
 - defining, 9-3
 - described, 9-3
 - for scopes, 9-7
 - for slices, 10-5
- Data Display catalog, A-17
- Data Format catalog, A-15
- data loader program, 3-10
- database
 - configurations, described, 1-3 to 1-10
 - identifying for Web access, 8-9
 - security, setting, 7-17
- database icon, creating, 5-19 to 5-21
- Database Manager
 - defining a slice, 10-3 to 10-7
 - defining groups, 7-4
 - defining libraries, 7-14
 - defining users, 7-8
 - deleting a scope, 9-14
 - described, 1-2
 - initialization file, B-14 to B-15
 - master startup, 2-6, 2-13
 - setting value limits, 12-13 to 12-14
 - using, 2-12
- database setups
 - defining, 5-3 to 5-5
 - distributing to users, 5-3
 - procedure, 5-3 to 5-5
- database startup
 - master, 2-6
 - options, 5-19 to 5-21
- data-based selection, preventing, B-8
- DataView catalog. *See* Data Cube catalog
- DBA
 - master startup, 2-6, 2-13
 - refresh ID, 5-13
 - tasks described, 2-2
 - tools. *See* administrative tools
- DBMANAGE.INI file, B-14 to B-15

- DB.MERGE.DBANAME utility, 11-3, 12-6
- DBSETUP.INI file, 5-3
- default report and graph, 12-15
- DF.CATALOG, A-15
- DI.CATALOG, 4-10, A-17
- Dimension catalog, 4-8, A-19
- dimension values
 - in data cubes, 10-5
 - valid characters, 11-3
- dimensions
 - adding to a variable, 3-11
 - described, 3-3
 - hierarchies, 3-6 to 3-8
 - levels, 11-5
- Display catalog, 4-10, A-17
- displaying menus and menu items, 13-2
- Distribution documents, Analysis Library, 8-12
- DMCAT catalog, 4-8, A-19
- downloading slices, 10-4
- drag-and-drop sorting, 12-11
- DSC files. *See* configuration files
- DV.CATALOG, 4-10, A-21

E

- EIF files for slice, 10-10
- Enable User Validation, 7-9, 7-17, 7-18
- error messages for value limits, 12-12
- Exception documents, Analysis Library, 8-12
- EXPORT property, RDC file, 5-24
- Express
 - accessing from Database Manager, 2-13
 - backing up work, 2-9
 - master startup, 2-6
 - recovering lost work, 2-9
 - using, 2-6
- Express databases
 - described, 4-2
 - remote, 4-2
- extractor, 3-11

F

FC.CATALOG, 4-10
FD.CATALOG, 4-10, A-23
Folder catalog, 4-10, A-23
Forecast catalog, 4-10
forecasts, adding future time periods, 11-19
format statement, 12-3
formatting measure values, 12-6
future time periods, adding to database, 11-19

G

GP.CATALOG, 4-9, A-24
Graph catalog, 4-10, A-25
graphs
 specifying default, 12-15
 value limits for, 12-11
GR.CATALOG, 4-10, A-25
Group catalog, 4-9, A-24
groups
 access modes, 7-5
 creating, 7-4
 described, 7-2
 maintaining, 7-6, 7-14
Growth documents, Analysis Library, 8-13
guest access, Web, 8-2

H

Help system, custom. *See* custom Help system
HideUID option, 5-22
hiding measures, 11-11
hiding menus and menu items, 13-2
hierarchies, 3-6 to 3-8
hybrid mode, 3-10

I

icon, database. *See* database icon, creating
importing user names, 7-11
INCAT catalog, A-27
Infer catalog, A-27
inferred measures, 4-8, A-27
INI files. *See* initialization files

initialization files

 described, B-2 to B-15
 modifying, B-2
installation type, Web, 8-8
integrated catalogs, 4-9

J

Java classes, 8-8
JAWS screen reader, 8-8, 8-19, D-2
JInitiator, 8-8, 8-19

L

lag-type analysis, 11-20
LAN, sharing shell database on, 5-13
language translation, 11-6, 13-13
latest time period, 11-21
LATEST.T variable, 11-21 to 11-23
levels
 customizing names, 11-5
 described, 3-6 to 3-8
libraries
 described, 7-12
 publishing to, 7-15
 relation to scopes, 7-13
limits, *See* value limits
local Express databases, defined, 4-6
login information, 5-14 to 5-19
LoginLogout option, 5-16
long names for dimension values, 11-2

M

Map catalog, 4-11, A-28
master database. *See* catalog database
MASTER library
 publishing to, 7-15
Master public library, 7-13
master startup, 2-6, 2-13
Maximum Values catalog, 4-9, A-31
MCALC function, 11-13 to 11-16
MCALCINT property, setting, 9-7, 9-8, 10-14
Measure catalog, 4-8, A-28
Measure dimension, 3-5

measures
 See also base measures, inferred measures
 described, 3-2
 formatting values, 12-6
 hiding, 11-11
medium names for dimension values, 11-2
members lists, 11-16
menus
 customizing in Analyzer, 13-2 to 13-6
 displaying and hiding in Analyzer, 13-2
 managing customizations, 13-7
 resetting customizations, 13-7
modifying
 initialization files, B-2
 translation tables, 11-24 to 11-27
MP.CATALOG, 4-11, A-28
MRCAT catalog, 4-8, A-28
MX.CATALOG, 4-9, A-31

N

naming
 levels, 11-5
 slice files, 10-12
negative format, 12-5
network connection
 communications setup, 6-3 to 6-5
 described, 1-5, 6-2
number format, 12-3

O

OP.CATALOG, 4-11, A-33
Open Later option, 5-14
Oracle Applications
 importing user names, 7-11
 single sign-on, 7-11, 8-3
OSAWEB.CFG, 8-17, D-1

P

path names in configuration files, 5-6
PB.AG.CATALOG, 4-10, A-2
PB.CA.CATALOG, 4-10, A-5
PB.CO.CATALOG, 4-10, A-5

PB.DA.CATALOG, 4-10, A-7
PB.DF.CATALOG, A-15
PB.DI.CATALOG, 4-10, A-17
PB.DV.CATALOG, 4-10, A-21
PB.FC.CATALOG, 4-10
PB.FD.CATALOG, 4-10, A-23
PB.GR.CATALOG, 4-10, A-25
PB.MP.CATALOG, 4-11, A-28
PB.SC.CATALOG, 4-11, A-34
PB.SG.CATALOG, 4-11, A-35
PB.SL.CATALOG, 4-11, A-36
PB.TB.CATALOG, 4-11, A-41
personal database
 catalogs in, 4-11
 remote, 4-3
personal databases
 using RDC file to specifying location, 4-4
portals, implementing, 8-16
positive format, 12-4
private libraries, described, 7-13
program files, copying English language
 versions, 13-13
public database
 catalogs in, 4-11
 described, 4-3
 prefix for catalogs, 4-11
public libraries, 7-13, 7-14
publishing objects, 7-15
publishing to file, setting export directory, 5-24

Q

QDR. *See* qualified data reference
qualified data reference, described, 2-8
quick start icon. *See* database icon, creating
Quota documents, Analysis Library, 8-13

R

Ranking documents, Analysis Library, 8-13
Ranking/Exception reports, hiding, B-10
RDBMS. *See* relational databases
RDC files
 described, C-2
 EXPORT property, 5-24

- modifying for single sign-on, 8-5
 - referenced in configuration file, 5-6
 - referenced in Web database setup, 8-9
 - specifying location for user home directories, 4-4
- Read Only Web access mode, 7-4
- refresh
 - shell database, 5-12 to 5-14
 - user name for DBA, 5-13
- Relational Access Administrator
 - described, 1-3, 3-10
 - using, 2-4
- Relational Access Manager
 - configuring for single sign-on, 8-5
 - described, 1-2, 3-10
 - hybrid mode, 3-10
 - specifying default report and graph, 12-15
- relational databases
 - Web access, 8-9
 - Windows client access, 5-18
- remote databases
 - accessing, 5-1 to 5-25
 - accessing in Web environment, 8-2 to 8-19
 - described, 1-4
 - maintaining a shell database, 5-12 to 5-14
 - public database name, 4-3
- remote Express databases
 - defined, 4-3
 - listed, 4-3 to 4-4
- remote personal databases, 4-4
- Remote Procedure Call. *See* RPC connection
- Report catalog, 4-11, A-41
- reports
 - specifying default, 12-15
 - value limits for, 12-11
- Review documents, Analysis Library, 8-13
- RPC connection
 - communications setup, 6-2
 - described, 1-7, 6-2

S

- Saved Selection catalog, 4-11, A-36
- SC.CATALOG, 4-11, A-34
- SC.CREATE.DRV utility, 10-11

- Scope catalog, 4-9, A-40
- scopes
 - associating with groups, 9-12
 - associating with libraries, 9-12
 - custom aggregates, 7-13, 9-2, 9-12
 - custom measures, 7-13, 9-2, 9-12
 - deleting, 9-14
 - described, 9-2
 - modifying, 9-14
 - refreshing, 9-13
 - removing, 9-14
- screen reader, 8-8, 8-19, D-2
- security
 - setting for database, 7-17
- Segment catalog, 4-11, A-35
- Selector
 - limiting tools, B-10
 - setting value limits, 12-13
- server database, described, 4-3
- server requirements for Web access, 8-2
- SG.CATALOG, 4-11, A-35
- sharing objects. *See* publishing objects
- shell database
 - directory path on LAN, 5-6
 - maintaining, 5-12 to 5-14
 - sharing on LAN, 5-13
 - special cautions, 5-13
- short names for dimension values, 11-2
- ShowCustomize option, 13-8
- single sign-on, 8-3
- SL.CATALOG, 4-11, A-36
- Slice catalog, 4-11, A-34
- slice database
 - adding saved selections, 10-7
 - creating, 10-8 to 10-11
 - creating directly, 10-8 to 10-10
 - creating from files, 10-10 to 10-12
 - defining, 10-3 to 10-7
 - directory path on LAN, 5-6
 - distributing, 10-12
 - downloading, 10-4
 - estimating size, 10-3 to 10-4
 - identifying data, 10-5
 - setting access mode, 5-10

- updating, 10-13 to 10-17
- upgrading, 10-17 to 10-18
- slice files
 - See also* slice database
 - naming, 10-12
- SOCAT catalog, 13-11, A-39
- SO.CATALOG, 4-9, A-37
- Source catalog, 4-9, A-37, A-39
- SP.CATALOG, 4-9, A-40
- SQL database. *See* Relational Access Manager
- Standard access mode, 7-3
- startup, master, 2-6, 2-13
- system information, accessing, 2-16

T

- Table catalog, 4-11, A-41
- tables. *See* reports
- TB.CATALOG, 4-11, A-41
- thick client mode
 - communications setup, 6-2
 - comparison with thin client mode, 1-9
 - described, 1-5
 - illustrated, 1-6
 - setting access, 5-10
- thin client mode
 - communications setup, 6-2
 - comparison with thick client mode, 1-8 to 1-9
 - described, 1-7
 - illustrated, 1-8
 - publishing to a file, 5-24
 - setting access, 5-10
- Time dimension
 - defining latest time period, 11-21 to 11-23
 - fluctuations in aggregate periods, 11-20
- time periods
 - chunking, 11-20
 - future, 11-19
- time stamp, updating, 2-9
- toolbar
 - customizing in Analyzer, 13-2
 - managing customizations, 13-7
 - resetting customizations, 13-7
- Tools menu, redisplaying, 13-7, B-11
- translation tables, modifying, 11-24 to 11-27

- Trend documents, Analysis Library, 8-13
- troubleshooting, tools for, 2-15

U

- UR.CATALOG, 4-9, A-43
- User catalog, 4-9, A-43
- user groups. *See* groups
- user home directory, 4-4, 5-25
- user IDs. *See* user names
- user names
 - adding to command line, 5-23
 - hiding, 5-22
 - importing from Oracle Applications, 7-11
 - setting up, 5-22
- User Option catalog, 4-11, A-33
- user validation, 7-9, 7-17, 7-18
- users
 - creating, 7-8
 - described, 7-8
 - importing from Oracle Applications, 7-11
 - maintaining, 7-9

V

- value limits
 - error messages for, 12-12
 - setting, 12-10 to 12-14
- variables
 - described, 3-2
 - for custom names, 11-3
- version information, accessing, 2-16
- virtual directories, 8-3, 8-14, 8-15, 8-17
- voice access in Web environment, 8-19
- voice access, in Web environment, D-2

W

- warehouse. *See* Relational Access Manager
- Web access
 - browser requirements, 8-3
 - identifying databases, 8-9
 - server requirements, 8-2
- Web Analysis Library. *See* Analysis Library
- Web configuration file, 8-17, D-1

Welcome screen, customizing in Web
environment, 8-16

Windows

changing measure display attributes, 12-2
database icon, creating, 5-19 to 5-21
default values for currency, 12-5
entering user name at command line, 5-23
workstation user names. *See* user names

X

XCATimeout, 6-5