

Data Warehouse Administrator's Guide

MetaCube™ ROLAP Option

for Informix® Dynamic Server™

Version 4.0
January 1998
Part No. 000-4189

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Copyright © 1981-1998 by Informix Software, Inc. or its subsidiaries, provided that portions may be copyrighted by third parties, as set forth in documentation. All rights reserved.

The following are worldwide trademarks of Informix Software, Inc., or its subsidiaries, registered in the United States of America as indicated by “®,” and in numerous other countries worldwide:

Answers OnLine™; INFORMIX®; Informix®; Illustra™; C-ISAM®; DataBlade®; Dynamic Server™; Gateway™; NewEra™; MetaCube™

All other names or marks may be registered trademarks or trademarks of their respective owners.

RESTRICTED RIGHTS / SPECIAL LICENSE RIGHTS

Software and documentation acquired with US Government funds are provided with rights as follows: (1) if for civilian agency use, with Restricted Rights as defined in FAR 52.227-19; (2) if for Dept. of Defense use, with rights as restricted by vendor's standard license, unless superseded by negotiated vendor license as prescribed in DFAR 227.7202. Any whole or partial reproduction of software or documentation marked with this legend must reproduce the legend.

Table of Contents

Introduction

Organization of This Manual	3
Types of User	4
Documentation	4
Printed Documentation	5
Online Help	7
Readme Files	7
Related Reading	8
Compliance with Industry Standards	8
Informix Welcomes Your Comments	8

Chapter 1

Overview of the MetaCube System

What Is a MetaCube Data Warehouse?	1-3
MetaCube Software Components	1-3
MetaCube Analysis Engine	1-4
Tools for Querying the Data Warehouse	1-4
Tools for Managing the Data Warehouse	1-5
MetaCube Architecture	1-7
Client/Server on Three Tiers	1-7
Client/Server on Two Tiers	1-9

Chapter 2

Setting Up a Data Warehouse

Overview of the Process	2-3
Design a Schema	2-5
Populate the Data Warehouse Tables	2-6
Install the MetaCube Software	2-6
Generate MetaCube's Metadata Tables	2-6
Create Authorized Users	2-7
Create Metadata	2-7
Define Public Filters and Queries and Design a Folder Structure	2-8
Using Folders	2-9

	Define User Access	2-9
	Start MetaCube Scheduler	2-10
	Build Aggregate Tables	2-10
	Build Sample Tables	2-11
	Continuing Maintenance of a Data Warehouse	2-12
Chapter 3	MetaCube Scheduler	
	About MetaCube Scheduler	3-3
	Starting the Scheduler in UNIX	3-4
	Skipping the Prompts	3-4
	Checking the Status of the Scheduler	3-4
	Stopping the Scheduler.	3-5
	Starting the Scheduler in Windows NT	3-6
	Running Scheduler as a Windows NT Service	3-6
	Configuring ODBC	3-7
Chapter 4	Agent Administrator	
	About MetaCube Agent Administrator	4-3
	The Agent Administrator User Interface	4-4
	Using MetaCube Agent Administrator	4-4
	Starting Agent Administrator	4-5
	Managing the Job Queue	4-5
	Submitting Jobs	4-9
	Other Administrative Tasks	4-10
Chapter 5	Secure Warehouse	
	About MetaCube Secure Warehouse	5-3
	The Secure Warehouse User Interface	5-3
	Active and Inactive Users	5-5
	The Process of Using Secure Warehouse	5-6
	Who Can Access Data?.	5-7
	What Data Can the User Access?	5-8
	When Can the User Access Data?	5-8
	How Does the User Access Data?	5-8
	Granting Access to the Data Warehouse	5-9
	Managing Groups of Users	5-10
	Managing Snap-ins	5-10
Chapter 6	Warehouse Optimizer	
	About MetaCube Warehouse Optimizer	6-3
	Before You Begin	6-4

Setting Fact Table Costs in Warehouse Manager	6-4
Configuring Memory for Warehouse Optimizer	6-5
Validating Existing Aggregate Tables	6-6
Creating a Sample Table	6-6
How to Calculate Combination Totals	6-6
The Warehouse Optimizer User Interface	6-8
Using Warehouse Optimizer: An Overview	6-9
Connecting to the Database	6-9
Configuring the Analysis	6-10
Choosing a Data Source	6-10
Selecting Analysis Type	6-10
Setting the Number of Dimensions in Candidate Aggregates	6-11
Generating Costs	6-12
Reducing Cost Generation Time	6-12
How Warehouse Optimizer Generates Costs	6-13
Recommending Aggregates	6-15
Audit Data	6-15
How Many Aggregates to Recommend	6-16
Viewing Recommended Aggregate Details	6-17
Registering Aggregates	6-18
Optimizer Performance Log	6-19
Updating Recommendations	6-20

Appendix A Submitting a Web Publisher Job in DOS

Appendix B Understanding Parameters

Parameters in QueryBack Jobs.	B-2
User-Defined Parameters	B-4
Parameters in SQL Jobs	B-5

Appendix C The metacube.ini File

Index

Introduction

Organization of This Manual	3
Types of User	4
Documentation	5
Printed Documentation	5
Typeface	7
Online Help	7
Readme Files	8
Related Reading	8
Compliance with Industry Standards	9
Informix Welcomes Your Comments	9

T

his manual contains information to assist you in using MetaCube data warehousing software.

Organization of This Manual

The chapters in this manual describe the process and the software components necessary to design, deploy, and manage a MetaCube data warehouse. Generally, this manual provides an overview of the many procedures a data warehouse administrator performs while creating and maintaining a data warehouse. Detailed descriptions of particular procedures can be found in other MetaCube manuals or in online help systems for each MetaCube product.

This manual includes the following chapters:

- This Introduction provides an overview of the manual.
- [Chapter 1, “Overview of the MetaCube System,”](#) introduces the concept of data warehousing and describes the software components of the MetaCube system, paying particular attention to the components generally used for managing a data warehouse. The chapter also describes the typical system architectures that MetaCube supports.
- [Chapter 2, “Setting Up a Data Warehouse,”](#) outlines the process necessary to design and deploy a MetaCube data warehouse. Where appropriate, this chapter references other sources of detailed information.
- [Chapter 3, “MetaCube Scheduler,”](#) introduces MetaCube Scheduler, a server-side scheduling mechanism that controls the operation of the various MetaCube Agent processes. The chapter describes how to install and monitor MetaCube Scheduler on both a UNIX and Microsoft Windows NT platform.

- [Chapter 4, “Agent Administrator,”](#) introduces MetaCube Agent Administrator, a tool for managing MetaCube Scheduler and the many Agent processes Scheduler can spawn.
- [Chapter 5, “Secure Warehouse,”](#) introduces MetaCube Secure Warehouse, a tool that gives data warehouse administrators the ability to control user access to DSS Systems and other sensitive user information.
- [Chapter 6, “Warehouse Optimizer,”](#) introduces MetaCube Warehouse Optimizer, a tool that helps data warehouse administrators design aggregate tables to optimize query performance for users.
- [Appendix A](#) describes MetaCube Web Publisher’s DOS utility for creating HTML-formatted files that hold the results of a query.
- [Appendix B](#) describes the use of parameters in the MetaCube system.

Types of User

This manual is written for data warehouse administrators who are responsible for creating and managing a data warehouse, the metadata contained in a data warehouse, and the end users who will access that metadata. Although data warehouse administrators should understand how all the components of a MetaCube system work, primarily they will use the following tools:

- MetaCube Secure Warehouse
- MetaCube Warehouse Manager
- MetaCube Warehouse Optimizer
- MetaCube Agent Administrator
- MetaCube Web Publisher

Documentation

MetaCube product documentation contains two key components:

- printed documentation
- online help

Printed Documentation

The printed documentation for MetaCube products is divided into two distinct types:

- manuals for products with graphical user interfaces
- manuals for application development tools

Manuals for products with user interfaces contain information about the various features of the product. This information is fairly high-level and conceptual in nature. These manuals are designed to give you information about the purpose of the product and what its capabilities are.

Manuals for application development tools are reference manuals that provide technical information about the tools.

In addition to this book, printed manuals for other MetaCube products include the following:

- *MetaCube Explorer User's Guide*. This manual is written for people who are responsible for analyzing data about their company's business. It describes the features of MetaCube Explorer for querying a MetaCube data warehouse in multi-dimensional terms to obtain meaningful reports that are the basis for timely business decisions.
- *MetaCube for Excel User's Guide*. This manual is written for people who use Microsoft's Excel spreadsheet for business analysis. After adding in MetaCube for Excel to the Excel software, an Excel user can query a MetaCube data warehouse in multi-dimensional terms to obtain spreadsheet or PivotTable reports.
- *MetaCube Warehouse Manager's Guide*. This manual is written for the data warehouse administrator and describes how to specify internal information about the data warehouse so that the MetaCube components are able to access and graphically present the database for querying.

- [*MetaCube Application Programmer's Manual*](#). This manual is written for a programmer who will write custom applications that interact with the MetaCube analysis engine. This manual describes MetaCube's OLE Automation programming interface.
- [*MetaCube SDK for Snap-Ins Programmer's Manual*](#). This manual is written for the C++ programmer who will write custom measure calculations for MetaCube Explorer and MetaCube for Excel using the MetaCube SDK for Snap-Ins. The SDK's Extension Wizard generates skeletal code that is a framework for adding custom C++ code for customized measure calculations.
- [*MetaCube SQL Optimizer User's Guide*](#). This manual describes how to use the MetaCube SQL Optimizer for connecting third-party query tools or custom query applications to the MetaCube analysis engine to access a MetaCube data warehouse. Queries are optimized to run against aggregate and sample tables, thereby significantly improving query performance against very large data warehouses.
- [*MetaCube Installation and Configuration Guide*](#). This manual describes how to install and configure the MetaCube software components on both the server and on PCs.
- [*Introduction to New Features*](#). This guide describes the new features and enhancements for Release 4.0 of MetaCube. It provides information to existing MetaCube users.

Typeface

The text in this manual uses the following set of conventions.

Convention	Meaning
<i>italics</i>	Emphasized words appear in italics. Also used for the names of MetaCube components and some terms that are specific to MetaCube Web Explorer
boldface	Used for the names of menu options, popup menu options, and dialog boxes.
monospace	Used for information that the product displays and information that you type.

Online Help

Each MetaCube product that employs a graphical user interface includes an extensive online help system that provides step-by-step instructions on the use of the product. The help system for most MetaCube products is accessed from the Help menu and consists of the following components:

- **Help Topics:** displays a complete online help system that contains “how to” topics and procedural information on using the product.
- **Context Sensitive Help:** allows the user to access help specifically on a feature of the user interface. Selecting this menu option changes the cursor into a help cursor. Clicking a feature of the window with the help cursor displays help about that specific feature.
- **Help on Help:** contains instructions on using the MetaCube online help. For the user unfamiliar with using online help, this help system explains how to use the various features of MetaCube’s online help system.

Note: For *MetaCube for Excel*, the Help system is accessed from the *MetaCube* menu in *Excel*. For *MetaCube Web Explorer*, the Help system is accessed from the Help button on the toolbar.

Context-sensitive help can be accessed by clicking the Help button on an application’s toolbar in the main window. After clicking the Help button, clicking any feature of the window allows you to obtain help about that specific feature. This is the same as clicking Context Sensitive Help on the window’s Help menu.

In dialog boxes, pressing **F1** displays a help topic that describes the features of that dialog box. From that topic, the user may jump to the main help system or, in many cases, to a procedural help topic that describes how to use the dialog box. In the main window of an application, after activating an area of the window by clicking it, **F1** provides a help topic that describes that particular area of the window.

Readme Files

In addition to the printed manuals, readme files are distributed with MetaCube products. These files contain technical information, including last-minute changes to product capability or documentation. Please read these files, as they contain important information.

Related Reading

For information on data warehousing, see *The Data Warehouse Toolkit*, by Ralph Kimball (John Wiley & Sons, Inc., 1996).

Compliance with Industry Standards

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992, on INFORMIX-OnLine Dynamic Server. In addition, many features of OnLine comply with the SQL-92 Intermediate and Full Level and X/Open C CAE (common applications environment) standards.

Informix SQL-based products are compliant with ANSI SQL-92 Entry Level (published as ANSI X3.135-1992) on INFORMIX-SE with the following exceptions:

- Effective checking of constraints
- Serializable transactions

Informix Welcomes Your Comments

Please tell us what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about any corrections or clarifications that you would find useful. Please include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Write to us at the following address:

Informix Software, Inc.
Technical Publications
300 Lakeside Drive, Suite 2700
Oakland, CA 94612

If you prefer to send electronic mail, our address is:

doc@informix.com

Or, send a facsimile to Technical Publications at:

650-926-6571

We appreciate your feedback.



Overview of the MetaCube System

What is a MetaCube Data Warehouse?	1-4
MetaCube Software Components	1-5
MetaCube Analysis Engine.	1-5
Tools for Querying the Data Warehouse	1-5
Tools for Managing the Data Warehouse	1-6
MetaCube Warehouse Manager	1-6
MetaCube Secure Warehouse	1-7
MetaCube Scheduler	1-7
MetaCube Agent Administrator	1-8
MetaCube Warehouse Optimizer	1-8
MetaCube Architecture	1-9
Client/Server on Three Tiers	1-9
Client/Server on Two Tiers.	1-11

T

his chapter introduces the MetaCube data warehouse, including its system architecture and software components.

What Is a MetaCube Data Warehouse?

Data warehousing is the process of integrating data from a variety of sources into a single repository. Business analysts can query a data warehouse and generate reports and graphs to support their decision-making processes.

A data warehouse may consist of a single, large, enterprise-wide database to which users and administrators can connect, or it may incorporate several smaller systems, often called data marts. Typically, each data mart addresses a specific subject area within a larger data warehouse.

MetaCube provides a sophisticated mechanism for querying data warehouses and data marts by integrating online analytical processing (OLAP) tools with relational database technology. MetaCube offers high-performance Structured Query Language (SQL) algorithms, mathematical summarization, cross-tabulation, pivoting functions, and decision-support query extensions. Moreover, MetaCube simplifies data warehouse searches by presenting data in everyday business language that ordinary users can understand.

MetaCube Software Components

The MetaCube software components consist of the MetaCube analysis engine, the tools for querying a data warehouse, and the tools for managing a data warehouse.

MetaCube Analysis Engine

The MetaCube analysis engine provides a multidimensional, object-oriented programming interface to an Informix relational database. It processes data requests submitted by querying tools, such as MetaCube Explorer, and generates SQL, which it sends to the database via an ODBC (Open Database Connectivity) driver. The MetaCube analysis engine extends the analytical functionality of the database by performing pivots and calculations on data sets before returning data to the querying tool.

All MetaCube applications connect to the database through the MetaCube analysis engine. Launching any MetaCube application automatically launches the MetaCube analysis engine itself.

The application programming interface for the MetaCube analysis engine is documented in the [MetaCube Application Programmer's Manual](#).

Tools for Querying the Data Warehouse

A MetaCube system provides the following graphical, ad hoc data access tools that allow both novice and experienced decision support analysts to access and analyze a data mart or data warehouse:

- MetaCube Explorer—a graphical interface that allows users to submit queries to the database through the object-oriented programming interface of the MetaCube analysis engine.
- MetaCube Web Explorer—the same interface as MetaCube Explorer except that MetaCube Web Explorer runs in a web browser.
- MetaCube for Excel—an add-in to Microsoft Excel. This product incorporates multidimensional access into standard Excel features, returning data directly into an Excel spreadsheet.
- MetaCube SQL Optimizer—a library of routines that allows the optimization of SQL generated by a query tool or custom query application so that an end user's query can run against a MetaCube data warehouse.

Tools for Managing the Data Warehouse

A MetaCube system provides the following tools for managing a data warehouse:

- MetaCube Warehouse Manager
- MetaCube Secure Warehouse
- MetaCube Scheduler and various MetaCube Agents
- MetaCube Agent Administrator
- MetaCube Warehouse Optimizer

MetaCube Warehouse Manager

MetaCube Warehouse Manager provides data warehouse administrators with a graphical interface for creating a multidimensional description of tables and columns in the Informix database. The MetaCube analysis engine stores that multidimensional description as a set of MetaCube system tables in the Informix database. These tables are known as metadata, and they can be used to configure applications that issue queries to the database through the MetaCube analysis engine, thus shielding users from the complexity of the physical model. Warehouse Manager can also describe summary tables, known as aggregate tables, and statistically representative sample tables. MetaCube can access these tables instead of the larger tables from which they were derived, thereby improving query performance.

The [*MetaCube Warehouse Manager's Guide*](#) describes the use of Warehouse Manager.

MetaCube Secure Warehouse

MetaCube Secure Warehouse provides data warehouse administrators with a graphical user interface for controlling end user access to data. With Secure Warehouse, an administrator can specify which users have access to a DSS System. (A DSS System is a view of a data warehouse, often corresponding to the requirements of a particular business unit, such as a finance or marketing department.) The data warehouse administrator can also control what information each user can see in a DSS System, and when that user can access information by performing queries.

No user of any MetaCube product can access a DSS System unless authorized to do so in MetaCube Secure Warehouse. Users of Web Explorer cannot even connect to the database unless they have been properly defined as users in Secure Warehouse.

Secure Warehouse is documented in this guide and in the product's online help.

MetaCube Scheduler

MetaCube Scheduler is a server-side scheduling mechanism that controls the operation of the various MetaCube Agent processes. Those processes can execute SQL statements, QueryBack functions, full and incremental aggregation functions, sample tables, Alert jobs, Web Publisher jobs, and even operating system commands. Scheduler allows users to submit jobs or job sets to the server. It evaluates each job's priority, the privileges of the user who submitted the job, and any dependencies or conditions that must be satisfied before the job can run. It then initiates the appropriate process to execute the job on the server.

MetaCube Scheduler is documented in this guide.

MetaCube Agent Administrator

MetaCube Agent Administrator provides data warehouse administrators with a graphical interface for administering all the features and functions of MetaCube Scheduler and the various Agents it generates. With Agent Administrator, you can manage the job queue, submit new jobs, store job sets, and perform other administrative duties. Agent Administrator also allows an administrator to create aggregate and sample tables, which can improve query performance.

Agent Administrator is documented in this guide and in the product's online help.

MetaCube Warehouse Optimizer

MetaCube Warehouse Optimizer analyzes a data model (the logical representation of a data warehouse) and reviews the queries issued against it to recommend a strategy for creating aggregate tables. To improve performance for resource-intensive queries, aggregate tables group and sum detailed transactional records. Normally, queries requesting such summary-level data require the database to perform the sum while processing the query, but MetaCube automatically routes those queries to an aggregate table if one exists, retrieving the answer without further processing. If desired, MetaCube Warehouse Optimizer can generate the metadata necessary to create aggregate tables.

Warehouse Optimizer is documented in this guide and in the product's online help.

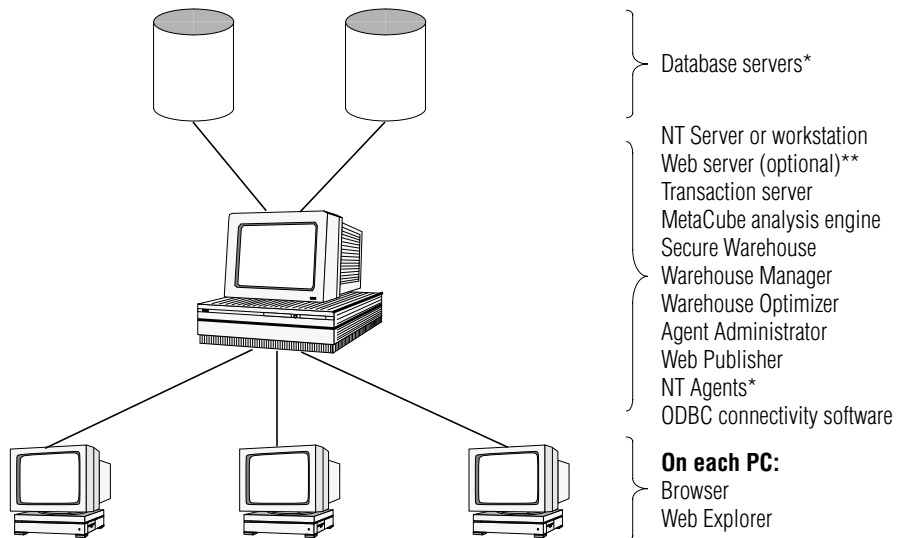
MetaCube Architecture

MetaCube's system architecture consists of software components configured in a two-tiered or three-tiered client / server design or a combination of both structures. The MetaCube system also incorporates security features that verify users at the time of connection.

Client/Server on Three Tiers

A three tiered architecture typically supports users running MetaCube Web Explorer through a web browser. [Figure 1 on page 1-8](#) shows a typical three-tiered architecture.

Figure 1
Three-tiered architecture



* MetaCube NT Agents can run on the middle tier or on the same platform with the database servers

** Web server may reside on any machine that can provide HTTP services

In this configuration, the middle tier is a PC or server machine running the Microsoft Windows NT operating system. The middle tier machine must run Microsoft Transaction Server, a system for managing shared OLE Automation Servers, such as the MetaCube analysis engine. Optionally, the middle tier can also serve as a Web server, although the Web pages for MetaCube Web Explorer can be served from any Web server on the local area network.

All of the MetaCube components used to create and manage a data warehouse and to process queries are installed and running on the middle tier. These components include the MetaCube analysis engine, Secure Warehouse, Warehouse Manager, Agent Administrator, Warehouse Optimizer, Web Publisher, and the MetaCube Agents. Optionally, MetaCube Agents can be installed on the database server tier.

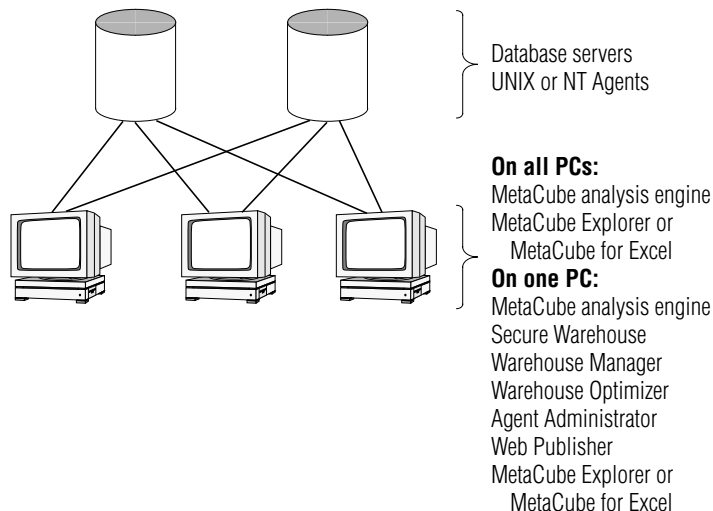
Typically the client tier consists of personal computers equipped with web browsers from which users can run MetaCube Web Explorer. In some situations the client tier may consist of personal computers running MetaCube Explorer or MetaCube for Excel served by the MetaCube analysis engine running on the middle tier.

Database servers constitute the server tier of this architecture. While Figure 1 shows two database servers, only one is necessary.

Client/Server on Two Tiers

MetaCube supports a two-tiered client / server architecture, as shown in Figure 2. In this configuration, the database server must run either UNIX or Window NT Agents. The clients are personal computers running the MetaCube analysis engine and either MetaCube Explorer or MetaCube for Excel. In this type of configuration, at least one PC must support the MetaCube administrative tools, including MetaCube Secure Warehouse, which authorizes user access. Note that while Figure 2 shows two database servers, only one is necessary.

Figure 2
Client/server architecture





Setting Up a Data Warehouse

Overview of the Process	2-3
Design a Schema	2-5
Populate the Data Warehouse Tables.	2-6
Install the MetaCube Software	2-6
Generate MetaCube's Metadata Tables	2-6
Create Authorized Users	2-7
Create Metadata	2-7
Define Public Filters and Queries and Design a Folder Structure	2-8
Using Folders	2-9
Define User Access	2-9
Start MetaCube Scheduler	2-10
Build Aggregate Tables	2-10
Build Sample Tables	2-11
Continuing Maintenance of a Data Warehouse	2-12

T

his chapter describes how to set up a MetaCube data warehouse. The first section provides an overview of the process. Subsequent sections explain each of the steps.

Building a data warehouse requires you to use all of the components of the MetaCube system. Throughout this chapter, sources for more detailed information are referenced. Those sources may be other chapters of this book, online help systems, or other MetaCube manuals.

Overview of the Process

This section provides a step-by-step description of the process necessary to build a data warehouse, starting at the planning phase, moving through installation, and ending with deployment of the system. Each of the steps listed in this section are described in greater detail in the subsequent sections of this chapter. For example, step 1, listed below, is described in [“Design a Schema” on page 2-5](#).

1. Design a schema

The first step in creating a data warehouse is to design a data model for storing information in the data warehouse. The MetaCube analysis engine supports a star, snowflake, or partial snowflake schema.

2. Populate the data warehouse tables

You must create the necessary tables in the Informix database and populate them with data.

3. Install the MetaCube software

If you have not done so already, you must install the MetaCube software.

4. Generate MetaCube's metadata tables

The MetaCube installation provides scripts that must be run to create metadata tables. Before executing these scripts you should create a user called “metacube,” who serves as the schema / table owner for MetaCube's metadata tables, and a user “metapub,” who is needed for the next step.

5. Create authorized users

Log onto MetaCube Secure Warehouse as the user “metapub” and create users with the necessary privileges to run the MetaCube data warehouse administrative tools: MetaCube Secure Warehouse, MetaCube Warehouse Manager, and MetaCube Warehouse Optimizer.

6. Create metadata

Use MetaCube Warehouse Manager to create or update metadata for all the DSS Systems and MetaCube data sources (not to be confused with ODBC datasources) that users will access. A DSS System is a view of a data warehouse, often corresponding to the requirements of a particular business unit, such as a finance or marketing department. MetaCube data sources correspond to the fact tables within each DSS System, such as a company's sales transactions. During this step you also define aspects of metadata required for the user interface of MetaCube's querying tools, such as MetaCube Explorer.

7. Define public filters and queries and design a folder structure

Log onto MetaCube Explorer as the user “metapub” and define public filters and queries for each DSS System in the data warehouse. If necessary, implement a folder structure to be used by all users in the organization.

8. Authorize user access

Use MetaCube Secure Warehouse to authorize access to DSS Systems for all users. With Secure Warehouse, you can also specify what information each user can see in the DSS System, and when that user can execute background queries and other jobs.

9. Start MetaCube Scheduler

MetaCube Scheduler is a server-side scheduling mechanism that controls the operation of the various MetaCube Agent processes. Scheduler allows MetaCube Agent Administrator and Explorer users to submit jobs to the server.

10. Create aggregate tables

If your data warehouse needs aggregate tables, use MetaCube Warehouse Optimizer to design them and to generate the necessary metadata descriptions. Use MetaCube Agent Administrator to submit the necessary jobs to create those tables.

11. Create sample tables

If your data warehouse needs sample tables, use MetaCube Warehouse Manager to create metadata descriptions of those tables. Use Agent Administrator to submit the jobs necessary to create those tables.

Design a Schema

When creating a data warehouse, the first step is to design a data model. To provide decision support, the database tables used to support a data warehouse must be built to minimize the overall number of records processed, unlike traditional transactional databases, which typically seek to minimize the number of records stored. A variety of techniques and tools exist to design and build data models. MetaCube supports star, snowflake, or partial snowflake schemas.

For a discussion of schema design, see the [MetaCube Warehouse Manager's Guide](#).

Populate the Data Warehouse Tables

The next step is to create and populate the tables in the underlying Informix database that will comprise your data warehouse. Central fact tables contain the factual data you want to track and report, such as sales information. Dimension tables contain descriptive data, such as product or geographic information, that allow the MetaCube analysis engine to present multidimensional views of data. The needs of your particular business and data warehouse will dictate the types of tables that must be created.

Install the MetaCube Software

If the MetaCube system has not already been installed, you should install it now, as described in the [MetaCube Installation and Configuration Guide](#).

Generate MetaCube's Metadata Tables

The MetaCube installation process provides you with many SQL scripts. One of the many functions of these scripts is the creation of metadata tables, which store descriptions of the data model used for the data warehouse. The metadata tables also store definitions and characteristics of jobs submitted to the MetaCube Agents, user properties, and saved queries and filters.

The user “metacube” owns the scripts that create the metadata tables. Either “metacube” must run those scripts or they must be edited to specify another owner of the metadata tables, and that owner must run the scripts.

The [MetaCube Installation and Configuration Guide](#) describes how to install metadata tables, including how to create the user “metacube.”

Create Authorized Users

Only authorized users can access the data warehouse administrative tools: MetaCube Secure Warehouse and MetaCube Warehouse Manager. When setting up a data warehouse, many tasks require you to use these two applications. Only authorized users can use MetaCube Warehouse Optimizer to create metadata descriptions of aggregate tables, although any MetaCube user can access the other capabilities of Warehouse Optimizer.

By default, the user “metapub” can always access Secure Warehouse, Warehouse Manager, and Warehouse Optimizer. The password for this user is determined during the installation process, as described in the [MetaCube Installation and Configuration Guide](#).

The first time you access Secure Warehouse, you should log in as metapub. Then you can register any other users who will serve as data warehouse administrators and have access to Secure Warehouse, Warehouse Manager, and Warehouse Optimizer. To give those users data warehouse access, click **Allow user to access Secure Warehouse** on the Connection tab of each user’s properties sheet in Secure Warehouse.

Create Metadata

Metadata is a logical, multidimensional description of relational data. Essentially, metadata serves as MetaCube’s map to a data warehouse.

For the most part, you create metadata using MetaCube Warehouse Manager, although you also generate user-related metadata with MetaCube Secure Warehouse. When you are setting up a data warehouse, you must use MetaCube Warehouse Manager to create objects representing elements of all the DSS Systems necessary for the data warehouse and all the MetaCube data sources necessary for each DSS System.

Primarily, the objects you define in Warehouse Manager represent dimensions, dimension elements, attributes, and fact tables. Dimension elements correspond to levels of hierarchy within a dimension. For example, a product dimension might have three hierarchical levels—that is, three dimension elements. The “supplier” dimension element represents the highest level of the dimension’s hierarchy; the next level is the “brand” dimension element; the lowest level is the “product” dimension element. Attributes represent the actual data a user might see in a report. For example, the product dimension element might contain two attributes: Product Name and UPC.

When you initially create the objects for a DSS System, you have the option of also creating metadata describing aggregate and sample tables. You may want to wait until users begin querying your DSS System before you create either of these types of tables. Actual usage may influence your decisions concerning sample and aggregate tables.

After you have created metadata for a DSS System, you should verify the integrity of that data, a simple one-step procedure using Warehouse Manager.

The [MetaCube Warehouse Manager’s Guide](#) describes in detail all the tasks necessary for creating metadata for a DSS System.

Define Public Filters and Queries and Design a Folder Structure

All users of Explorer, Web Explorer, and MetaCube for Excel have access to public queries and filters stored in metadata for a DSS System. As the data warehouse administrator, you should create all necessary public queries and filters by logging into Explorer as the user “metapub.” The procedure for creating a public filter or query is the same as the procedure for creating any filter or query in Explorer except that the user performing these actions is “metapub.” The user “metapub” can also designate default filters. For more information about creating queries, filters, and default filters, see the [MetaCube Explorer User’s Guide](#).

Because public queries are typically accessed by many users, modifications to or deletions of public queries should be done carefully. Be sure to notify all users of changes made to these objects.

You can define public filters and queries any time after metadata has been created for a DSS System and the “metapub” user has been granted access to that DSS System in MetaCube Secure Warehouse.

Using Folders

When users save queries and filters, they are stored in folders and sub-folders organized in a tree structure. The top-level folder is named ROOT. You may want to create sub-folders if your users can create and store filters and queries, especially if you anticipate that a DSS System will require a large number of filters. Many MetaCube installations create a folder for each dimension so that a user can easily locate filters associated with a particular dimension, such as Geography.

All folders in the data warehouse are visible to all users of Explorer, Web Explorer, and MetaCube for Excel. That is, a single set of folders and sub-folders are all accessible to all users. For more information about folders, see the [MetaCube Explorer User's Guide](#).

Define User Access

Use MetaCube Secure Warehouse to control user access to DSS Systems. With Secure Warehouse's graphical interface, you can specify which users have access to a DSS System, what information each user can obtain from the DSS System, and when that user can access information by executing background queries and other types of jobs (only administrators can execute other types of jobs).

Essentially, you accomplish these tasks by identifying the users who will be managed, assigning DSS System access to those users, and applying mandatory filters to users on a per-DSS-System basis. Mandatory filters limit what information a user can actually see in each DSS System.

You can define user access with Secure Warehouse any time after you have created metadata for a DSS System. [Chapter 5, “Secure Warehouse,”](#) introduces MetaCube Secure Warehouse, and online help for Secure Warehouse provides all necessary procedural descriptions.

Start MetaCube Scheduler

MetaCube Scheduler is a server-side scheduling mechanism that controls the operation of the various MetaCube Agent processes. Scheduler allows users of MetaCube Agent Administrator to submit jobs to the server and MetaCube Explorer users to submit background queries. It evaluates each job's priority, the privileges of the user who submitted the job, and any dependencies or conditions that must be satisfied before the job can run. It then generates the appropriate process to execute the job on the server.

You can start MetaCube Scheduler any time after the MetaCube software is installed, but you *must* start Scheduler before you run any jobs using MetaCube Agent Administrator or MetaCube Warehouse Optimizer. The [MetaCube Installation and Configuration Guide](#) describes how to install the Scheduler. In this book, [Chapter 3, “MetaCube Scheduler,”](#) describes how to start the Scheduler.

Build Aggregate Tables

To improve performance, most data warehouses include aggregate tables, which contain summarized data. The MetaCube analysis engine automatically routes queries for summarized information to an aggregate table if a suitable one exists.

Using MetaCube Warehouse Optimizer, you can automatically determine the best type of aggregate tables to build, and you can generate the necessary metadata descriptions of those tables. If you prefer, you can use Warehouse Manager to create the metadata descriptions of those aggregate tables yourself. After the metadata descriptions of aggregate tables exist, you use MetaCube Agent Administrator to submit a job that generates the aggregate table or tables.

Note that any MetaCube user can access Warehouse Optimizer, but to create metadata descriptions of aggregate tables, you must be an authorized data warehouse administrator. In other words, you must be a secure user.

Although you can generate aggregate tables any time after their metadata descriptions have been built and the MetaCube Scheduler is running, you may want to activate query auditing and wait until users have begun querying your data warehouse to determine the most strategic use of aggregate tables.

[Chapter 6, “Warehouse Optimizer,”](#) describes MetaCube Warehouse Optimizer, and [Chapter 4, “Agent Administrator,”](#) describes MetaCube Agent Administrator. Online help for both products provides all necessary procedural descriptions.

Build Sample Tables

To improve performance in large data warehouses, one or more sample tables can provide statistically accurate results for queries in much less processing time than it would take to query a large fact table. A MetaCube Agent, MetaCube Sampler can build sample tables based on your specifications. You may wish to build several sample tables in your data warehouse, each with a different level of accuracy. This offers users flexibility when specifying the degree of confidence they need for a particular report.

Using Warehouse Manager, you can generate the necessary metadata descriptions of sample tables. After the metadata descriptions exist, you can submit a job to generate the physical sample table or tables using MetaCube Agent Administrator.

Although you can create sample tables any time after you build their metadata descriptions and the MetaCube Scheduler is running, you may want to wait until users have begun querying your data warehouse to determine the most strategic use of sample tables.

The [MetaCube Warehouse Manager’s Guide](#) describes how to create metadata for sample tables. [Chapter 4, “Agent Administrator,”](#) describes the use of MetaCube Agent Administrator, and online help for Agent Administrator describes how to submit a Sample job.

Depending on the product families you have purchased for your Informix Dynamic Server, you may not need to use MetaCube Sampler to create physical sample tables. The Extended Parallel Option, for example, provides “on the fly” sampling that the MetaCube analysis engine can use in place of physical sample tables.

Continuing Maintenance of a Data Warehouse

After you have created a data warehouse, you will need to perform additional procedures to maintain it. Some of the most common tasks are listed below:

- Use Secure Warehouse to add new users and remove users who are no longer authorized.
- Update aggregate tables to reflect new information added to a fact table. To accomplish this you must first create an incremental fact table to house information that is added periodically. Then you must use Agent Administrator to submit two jobs: one creates incremental aggregate tables and the other incorporates those tables into the existing fact and aggregate tables. Optionally, you can check the validity of the incremental aggregate tables before incorporating them into the existing aggregate tables.
- Update sample tables to reflect new information added to a fact table. Use Agent Administrator to submit a job that generates new sample tables.
- Use MetaCube Warehouse Manager to modify metadata.
- Use Agent Administrator to submit other types of jobs, such as SQL Select and SQL (non-Select) jobs. For example, you could schedule a job that automatically updates your current period.
- Use Agent Administrator to control MetaCube Scheduler.

You can use MetaCube Agent Administrator to define a Job Set that automates many of the tasks described above. With Agent Administrator, you can define dependencies between jobs in a Job Set, and you can run the Job Set job at any time or schedule it to run on a periodic basis. For more information on dependencies and Job Sets, see [Chapter 4, “Agent Administrator,”](#) and Agent Administrator’s online help.

MetaCube Scheduler

About MetaCube Scheduler	3-3
Starting the Scheduler in UNIX	3-4
Skipping the Prompts	3-4
Checking the Status of the Scheduler	3-5
Stopping the Scheduler	3-6
Starting the Scheduler in Windows NT	3-7
Running Scheduler as a Windows NT Service	3-7
Configuring ODBC	3-8
Configuring Database logins and ODBC Data Sources	3-8
Configuring Startup Options.	3-9

This chapter introduces the MetaCube Scheduler, a server-side scheduling mechanism that controls the operation of the various MetaCube Agent processes.

About MetaCube Scheduler

MetaCube Scheduler allows users to submit jobs to the server. It evaluates each job's priority, the privileges of the user who submitted the job, and any dependencies or conditions that must be satisfied before the job can run. It then spawns the appropriate MetaCube process to execute the job on the server.

MetaCube Scheduler offers more sophisticated features than an operating system scheduler, such as UNIX cron. Examples of the MetaCube Scheduler's features include:

- user- and job-level priorities
- ability to designate the times and days of the week at which a particular user may run jobs
- multi-processor support
- support for multiple jobs, with dependencies between jobs, as well as dependencies on external conditions
- ability to submit and monitor jobs from either the client or the server.

MetaCube Scheduler serves as the parent process to all of the MetaCube Agent processes. These processes include jobs spawned to execute SQL statements, QueryBack functions, full and incremental aggregation functions, and even operating system commands.

Once the Scheduler has been started, all MetaCube Agent functions can be managed from the graphical Windows environment of MetaCube Agent Administrator. The Scheduler itself, however, must be started at the server.

Starting the Scheduler in UNIX

To start the Scheduler, log on to the server operating system under which you installed MetaCube Agents software. At the operating system prompt, type:

```
sc run
```

where “sc” is the name of the Scheduler process. MetaCube Scheduler will prompt you for a login and database name. After entering this information, you will be returned to the operating system prompt.

Skiping the Prompts

To preclude the Scheduler from prompting you for a login and database name, you can create a resource file, named `.scrc`, to store the appropriate values for the login and database name, which the Scheduler passes to the database. The `.scrc` file is documented in [MetaCube Installation and Configuration Guide](#) for your server platform.

In the `.scrc` file, enter a line using the following format:

```
loginID#database_name
```

where *loginID* is your Informix database login and *database_name* is the name of the database where the metadata and data warehouse files are stored.

Note: Your database login must have DBA privileges.

Checking the Status of the Scheduler

You can check the status of the Scheduler process and any queued jobs by typing

```
sc status
```

at the operating system prompt. This produces status information similar to that shown below:

```
Run_State is set to           Normal Operation
max_jobs is                  5
sleep_time is                 30
number of jobs running:      0
number of jobs pending:      0
```

This status information is actually stored in a database table. In the event of system failure or other abnormal occurrence, this information may not be precisely synchronized with actual operating system events.

You can verify that the Scheduler process is actually running at the operating system level by viewing the running processes. The syntax for checking operating system processes varies depending on your operating system platform. The examples below illustrate how to view running processes from the BSD UNIX and UNIX SYSTEM V operating systems.

Example: BSD UNIX

```
ps -aux | grep sc
```

The `ps` command provides output similar to that shown:

```
root      0 80      0      0 ?      T   Sep 12 0:02 sched
user1    16207  9    147   107 pts/12  S   10:47:44  0:00 grep sc
user1    16200 14    616   237 pts/12  S   10:47:16  0:00 sc run
```

Example: UNIX System V Release 4

```
ps -ef | grep sc
```

Stopping the Scheduler

To stop MetaCube, at the operating system command prompt type:

```
sc stop
```

The Scheduler stops, but any processes already spawned by the Scheduler will complete. The operating system displays the message:

```
MetaCube Scheduler Stopped...
Current Jobs will finish.
```

and you are returned to the operating system command line. This action records in the database a command to stop the Scheduler process. The process does not necessarily stop immediately. The next time the process wakes from its sleep cycle, it checks for the stop command. When it finds the command, the Scheduler exits gracefully. It is possible to stop the process using operating system commands to kill the process, but this is not recommended, as it may leave the database in an indeterminate state.

With the MetaCube Agents Control dialog box in MetaCube Agent Administrator, you can view the status of Scheduler and you can stop Scheduler. You cannot, however, use that dialog box to start Scheduler. For information on the MetaCube Agents Control dialog box, see [“Other Administrative Tasks” on page 4-10](#)

Starting the Scheduler in Windows NT

The MetaCube Scheduler is implemented on Windows NT as a Windows NT service and it is managed through the **Services** dialog found in the Control Panel. The Scheduler communicates with the target database via an ODBC User Data Source.

The ODBC User Data Source and the Scheduler service must be managed by the same Windows NT user.

Running Scheduler as a Windows NT Service

As a Windows NT administrator with permission to start a service, you can specify the service type for MetaCube Scheduler and then start it.

More than one user may start MetaCube Scheduler. Each user who does so must be a member of the Administrator group with the added permission to log on as a Service. Furthermore, that user must specify and use a separate ODBC User Data Source for the Scheduler to use.

1. Assure that you have run the MetaCube Agents for Windows NT Configuration Utility.
2. Open the Windows NT Control Panel.

3. Click **Services**. The Services dialog box lists all the background services registered within the operating system kernel.
4. In the **Services** dialog box:
 - a. Click **MetaCube Scheduler** in the services list, then click **Startup**, opening a second Service dialog box.
 The default Startup Type is Manual but you may choose to make it Automatic. Because the Scheduler spawns all other MetaCube Agent processes, ensure that it is running continuously to provide user access to all MetaCube Agent processes.
 - b. In the **Log On As** frame, specify the login name and password of the Windows NT user who created the ODBC User Data Source and who will start and stop the Scheduler. Alternatively, you can use the Task Manager to view processes, where you should see a process named metasc.exe.
5. Click **OK**.
6. Click **Start** to start the MetaCube Scheduler manually the first time. A Service Control message indicates that the Scheduler is being started and its status changes to Started. To stop the Scheduler, click **Stop**.
7. Verify that Scheduler is successfully running as a Windows NT service by viewing the sc.log file in the directory where the MetaCube Agents are installed.

Note: The Scheduler will fail to start if you have not run the Server Installation Command Files for Windows NT. For information on running these scripts, see the [MetaCube Installation and Configuration Guide](#).

Configuring ODBC

For MetaCube Scheduler for Windows NT to run properly, you need to verify that cursor behavior for the ODBC connection for the MetaCube Agents is set correctly. To do this:

1. Activate the ODBC Administrator on the Windows NT platform, then select the User Data Source that uses the INFORMIX-CLI network driver.
2. Select the Advanced options for that User Data Source.
3. Set the Cursor Behavior to “1 - Preserve.”

Configuring Database logins and ODBC Data Sources

When the MetaCube agents for Windows NT were installed, the MetaCube Agents for Windows NT Configuration Utility prompted you for a database login and password, as well as for an ODBC User Data Source to be used by the Scheduler and the other agents. This utility, called NTConfig.exe, can be used to make changes to these options at a later time. This utility may be found in the MetaCube Agents for NT program group and consists of:

- two fields in which to type the database login and password
- a list box in which to select the ODBC User Data Source

Configuring Startup Options

Any Windows NT user with the qualifications described below can start and stop the MetaCube Scheduler.

To start the Scheduler for the first time after installation or immediately after you have reconfigured the Scheduler to change either the user name and password or the ODBC User Data Source information:

1. Click the Startup option in the **Services** dialog box.
2. Highlight the MetaCube Scheduler service in the list of available services.
3. Click **Startup**.
4. Enter the Administrator user's password, then click **OK**.

Once this is done, you can start and stop the Scheduler without entering this information again unless you either change the database login for the Scheduler or create a different ODBC User Data Source for the Scheduler to use.

Agent Administrator

About MetaCube Agent Administrator	4-3
The Agent Administrator User Interface	4-4
Using MetaCube Agent Administrator	4-5
Starting Agent Administrator	4-5
Managing the Job Queue	4-5
How MetaCube Scheduler Processes Jobs in the Queue	4-6
Information Provided by the Job Queue	4-6
Displaying the Contents of the Queue	4-8
Modifying a Job in the Queue	4-8
Submitting Jobs.	4-10
Other Administrative Tasks	4-11

T

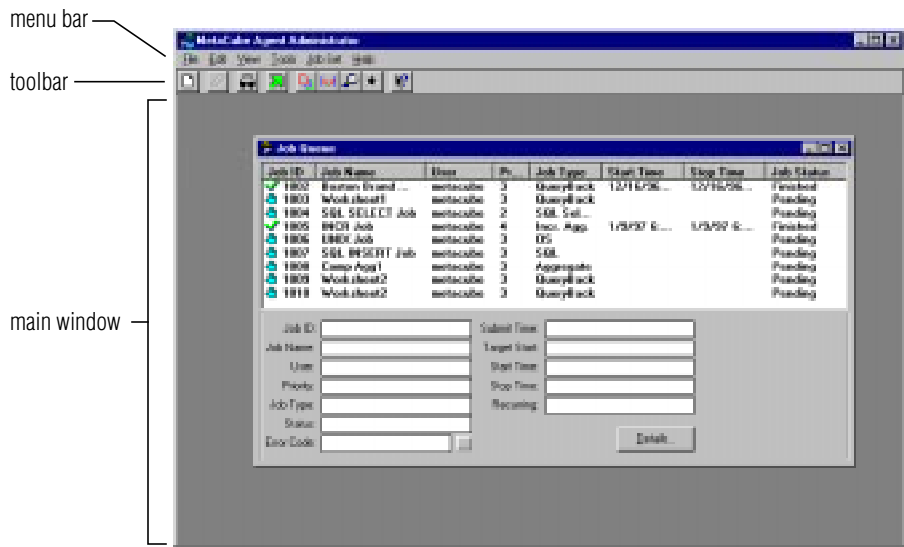
his chapter introduces MetaCube Agent Administrator, a tool for managing MetaCube Scheduler and the many Agent processes the Scheduler can initiate.

About MetaCube Agent Administrator

MetaCube Agent Administrator is a graphical software system that allows you to administer all the features and functions of MetaCube Scheduler as well as the various Agents that it spawns. With Agent Administrator, you can manage the job queue, submit new jobs, and perform other administrative duties that a MetaCube installation may require.

The Agent Administrator User Interface

MetaCube Agent Administrator has a simple user interface. It consists of a menu bar, a tool bar, and a blank main window.



Using MetaCube Agent Administrator

Using MetaCube Agent Administrator consists of the broad tasks listed below:

- starting Agent Administrator
- managing the Job Queue
- submitting Jobs
- performing other miscellaneous administrative tasks

Note: Agent Administrator's online help system provides step-by-step procedures for all the tasks listed above.

Starting Agent Administrator

To start MetaCube Agent Administrator, double-click the MetaCube Agent Administrator icon in the MetaCube program group. After the initialization banner is displayed, a login dialog box prompts you for a user name and password. Logging in connects Agent Administrator to a database.

If you have never previously defined a database connection for MetaCube Agent Administrator, you must do so before attempting to log in. Click the **Cancel** button on the MetaCube Login dialog box, and then choose **Options** on the **Tools** menu. The Preferences dialog box is displayed, which allows you to define database connections. See Agent Administrator's online help for more information on configuring a database connection.

Managing the Job Queue

The job queue is a list of all jobs submitted to the Scheduler for server-side processing, regardless of the application from which jobs are submitted. Database tables store information for each job, including the commands associated with the job, its priority, and the identity of the user who submitted the job.

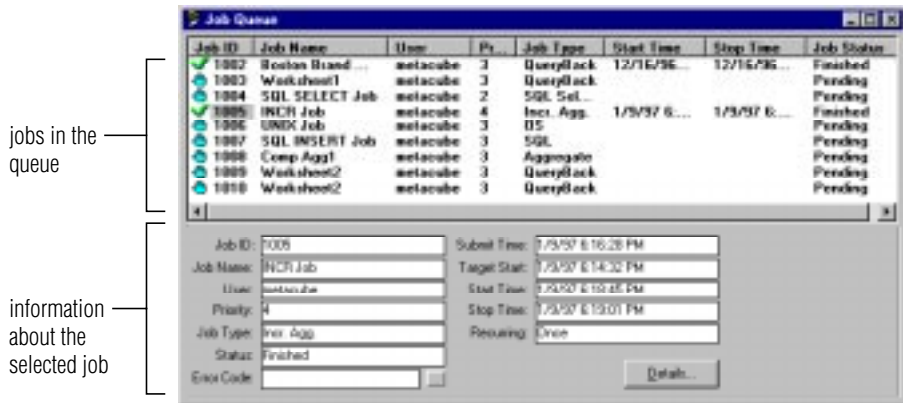
How MetaCube Scheduler Processes Jobs in the Queue

MetaCube Scheduler processes the jobs stored in the queue based on their target start time, their priority, and the maximum number of jobs that can be run concurrently (a value set using Agent Administrator). The Scheduler runs jobs until either the maximum number of concurrent jobs is met or no more jobs require processing. The priority assigned to jobs determines the order in which concurrent jobs run. If a job is submitted to be run immediately, it is assigned a start time of now. If a job is scheduled to be run in the future, the Scheduler runs the job at the designated time.

Periodically the Scheduler polls to see if any client-side processes have made changes to the queue, such as the submission or deletion of a job. The rate at which the Scheduler performs this polling is a value set using Agent Administrator. The Agent Administrator application does not perform any automatic polling of the job queue, so you must refresh the queue yourself to retrieve current information about jobs.

Information Provided by the Job Queue

In Agent Administrator, all jobs in the queue, whether pending, executing, or complete, appear in the Job Queue dialog box.



The Job Queue dialog box lists all the jobs in the queue. Each entry provides the following information for a job:

- A unique identification number
- An icon indicating whether the job is pending, running, or complete and whether the job returned error messages. The status of each entry is also reported with a word in the Job Status column (Pending for jobs waiting to execute, Running for jobs currently executing, or Finished for jobs that have executed).
- The name assigned to the job
- The user who submitted the job
- The priority assigned to the job
- The type of job
- The time when the Scheduler spawned a process to execute the job (blank if the job is pending)
- The time when the job completed (blank if the job is pending or running)

The bottom portion of the Job Queue dialog box provides information in textual form about a specific job. Clicking a Job ID in the job list selects the entry for that job and obtains the most recent information for that job from the database. The bottom portion of the screen provides all of the same job information provided in the job list above but it also delivers the following additional information:

- Recurring status (does the job run once or repeatedly)
- Any error code returned with the job and a button that displays information about the error
- A Details button that displays additional information about the job

Displaying the Contents of the Queue

Using Agent Administrator, you can perform any of the following tasks related to the display of information in the job queue:

- Display the Job Queue dialog box
- Refresh the Job Queue dialog box
Refreshing displays the most recent job data in the Job Queue dialog box.
- Display the most recent information for a particular job
- Display details about a job
The nature of this detailed information varies from job to job.
- Filter the queue
You can define one or more filters that instruct Agent Administrator to display or not display certain types of jobs. When specifying a filter, you choose a property, such as Job Type, a comparison operator, such as equals, and a value to which the property should be compared. For example, you might create a filter that displays only job types equal to SQL Select jobs.
- Sort the queue
By clicking on the header for any job list element, such as the header for the job name or the job ID, you can sort the queue in ascending or descending order for that element.

Modifying a Job in the Queue

While using Agent Administrator, you can modify the contents of the job queue in certain ways. Agent Administrator allows you to:

- Delete a job

Deleting a job from the queue prompts the Scheduler process to delete from the database any temporary tables that may have been created to store the job's result set. A job can be removed from the queue by either the user who originally submitted the job or the data warehouse administrator.

Note: Removing an Aggregate, Incremental Aggregate, or Sample job does not remove the tables created by those jobs.

- Change a job's dependencies

Agent Administrator allows you to impose a set of dependencies that must be satisfied before a job or group of jobs executes. Each time the Scheduler polls the database for new jobs, these dependencies are checked. You can stipulate that the execution of a job depends on the *completion* of another job or on the *successful completion* of a job. You can also stipulate that the execution of a job depends on the satisfaction of an arbitrary condition, as evaluated by a SQL statement. SQL dependencies should return an integer value, such as SELECT COUNT (*). A returned value of zero indicates that the condition *has not been satisfied*. Values returned other than zero indicate the condition *has been satisfied*.

- Change a job's priority

Agent Administrator allows you to assign a priority of between 1 (the lowest) and 5 (the highest) to a job. To obtain an overall priority for a job, MetaCube Scheduler sums the job's priority with the priority assigned to the user who submitted the job. (User priorities are assigned using MetaCube Secure Warehouse.) MetaCube Scheduler uses this overall priority to determine which jobs runs first when multiple jobs must run concurrently.

- Change a job's time and frequency of execution

Agent Administrator allows you to schedule a job for execution at a specific date and time. If nothing is specified, a job will be scheduled to execute right away. You can also specify that a job run on a recurring basis, such as weekly. By default, jobs run only once.

- Change a job's status

Jobs can have three possible states: “Pending” jobs are waiting to run, “Running” jobs are currently executing, and “Finished” jobs have completed their execution.

Submitting Jobs

Agent Administrator allows you to submit the types of jobs described in the following table. For all of the jobs listed below, you can define the job's dependencies, priority, and time and frequency of execution.

SQL Non-Select	Executes data modifications using SQL statements, such as UPDATE, INSERT, and DELETE. A SQL non-Select job can also perform database maintenance tasks on tables and indexes using statements such as CREATE, ALTER, and DROP.
SQL Select	Runs SQL Select statements that return results stored in a table in the database
Operating System	Runs a specific operating system command. These may be shell scripts or custom executables.
Full Aggregate	Creates new aggregate tables based on a description registered in MetaCube's metadata. Aggregate tables summarize transactional records, thus reducing the calculations necessary to perform a query.
Incremental Aggregate	Creates incremental versions of existing aggregate tables based on changes to a fact table
Post Incremental Aggregates	Incorporates incremental versions of aggregate tables into existing aggregate tables, inserting or replacing rows as necessary. This job also incorporates an incremental fact table into the existing fact table.
Sample	Creates a small table that is a randomly generated and statistically representative subset of a larger fact table

Web Publisher	Creates an HTML-formatted file that holds the results of a query. Note: You can use Agent Administrator to submit a Web Publisher job only when all MetaCube components are installed on the same Windows NT platform. Otherwise, you must submit the job through DOS. Appendix A provides instructions on submitting a Web Publisher job through DOS.
Alert	Sends an email message when a specified SQL statement returns one or more rows
Job Set	Submits a collection of jobs that can consist of any of the types of jobs listed in this table, including another Job Set.

Other Administrative Tasks

Although managing the job queue and submitting jobs are the primary tasks you will perform using Agent Administrator, you can also perform some other important administrative tasks:

- Define settings for MetaCube Scheduler

Agent Administrator allows you to control the MetaCube Scheduler by specifying the following:

- The number of jobs that can run concurrently
- The frequency with which the Scheduler checks for changes to the queue
- The mode of operation for the Scheduler (normal, normal but accepting no new jobs, or stopped)
- The amount of information recorded at the server when jobs are processed

- **Define parameters**

Agent Administrator allows you to define parameters, which are SQL statements that return one value. Parameters allow users of MetaCube applications to include information in a filter that may vary from situation to situation. For example, a parameter might specify a period of time, such as weekly or monthly. Or, a parameter might allow a user to provide input, such as typing the name of a company that is included in a filter.

[Appendix B](#) provides a complete description of parameters.

- **Define system messages**

Agent Administrator allows you to define system messages, which are generally used to inform users about recent changes to the database. These messages are accessible through MetaCube Explorer, Web Explorer, MetaCube for Excel, and MetaCube's application programming interface.



Secure Warehouse

About MetaCube Secure Warehouse	5-3
The Secure Warehouse User Interface	5-4
Active and Inactive Users	5-5
Users and Their Connection Information	5-6
The Process of Using Secure Warehouse	5-7
Who Can Access Data?	5-7
What Data Can the User Access?.	5-8
When Can the User Access Data?	5-8
How Does the User Access Data?	5-9
Granting Access to the Data Warehouse.	5-9
Managing Groups of Users	5-10
Managing Snap-ins.	5-11

T

his chapter introduces MetaCube Secure Warehouse, a tool that gives data warehouse administrators the ability to control user access to DSS Systems.

About MetaCube Secure Warehouse

MetaCube Secure Warehouse gives a data warehouse administrator a graphical tool for controlling access to DSS Systems. With it, you can easily specify which users have access to a DSS System, what information each user can see in the DSS System, and when that user can see it.

Essentially, the administrator accomplishes this by identifying the users who will be managed, giving those users access to DSS Systems and applying mandatory filters to users on a per-DSS-System basis. The mandatory filters limit what information a user can actually see in each DSS System.

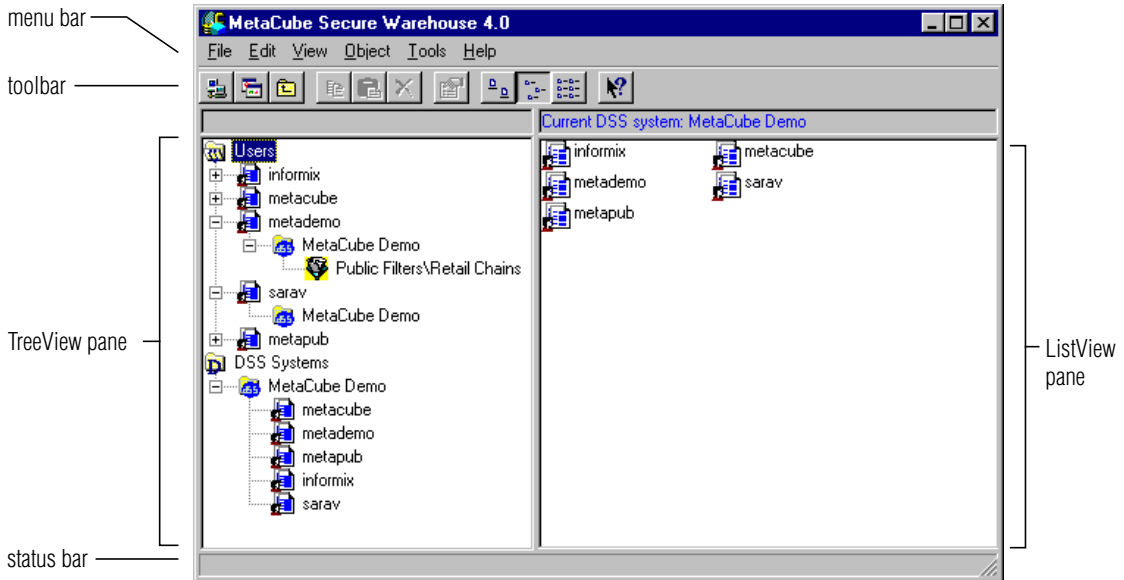
Most of the actions you perform with Secure Warehouse consist of defining information about users. That information is stored in metadata, although some connection information for users is stored in the registry of the computer running the MetaCube engine.

When you use Secure Warehouse, you do not perform a “save” as you would in many computing applications. Instead, almost all of the changes you make are instantly applied to metadata tables.

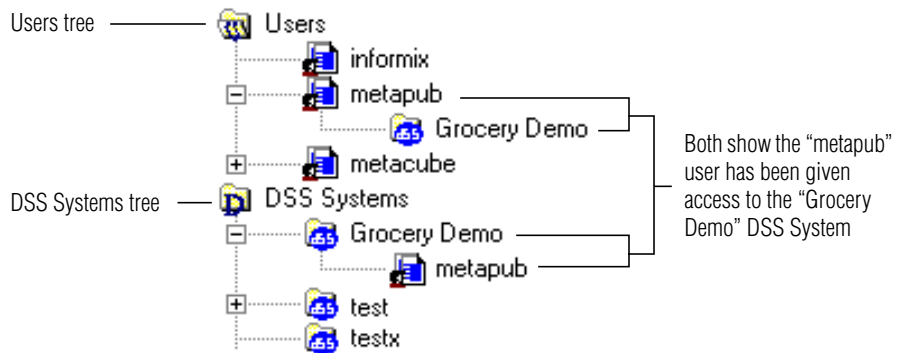
The Secure Warehouse User Interface

The Secure Warehouse user interface consists of two panes:

- The TreeView pane shows two hierarchical tree lists: one presents nodes that represent users, and the other presents nodes that represent DSS Systems available for a current database connection. The highest level node in the user list is called Users. The highest level node in the DSS System list is called DSS Systems.
- The ListView pane shows the contents of the node currently selected in the TreeView.



You can give a user access to a DSS System by manipulating a user or DSS System node or by defining the properties of the user or DSS System. When you assign a DSS System to a user, both trees in the user interface reflect the change. In the Users tree, a node representing the DSS System is displayed beneath the user. Similarly, in the DSS Systems tree, a node representing the user is displayed beneath the corresponding DSS System node. The figure below illustrates this redundant display of information.



The Secure Warehouse user interface provides a standard set of tools, including a menu bar, toolbar, status bar, and a shortcut menu, which is displayed when you right-click while a node is selected. You can also drag and drop or copy and paste nodes on the ListView or TreeView panes.

Active and Inactive Users

The ListView and TreeView panes of Secure Warehouse present two kinds of users:

- Colored user icons represent active users. Active users must be defined in metadata, and the connection information that is stored in the MetaCube registry for them must indicate they are associated with the current connection. For more information on how Secure Warehouse makes this determination, see [“Users and Their Connection Information.”](#)

- Gray user icons represent inactive users. Inactive users are defined in metadata, but the connection information that is stored in the MetaCube registry for them indicates they are *not* associated with the current database connection. Probably they are defined in multiple database connections and their connection information matches one of those other database connections. You cannot use Secure Warehouse to manipulate an inactive user. First you must change the user into an active user by either switching to the connection that is the user's default database connection or by changing the user's default connection information to match the active database connection.

Users and Their Connection Information

For every user managed in Secure Warehouse, the following connection information is stored as part of the registry information for the MetaCube engine to which Secure Warehouse is connected:

- User ID
- Connect string (the string identifying the ODBC data source)
- Metaschema (the prefix identifying the schema or owner of the metadata tables in the database)
- Database type

Typically, the MetaCube engine employs this user-connection information to facilitate connections between Web Explorer users and data sources. But Secure Warehouse also uses this connection information to perform several functions. In particular, Secure Warehouse determines whether a user is active or inactive by comparing a user's connect string, as stored in the registry, with the connect string that you provide when you establish Secure Warehouse's current database connection.

The Process of Using Secure Warehouse

When you use Secure Warehouse, you must log into the MetaCube engine, which establishes a connection between Secure Warehouse and the database. If necessary, you may need to change the current database connection. You may also want to choose a different DSS System.

After establishing the appropriate connections, you are ready to define the following:

- Who (which Informix users) can access data?
- What data can those users access?
- When can those users access data?
- How do those users access data?

Who Can Access Data?

The first step in using Secure Warehouse is identifying which users you intend to manage. You can choose from any user who has been explicitly granted CONNECT, RESOURCE, or DBA privileges to the Informix database. Users granted access implicitly through the PUBLIC keyword are not visible to Secure Warehouse. For more information on granting connection privileges to users, see Informix's system administration documentation.

Note: Although any user granted CONNECT, RESOURCE, or DBA privileges is visible in Secure Warehouse, no user will be able to log into MetaCube and connect to the database until the database administrator has provided that user with a login ID and password *and* the data warehouse administrator has granted the user access privileges in Secure Warehouse.

When you are identifying users to be managed with Secure Warehouse, you can choose individual users from those authorized to access the Informix database, or you can automatically add all the users already defined in metadata. Typically this automatic procedure, called Synchronizing, is most useful for existing MetaCube installations, where users are already defined in metadata but they have not yet been added to Secure Warehouse. After synchronizing all of your users, you will probably add or remove users on a one-by-one basis to keep Secure Warehouse current.

When you add a user to Secure Warehouse, you have the option of defining the user's properties. This means you can specify connection, performance, and security information for that user.

What Data Can the User Access?

You can give a user access to a DSS System by assigning that user to the DSS System or by assigning that DSS System to the user. Either action accomplishes the same goal. Multiple users can be assigned to the same DSS System and multiple DSS Systems can be assigned to a single user.

You can further control a user's access to a DSS System by defining mandatory filters that limit a user's access to certain categories of detailed information. For example, a user may be allowed to access a DSS System for summary sales data and a mandatory filter may be assigned to that user so he can only query sales data from the Eastern Region.

When Can the User Access Data?

Users can have unlimited ability to query a database, either by submitting real-time queries or QueryBack jobs (jobs that are queued on the server and processed in the background). By defining user properties, you can limit the user to QueryBack jobs exclusively. You can also assign a priority level to the user and restrict the hours when the user's jobs are executed. For most users, a restriction on hours will only apply to QueryBack jobs, but if the user is an administrator, the restriction will also apply to any jobs submitted using MetaCube Agent Administrator.

How Does the User Access Data?

By defining a user's properties, you can specify the following aspects of how a user's query is processed:

- A threshold value for slow queries. A warning will be issued for queries that require a computational cost greater than this value. The cost is determined by the fact, aggregate, or sample table that the query will access. In MetaCube Warehouse Manager, each of these data sources is assigned a cost.
- A maximum number of rows that can be retrieved
- PDQ Priority. You can enable PDQ Priority and specify a PDQ Priority value or let the user specify his or her own PDQ Priority

- Data skipping
- Auditing of user's queries. MetaCube Warehouse Optimizer can analyze this data and recommend an aggregation strategy.

Granting Access to the Data Warehouse

Because MetaCube Secure Warehouse and MetaCube Warehouse Manager control the data warehouse, only authorized users should be granted permission to use these applications. Users with this type of permission are called *secure users*.

As the data warehouse administrator, you can access Secure Warehouse and Warehouse Manager in two ways:

- Logging on as the user “metapub.” When a system is initialized, the user “metapub” is by default the only secure user. Any other user may be granted secure user privileges by setting the appropriate property for that user.
- Logging on with your own user ID *after* you have been granted access to Secure Warehouse and Warehouse Manager. You can be granted access by first logging in as “metapub” and then granting your user ID secure user privileges with the user properties sheet. After your user ID has been granted secure user privilege, you can revoke secure user privilege from the user metapub. However, if you revoke metapub's secure user privilege without first granting it to another user, *no one will be able to access Secure Warehouse and Warehouse Manager*. Correcting this situation requires SQL manipulation of the metadata tables, as instructed by MetaCube technical support.

Typically, the first time you establish a database connection for Secure Warehouse, you will log in as metapub. During that first session, you will define the connection properties for your own user ID so you have access to Secure Warehouse and MetaCube Warehouse Manager. Then, in the future you can log on to either application with your own user ID.

Managing Groups of Users

You can manage groups of users by assigning database roles to individuals. A database role is a classification of privileges. For example, you can assign a set of privileges to a certain role, such as *engineer*, and then assign that *engineer* role to many users rather than performing the repetitious work necessary to grant the same set of privileges to many different users.

Assigning a role to a user causes the MetaCube analysis engine, when connecting to the database, to issue a SET ROLE statement on behalf of the user, thus allowing any MetaCube application to take advantage of database-level security.

Note: When you assign roles, you are granting access to tables in the Informix database. Assigning roles does not allow you to control access to DSS Systems for groups of users, nor does it allow you to apply mandatory filters to those groups.

Use the Informix database to assign privileges to a role. Use Secure Warehouse to assign roles to users. For more information on defining a role, refer to Informix's SQL syntax documentation.

Managing Snap-ins

You can use Secure Warehouse to manage the snap-ins available for each instance of the MetaCube analysis engine. Any users logged into a MetaCube product through that MetaCube analysis engine will have access to the snap-ins that are currently enabled for that engine.

To be compatible with Secure Warehouse, snap-ins must be created using MetaCube SDK for Snap-Ins Version 4.0 or later.

Warehouse Optimizer

About MetaCube Warehouse Optimizer	6-3
Before You Begin	6-4
Setting Fact Table Costs in Warehouse Manager	6-4
Configuring Memory for Warehouse Optimizer	6-5
Validating Existing Aggregate Tables	6-6
Creating a Sample Table	6-6
How to Calculate Combination Totals	6-7
The Warehouse Optimizer User Interface	6-9
Using Warehouse Optimizer: An Overview	6-10
Connecting to the Database	6-10
Configuring the Analysis.	6-11
Choosing a Data Source	6-11
Selecting Analysis Type	6-11
Setting the Number of Dimensions in Candidate Aggregates	6-12
Generating Costs	6-13
Reducing Cost Generation Time	6-13
Should You Estimate Costs?	6-13
How Warehouse Optimizer Generates Costs.	6-14
Recommending Aggregates	6-16
Audit Data	6-16
How Many Aggregates to Recommend	6-17
Viewing Recommended Aggregate Details	6-18
Aggregate Recommendations Report	6-18
Cost Graph	6-19

Registering Aggregates	6-19
Optimizer Performance Log	6-20
Updating Recommendations	6-21

T

his chapter introduces MetaCube Warehouse Optimizer and describes the functions it performs.

For step-by-step instructions explaining how to use Warehouse Optimizer, please refer to Warehouse Optimizer's online help system.

About MetaCube Warehouse Optimizer

MetaCube Warehouse Optimizer is a graphical tool designed to help data warehouse administrators decide how best to use aggregation to improve query performance.

When processing queries, the MetaCube analysis engine calculates the most efficient path for retrieving data. Many queries request summary data. Normally, queries requesting such summary-level data require the database to perform the sum while processing the query. You can speed MetaCube's processing of these queries by creating aggregate tables, which are special tables containing nothing but summaries of more detailed data. The MetaCube analysis engine will automatically access aggregate tables if they exist.

In the typical data warehouse, where query performance is critical, you should probably create at least some aggregate tables. When developing aggregate tables, your objective is to create summary tables that provide the greatest performance benefit at an acceptable cost in terms of additional storage and maintenance. Data density and usage patterns are the primary factors in determining the optimal aggregates.

MetaCube Warehouse Optimizer can analyze data density and usage patterns and recommend aggregate tables. To perform this analysis, Warehouse Optimizer queries your data model to determine the actual density of data along different combinations of dimensions and incorporates into its analysis a weighted audit of users' queries. To help you make an informed decision about aggregate use, Warehouse Optimizer can display information on the aggregates it recommends. Once you choose aggregates, Warehouse Optimizer can create their metadata descriptions.

Over time, use of a data warehouse is the best indicator of which aggregate tables to create and maintain. Before the data warehouse is in use, however, you need to make some assumptions about the queries users will execute most frequently. At a later date, you can update your aggregates to reflect the needs of your users more accurately.

Optimizing aggregate tables can be a time consuming process, but updates are rarely needed and once the optimization is complete, users can greatly improve their query performance.

For more information on aggregates, refer to the [MetaCube Warehouse Manager's Guide](#).

Before You Begin

Before you begin using Warehouse Optimizer, you should ensure its accurate and efficient performance by doing the following tasks.

Setting Fact Table Costs in Warehouse Manager

Warehouse Optimizer bases its aggregate recommendations on query costs obtained from the MetaCube analysis engine and relative values Warehouse Optimizer assigns to possible aggregates. The MetaCube analysis engine obtains query costs from the fact table costs set in Warehouse Manager. Although fact table costs can be set to a relative value for other MetaCube products, in order for Warehouse Optimizer to produce accurate recommendations, a fact table's cost must be set to its actual row count.

Configuring Memory for Warehouse Optimizer

The aggregate recommendation stage of Warehouse Optimizer is a memory-intensive process. Warehouse Optimizer reads all possible combinations of dimension elements into memory. Each combination represents a potential user query. Warehouse Optimizer requires approximately 18 to 20MB of memory, regardless of how many combinations it processes. Warehouse Optimizer also requires at least 10KB of memory for storage. (Warehouse Optimizer stores a minimum of 100 combinations in memory.) When configuring memory, consider the trade-off between less memory and more processing time, the size of your analysis, and the memory available on your computer.

Each combination of dimension elements requires roughly 100 bytes of memory, varying slightly with the number of dimensions and combinations included in your analysis. Warehouse Optimizer calculates the number of combinations it can read into memory based on the amount of memory you allocate in the `metacube.ini` file. (Using the `metacube.ini` file to allocate memory is described in [Appendix C](#).) If no value is set, Warehouse Optimizer reads 100,000 combinations into memory by default.

If you do not allocate enough memory for all combinations to be read into memory at once, Warehouse Optimizer creates a cache file, called `_whocach_cc.mwo`. This file is located in the same directory as the Warehouse Optimizer analysis file and uses only 40 bytes of disk space per combination. Warehouse Optimizer swaps combinations into memory from this cache file. For example, if you allocate 10KB of memory, and you are processing 1000 combinations, approximately 100 combinations are swapped into memory at a time, while the rest are stored in the cache file until needed. Once the recommendation stage is complete, Warehouse Optimizer cleans up and removes the cache file, unless the application terminates abnormally, in which case the cache file will be cleaned the next time you initiate the recommendation stage. With less memory swapping, Warehouse Optimizer recommends aggregates faster.

Validating Existing Aggregate Tables

If you are running Warehouse Optimizer to validate existing aggregate tables, you must uncheck the **Valid** check box in MetaCube Warehouse Manager for each aggregate. If you do not do this, Warehouse Optimizer recommends new aggregates based on how the data warehouse performs when it uses existing aggregates.

Creating a Sample Table

You may want to create a sample table on which to run your analysis, thereby reducing query time and speeding up computation of storage costs. If the number of rows in your fact table is in the millions, you should use a sample table. To create a sample table, first use MetaCube Warehouse Manager to produce a metadata description for the sample table, and then use MetaCube Agent Administrator to submit a Sample job. Informix recommends that any sample table you create should be at least twice as large as your largest dimension table.

The [MetaCube Warehouse Manager's Guide](#) describes how to create metadata for sample tables. [Chapter 4, "Agent Administrator,"](#) in this manual describes the use of MetaCube Agent Administrator, and on-line help for Agent Administrator describes how to submit a Sample Job.

How to Calculate Combination Totals

There are two combination totals referred to in this chapter—the number of *base combinations*—all combinations consisting of one or two elements—and the total number of combinations.

To calculate the number of base combinations:

1. Add the total number of dimension elements in each dimension together.

Use the following description of a DSS System as an example:

- Dimension A has 4 dimension elements.
- Dimension B has 2 dimension elements.
- Dimension C has 3 dimension elements.

Using this example you get the following calculation for step 1:

$$4 + 2 + 3 = 9$$

2. Multiply the total number of dimension elements contained in a dimension by the total number of dimension elements contained in any other dimension. Do this for all unique two-dimension combinations and add the totals together.

The calculation based on the above sample would be:

$$(4 \times 2) + (4 \times 3) + (2 \times 3) = 26$$

3. Add the total from step 1 and the total from step 2.

For example:

$$9 + 26 = 35$$

In the example DSS System there are 35 base combinations.

To calculate the total number of combinations in a DSS System:

1. Add one to the total number of dimension elements in each dimension.

Using the same sample DSS System as above, you get:

$$4 + 1 = 5$$

$$2 + 1 = 3$$

$$3 + 1 = 4$$

2. Multiply these totals together.

Using the totals from step 1:

$$5 \times 3 \times 4 = 60$$

3. Subtract 1 from the total.

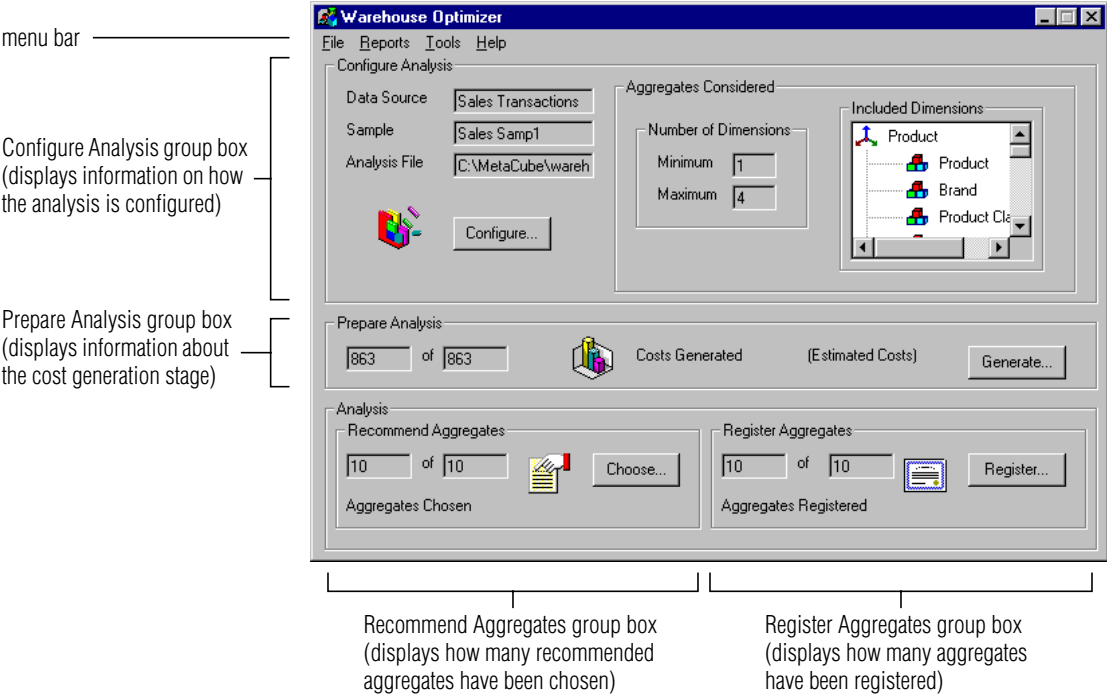
Using the total from step 2:

$$60 - 1 = 59$$

In the example DSS System there are a total of 59 combinations.

The Warehouse Optimizer User Interface

MetaCube Warehouse Optimizer displays a single window in which you step through the stages of using Warehouse Optimizer—analyzing the data warehouse, selecting aggregates, and creating metadata for the aggregate tables. You must manually initiate each of these stages by clicking the corresponding button in the main window. Group boxes and command buttons remain dimmed until they can be used.



Using Warehouse Optimizer: An Overview

After connecting Warehouse Optimizer to the database, the process of using it consists of four main stages:

- Configuring the analysis
- Calculating the storage and query cost for combinations of dimension elements
- Recommending aggregates
- Registering aggregates (only users authorized in MetaCube Secure Warehouse can perform this stage)

You may perform the entire process at one time, or you may perform different stages at different times. Warehouse Optimizer stores all the results of its analysis in a local analysis file, which you can reopen at any time to continue, complete, or change the analysis.

You must complete one stage of analysis before moving on to the next, except when recommending aggregates. In that case you may move on to registering aggregates after only one aggregate has been recommended.

Connecting to the Database

To use Warehouse Optimizer, you must first log in to MetaCube with a user name and password. This connects Warehouse Optimizer to the database that houses your data warehouse.

If you have never previously connected to MetaCube, you need to define a database connection for Warehouse Optimizer before attempting to log in. Choosing **Options** on the **Tools** menu displays the Preferences dialog box, which allows you to define database connections.

Configuring the Analysis

There are several concerns when configuring your analysis:

- Choosing a Data Source
- Selecting Analysis Type
- Setting the Number of Dimensions in Aggregates (used for partial analysis only)

Choosing a Data Source

You can run Warehouse Optimizer against a fact table or a sample table, if one exists. Running your analysis against a sample table can decrease processing time without much impact on the accuracy of the results. If the number of rows in your fact table is in the millions, you should use a sample table. For more information on using sample tables in Warehouse Optimizer refer to [“Creating a Sample Table” on page 6](#).

Selecting Analysis Type

MetaCube Warehouse Optimizer can identify candidates for aggregation by operating in two modes:

- **Full analysis** considers all possible combinations of the dimension elements that are referenced in the fact table’s metadata.
- **Partial analysis** allows you to select specific dimension elements to exclude from consideration and to choose the maximum and minimum number of dimensions to include in each candidate aggregate.

Typically, you should run a full analysis. Partial analysis may be useful when a fact table contains very detailed data for a given dimension element but users rarely query data at that level. In such a case, you might simplify the aggregate analysis by excluding these seldom-used dimension elements.

You might also use partial analysis when you know users query only at a particular level or set of levels. For example, if users often query by the dimension element *Day* along with any of the *Product* dimension elements, you might restrict analysis to only combinations that include these elements. These combinations are the only ones considered as potential aggregate combinations or *candidate aggregates*.

Setting the Number of Dimensions in Candidate Aggregates

When performing a partial analysis, Informix suggests you also set the maximum and minimum number of dimensions included in each candidate aggregate. Aggregates with too many dimensions are frequently too large to provide a performance benefit, and aggregates with too few dimensions are rarely useful.

When you eliminate dimensions or dimension elements for a partial analysis, they will not be considered in candidate aggregates (saving time and resources during the storage cost calculation and aggregate recommendation stages). However, this does not reduce the number of combinations that Warehouse Optimizer generates during analysis. Although dimension elements can be eliminated from consideration in candidate aggregates, their elimination does not exclude them from being included in queries a user might submit. To calculate the average cost of a user query against the fact table and any existing aggregate tables, Warehouse Optimizer generates *all* possible query combinations and their associated query costs. Then Warehouse Optimizer compares the average query cost for the existing data warehouse with the average query cost for the data warehouse assuming that the candidate aggregates are created. Based on this comparison, Warehouse Optimizer recommends aggregates that yield the lowest average query cost.

If your DSS System consists of a large number of dimensions and dimension elements that result in more than 500,000 combinations, Informix recommends that you create another DSS System consisting of a subset of the original DSS System and run Warehouse Optimizer on the subset. The subset DSS System should include the dimensions and dimension elements that are accessed frequently in user queries.

Once the analysis has been configured, you need not rerun this stage unless you want to change the configuration or the metadata in your database has changed.

Generating Costs

In this stage of analysis, Warehouse Optimizer determines:

- the cost of each possible query against the existing fact table and any existing aggregate tables.
- the cost of storing each candidate aggregate table.

Reducing Cost Generation Time

Although generating costs can be a lengthy process, you can use any of the following strategies to reduce computation time:

- Estimate costs (discussed below)
- Run a partial analysis, eliminating some dimensions or dimension elements to reduce the number of queries to the database
- Run your analysis against a sample table, reducing the number of rows to process in your queries and therefore reducing query time

Important: *Informix recommends that you estimate costs when using a sample table—estimated storage costs for combinations with more than two elements are generally more accurate than those costs computed by submitting queries against the sample tables.*

Should You Estimate Costs?

If you choose to estimate costs, Warehouse Optimizer submits only queries for the aggregate candidates made of base combinations—all combinations consisting of one or two elements—and estimates costs for all remaining combinations. If you have more than 200 base combinations and a fact table with more than 100,000 rows, generating actual costs can take a day or more. However, Warehouse Optimizer estimates the remaining costs at a rate of 100,000 combinations per hour.

If you do not estimate costs, Warehouse Optimizer submits queries for all candidate aggregate combinations. The time needed for a cost computation varies, depending on the load on the database server, the size of the dimension tables, and the computational speed and power of your system. However, to give you an idea of the time involved, if the fact table is large (for example, over 100,000 rows) and the number of combinations is also large (more than 10,000), the cost-computation stage can take days.

Important: Informix recommends that you estimate costs if your analysis includes more than 10,000 combinations.

How Warehouse Optimizer Generates Costs

Eliminating combinations in partial analysis removes them from consideration as possible aggregates but does not eliminate them from being possible queries that a user might submit. Consequently, Warehouse Optimizer computes the query cost for all combinations but computes the storage cost for candidate aggregates only.

Warehouse Optimizer quickly computes query cost, determining whether the query can be executed against an existing aggregate or the original fact table. The cost is the total number of rows processed for the query.

The time needed to compute storage costs can vary greatly depending on the number of combinations of candidate aggregates for which Warehouse Optimizer computes *actual* storage costs. To determine the actual storage cost of candidate aggregates, Warehouse Optimizer submits a GROUP BY query corresponding to each candidate aggregate table. Warehouse Optimizer creates a view, called **tempview**, to store the results while counting the number of rows returned.

Because the computation of storage costs can be a lengthy process and Warehouse Optimizer must remain connected to the database throughout, failures can occur during this stage. During processing, any failure in executing a query to the database returns the error:

```
Runtime Error '3146' ODBC call failed
```

Listed below are some possible causes for this error:

- Improper specification of the metadata might cause references to invalid or nonexistent tables in the SQL statement generated by the MetaCube analysis engine in creating the aggregate views. Use MetaCube Warehouse Manager to verify the accuracy of the metadata for the fact table, dimensions, and sample table for the data source you are using.
- Connection to the database server failed. If this occurs, rerun cost generation. Warehouse Optimizer continues running where it left off when the error occurred.
- The computer where the database server is located was down or inaccessible for any reason.
- More than one Warehouse Optimizer was running against the same data source. Problems that might occur include:
 - Warehouse Optimizer creates each view with the same name, **tempview**, for each storage cost computation. Therefore, another Warehouse Optimizer running on the same data source cannot create this view.
 - One Warehouse Optimizer process might drop the tempview created by a second Warehouse Optimizer process—the first process creates the view and the second process deletes the view.
- If an earlier Warehouse Optimizer process failed before dropping the view it created, orphan processes maintaining a lock on the view may be left on the database server. Consequently, an attempt to drop the view fails. Using the operating system, kill any leftover processes before starting the cost generation stage.
- Insufficient temporary space in the database to run the query associated with creating each view.
- Insufficient shared memory configured on the system running the database server. This will return an Informix error:
`Informix Error '208' Memory Allocation failed`
For more information on configuring shared memory, refer to Informix's database administration documentation.
- Incorrect permissions to create or drop a view or to execute a query against the selected data source. Make sure that the Warehouse Optimizer user has sufficient permissions.

Note: *If Warehouse Optimizer seems to be taking too long to process a single aggregate combination on fact and dimension tables that are not large, you can stop and restart. Warehouse Optimizer resumes analysis where it left off. Lengthy processing might be the result of a query waiting for resources to be released in the database. However, if dimension tables are large (even though the fact table is small), the joins required to run the query might take some time to complete.*

Once costs are generated, you need not rerun this stage unless you change the analysis configuration or the data density of your database has changed. If you change the analysis configuration, Warehouse Optimizer automatically updates storage costs. You can rerun Warehouse Optimizer to regenerate aggregate recommendations at any time, such as after MetaCube has collected user query audit data.

Recommending Aggregates

Recommending aggregates consists of the following tasks:

- Analyzing query audit data, if you choose to use this information
- Reading the combinations generated in previous stages into memory. If not enough memory is configured to hold all combinations, then the combinations are first read onto disk, then swapped into memory a portion at a time.
- Choosing the best aggregates

You must decide two things for aggregate recommendation:

- Whether to use audit data
- How many aggregates to recommend

Audit Data

Audit Data is information about user queries, such as which data is requested, who is requesting it, how long the query takes to process, how many rows are retrieved, and other similar criteria. You can view query audit data in a columnar text format by running the **Query Audit Detail** report. This data, if available, can be very useful in recommending aggregates that are specific to the needs of your users.

Important: You must specify for each user whether you want MetaCube to collect audit data. To do this, use the **Audit Queries of this User** check box in user properties in MetaCube Secure Warehouse.

If you specify that you want to use audit data, you must indicate the weight that audit data has in the analysis process. The weight ranges from one to ten, with each number representing a percentage value.

Audit Weight	Percentage
1	10%
2	20%
3	30%
4	40%
5	50%
6	60%
7	70%
8	80%
9	90%
10	100%

The percentage you enter reflects your confidence in the audit data. If the audit data accurately reflects full use of the data warehouse by all users, you should specify 10, for 100% weighting of the audit data.

How Many Aggregates to Recommend

The main concern in deciding how many aggregates you want Warehouse Optimizer to recommend is time. The time required to analyze audit data is proportional to the amount of audit information available. As a guideline, assume that audit data containing 2,000 items for 6,000 combinations takes about ten minutes.

The time required for reading combinations into memory or the cache file is approximately five to ten minutes for one million combinations. Swapping combinations from the cache file into memory will slow the next stage, choosing aggregates.

The time required to choose aggregates is proportional to (mn^2) where:

- m is the number of aggregates to recommend
- n is the number of combinations included as candidate aggregates

Recommending 10 aggregates from 50,000 combinations takes about 14 hours of processing time if you have configured enough memory for all combinations to be read directly into the memory.

Viewing Recommended Aggregate Details

You can view detailed information on the recommended aggregates to aid you choose aggregates to register. You can view this information in the following forms:

- Aggregate Recommendations Report
- Cost Graph
- Aggregate Information in the Register Aggregates dialog box

Register the aggregates that will give you the most improvement in query performance with the least storage cost. In other words, register the aggregates at the point of diminishing returns.

Aggregate Recommendations Report

When Warehouse Optimizer completes its recommendations, you are asked if you want to view the recommendations report now. To view this data later, select **Aggregate Recommendations** from the **Report** menu. The report lists the following information about each recommended aggregate.

Item	Meaning
Rank	The ranked order of the aggregate table
Agg Size (Rows)	The number of rows in the aggregate table
Avg Query Cost	The average cost to run a query if the aggregate table and all higher ranking recommended aggregate tables are created
Dimension Elements	The dimension elements included in the aggregate table

Cost Graph

The Cost Graph displays the aggregate recommendation data in chart format. To access the graph, select **Cost Graph** from the **Report** menu. The graph shows:

- the average cost to run a query
- the cumulative storage costs (in number of rows)

The Cost Graph is useful for presenting query cost and storage cost information, and quickly finding the point of diminishing returns in a graphical representation. Using the chart and the data generated by Warehouse Optimizer, you can determine the proper balance of storage required for a set of recommended aggregates and the improvement in query performance that the aggregates can provide for your data warehouse.

Registering Aggregates

Once you have found the point of diminishing returns, you can decide how many aggregates to register and change the names of the aggregates to be more meaningful to you, if desired.

Only users who have been granted permission to the data warehouse administrative tools in MetaCube Secure Warehouse (secure users, in other words) can register aggregates. If you are not a secure user, save your analysis file and contact your data warehouse administrator, who can register aggregates using Warehouse Optimizer or MetaCube Warehouse Manager.

In the registering aggregates stage, use the information provided about recommended aggregates to help you decide which of the aggregates you want to register in metadata. Pay close attention to the query and storage costs displayed in the Aggregate Information group box. The query cost of each aggregate is the average cost of a query, assuming that the selected aggregate and all higher ranking aggregates are built. The storage cost is the cumulative cost of storing the selected aggregate and all higher ranking aggregates.

When registering aggregates, you can specify how many of the highest ranking recommended aggregates you want to register. You cannot select particular aggregates from a list of recommended aggregates, since the query and storage costs listed for each aggregate are based on all higher ranking aggregates being built. For example, if Warehouse Optimizer recommends five aggregates, you can choose to register two, and Warehouse Optimizer will register the top two. You cannot pick particular aggregates from that list of five and register them.

Registering recommended aggregates does not take long, because the only processing necessary is the creation of metadata for the aggregate.

Once you register aggregates, the check boxes next to the aggregates in the list box show which aggregates have been registered in the current session. You may reselect aggregates from the current recommendations at any time. If you choose to register more aggregates than were previously registered, only those aggregates that have not been registered before will be registered. If, however, you specify fewer aggregates than were previously registered, Warehouse Optimizer will ask if you want to register the aggregates under new names. Warehouse Optimizer generates metadata for the aggregates using a new name for each aggregate table. To avoid duplicate aggregate definitions, use Warehouse Manager to delete metadata for previously registered aggregate tables.

Once you register the aggregates in metadata, you must use MetaCube Agent Administrator to run the jobs that create aggregate tables.

Optimizer Performance Log

Warehouse Optimizer can create a performance log, which you can use to view information on the number of combinations processed and the processing time necessary for the cost generation and aggregate recommendation stages. The performance log, called **who.log**, is located in the MetaCube installation directory. It can be turned on and off by setting a value in the metacube.ini file, as described in [Appendix C](#).

The performance log tracks the following information:

Stage	Data Tracked
Generating Costs	If estimating costs: ■ Number of combinations for which actual costs were generated—number of base combinations ■ Number of combinations for which estimated costs were generated ■ Amount of time to generate actual costs ■ Amount of time to generate estimated costs
	If not estimating costs: ■ Number of combinations for which costs were generated ■ Amount of time needed to generate costs
Recommending Aggregates	■ Total number of combinations processed ■ Number of combinations stored in memory at a time ■ Start and finish time for each recommendation

Important: You can turn on and off performance logging in the **metacube.ini** file. For step-by-step instructions, refer to the online help system.

Updating Recommendations

You will want to update the recommendations made by Warehouse Optimizer from time to time. For instance, after collecting user query audit data, rerun Warehouse Optimizer to update the recommendations based on the data stored in the audit tables. You can update recommendations without reconfiguring the analysis or regenerating costs unless you select a different data source or new aggregate tables have been created. Updating your aggregate table recommendations with Warehouse Optimizer is an important step in keeping your data warehouse running as efficiently as possible.

Submitting a Web Publisher Job in DOS

MetaCube Web Publisher is a utility for creating HTML-formatted files that hold the results of a query. These files can be viewed using a web browser. This allows users who do not have MetaCube software installed on their PCs to view reports generated through queries to the data warehouse using a local intranet web server. The MetaCube Web Publisher can also create a table of contents HTML page containing links to all reports generated in HTML format. Thus, from a single page, users can easily locate and view query results.

At the DOS prompt, you can issue a command that will cause MetaCube to generate HTML pages using the Web Publisher software that resides on a Windows NT PC. Executing the DOS command does not involve using MetaCube Agent Administrator or MetaCube agents. It performs the query directly and returns HTML-formatted pages.

To schedule a recurring DOS-based Web Publisher job, use the Windows NT or Windows 95 scheduler to run the job periodically.

The following table lists the components of the command line call to the MetaCube Web Publisher, that is, the “metapubl” utility.

Component	Required/Optional	Description
metapubl	Required	The MetaCube Web Publisher utility; begins every DOS command.
/user=	Optional	User login ID; if not provided, MetaCube uses information on the most recently accessed database as contained in the metacube.ini file.
/pass=	Required	User password; used to log on to the database.
/conn=	Optional (not recommended)	The ODBC data source for connecting to the database; if not provided, MetaCube uses the most recently accessed database as contained in the metacube.ini file. If provided, and it differs from the entry in metacube.ini, the metacube.ini file is rewritten with the value specified here.
/dss=	Optional	The DSS System name; if not provided, MetaCube uses information on the most recently accessed database as contained in the metacube.ini file.
/schema=	Optional	Provides a metaschema prefix; defaults to "metacube." (that is metacube followed by a period)
/outp=	Required	Name of the HTML file to produce
/head=	Optional	Name of the HTML or text file to add to the beginning of the HTML page generated. The filename must end with the extension .htm.
/foot=	Optional	Name of the HTML or text file to add to the end of the HTML page generated.
/query= or /jobid=	Required	Name of the query or the QueryBack job ID number that will produce the results to be formatted into an HTML page.

Component	Required/Optional	Description
/type=	Optional	Type of output to produce; defaults to report. The argument can equal one of the following: ■ report ■ bar ■ 3dbar (a three-dimensional bar chart) ■ percent ■ line ■ area ■ stacked
/toc=	Optional	Name of the table of contents file to be created or appended or the file to which this HTML page should be linked.
/log[=]	Optional	Turns on logging, which tracks the progress of creating an HTML page using Web Publisher. The default name of the log file is metapubl.log. To use another name for the log file, use the optional = sign and provide the filename.
/debug	Optional	Turns on debug mode, which tracks detailed information on the progress of creating an HTML page using Web Publisher. The name of the log file is always metapubl.log.

This sample command illustrates the use of the required components:

```
metapubl /pass=informix /outp=mo_sales.htm "/query=Monthly Sales"
```

If an argument contains any embedded spaces, such as `"/query=Monthly Sales"`, the entire argument must be enclosed in double quotes. Optionally, any argument can be enclosed in double quotes.

To specify the full pathname for a query that is not located in the ROOT directory, use commas to separate sub-directory names, as in the following example:

```
"/query=root,myfolder,query_folder,query_name"
```

To put the output file in a location other than the current directory, type the full pathname, as in the following example:

```
"/outp=c:\root\myfolder\output_folder\output_filename"
```

Understanding Parameters

A parameter is defined by a name and an SQL statement. When a parameter is encountered in a query's specification, the SQL statement is executed to return a result of a single value. That value is then substituted for the parameter before the query itself is executed.

Parameters are used in filters applied to queries. For example, using the MetaCube demonstration database, a filter might be defined as follows:

```
Company = <<Enter Company Name>>
```

When applied to a query, this filter:

- causes the **Query Parameters** dialog box to display, so that the user can type in the name of the company on which to filter
- limits data returned from the data warehouse to only data for the company entered by the user in the **Query Parameters** dialog box

Suppose, for example, the user types "Electrotron Inc." in the **Query Parameters** dialog box. In that case, the <<Enter Company Name>> parameter causes the following SQL to be generated by the MetaCube analysis engine:

```
WHERE metademo.companies.company_name =  
'Electrotron Inc.'
```

This WHERE statement limits the data returned by the query to data for the Electrotron Inc. company only.

The relative time filters provided by MetaCube Explorer, Web Explorer, and MetaCube for Excel are actually parameter names. When a MetaCube relative time filter is applied to a query, the MetaCube analysis engine generates a WHERE clause that returns a code for the time period defined by the filter. For example, in the MetaCube demonstration database, a Current Week time filter, defined as:

```
Fiscal Week In (<<Current Period>>)
```

when applied to a query that retrieves data from the Sales Transactions fact table, generates the following WHERE clause:

```
WHERE metademo.sales_transactions.fiscal_week_number In (65)
```

The number 65 is the result of executing the Fiscal Week parameter's SQL statement; that statement returns the code that indicates, in the time dimension table, the row that contains the data of the current fiscal week. When the query, constrained by the Fiscal Week relative time filter, executes, only data for the current fiscal week is returned. The user is not prompted for relative time parameters.

Parameters in QueryBack Jobs

It is common for jobs to be submitted well before they are actually executed by MetaCube Scheduler. This happens most frequently when a job is set to recur automatically. For example, a job may be submitted in January that will execute twelve times throughout the year. In such cases, any parameter values in the query must be dynamically substituted at run-time.

For example, suppose a MetaCube user submits a QueryBack job that generates a sales report every week. The query may contain a relative time filter that defines the "current week." To produce the report, the MetaCube QueryBack agent must look up the current week information, adjust the SQL required to generate the report, and then perform the query. A parameter is the query definition's dynamic component, changing each time the QueryBack process executes the query.

When submitting queries to MetaCube QueryBack, users can

- specify a single future date and time at which QueryBack should execute the query

- request that QueryBack process the query at recurring intervals in the future, such as every day or every month.

When submitting a relative time query at a future date or at recurring intervals, QueryBack must be able to update the query to retrieve the most recent data.

To support this QueryBack functionality, you must, using MetaCube Agent Administrator, specify the definition of a parameter for every dimension element of the *Time* dimension. This allows MetaCube QueryBack to determine the most recent time period for that dimension element when the query is run.

The definition of a parameter is an SQL statement that returns one and only one value. In the case of the relative time parameter, the value returned is a code that flags the current (or most recent) time period relevant to the query. For example, in the MetaCube demonstration database, the SQL statement that returns the current fiscal week code is:

```
SELECT fiscal_week_number
FROM metademo.time_dimension
WHERE current_period = "Y"
```

The result of executing this SQL statement is the return of the code value of 65 which, in the demonstration database, flags the most recent (or current) week in the database. Refer to the [MetaCube Warehouse Manager's Guide](#) for a discussion of relative time and the current period flag.

If you examine the time dimension table for the MetaCube demonstration database, you will notice that only one row is flagged in the *current_period* column with a "Y"; the dates in that row are the most current in the database. MetaCube QueryBack substitutes the relevant date for the current query and executes it, retrieving data for that time period.

The general syntax for SQL statements to retrieve the most recent time periods is:

```
SELECT dimension_element_column
FROM time_dimension_table
WHERE current_period_column = "Y"
```

Using this statement as a model, you can create the required SQL statements for every time dimension element in your own data warehouse.

User-Defined Parameters

You may use MetaCube Agent Administrator to pre-define parameters that Explorer, Web Explorer, or MetaCube for Excel users can then incorporate into parameterized filters in the same way that they incorporate relative time filters. Pre-defining parameters for users

- makes it easier for them to create their own parameterized filters
- assures that queries containing parameters can be submitted to QueryBack and execute successfully.

As an illustration, you might define a parameter named First Brand. The SQL executed for this parameter, shown here:

```
SELECT brand_name FROM brands WHERE brand_code IN (SELECT  
MIN(brand_code) from brands)
```

retrieves data only for the first brand listed in the Brands table.

An Explorer or MetaCube for Excel user could define a filter, using this parameter:

```
Brand = <<First Brand>>
```

This usage of a user-defined parameter produces no **Query Parameters** dialog box for user input when the query is executed. However, the query will retrieve only information for the single brand—the first brand in the Brands table. In addition, because the parameter is defined through Agent Administrator, any query using this parameter will execute successfully using QueryBack.

The parameters you define using MetaCube Agent Administrator can be used for a wide variety of tasks to ultimately retrieve the value used for filtering a MetaCube query. Preparing useful parameters for MetaCube users can increase their ability to retrieve meaningful reports from the data warehouse.

Parameters in SQL Jobs

MetaCube supports parameters in SQL jobs as well as QueryBack jobs. To submit an SQL job that is dynamically altered at runtime, include the parameter name enclosed in double angle brackets

<< >>

For example, suppose you wanted to run an UPDATE statement every week. The UPDATE statement changes the label on a report to reflect the name of the current week, as the week is defined in the MetaCube system. Assuming that `fiscal_week` is a parameter name, the SQL statement can then be submitted with the embedded parameter for fiscal week: <<fiscal_week>>

```
UPDATE report_labels SET btitle =  
    SELECT week_description FROM time_dimension  
    WHERE week_code = <<fiscal_week>>
```

When the SQL statement executes, a numeric value is substituted for <<fiscal_week>>.

Parameters in SQL Jobs

--

The metacube.ini File

The metacube.ini initialization file, located in the Windows directory of any PC running any MetaCube application except MetaCube Web Explorer, stores some MetaCube preferences and configuration information. You can modify these values by editing the file using a text editor such as Windows Notepad.

The following table explains the entries in the metacube.ini file. The parameters listed in metacube.ini are a subset of all MetaCube preferences. Most preferences in MetaCube applications are set within the application's software, and they cannot be modified using metacube.ini.

The metacube.ini file is divided into sections, with each section heading denoted by square brackets, such as [Engine] or [Agents]. Within each section, parameters can appear in any order. Parameter names are not case sensitive.

[General]	
Configuration	The currently active configuration by which MetaCube connects to the Informix database.
PassThroughDriver	The full pathname to the database's ODBC driver.
OptimizerMemory	Allocates memory (in kilobytes) for use during the aggregate recommendation stage of MetaCube Warehouse Optimizer.
LogOptimizerPerf	Indicates that the performance log for MetaCube Warehouse Optimizer is turned on. (0=off; 1=on)
InstallDir	The directory in which MetaCube software is installed.
Config1 Config2 Config3 etc.	Other saved configurations. All saved configuration names appear as a choice in the Preferences dialog box for all MetaCube applications.
[config_name]	Each configuration listed in the [General] section has its own section of the configuration file. The section name is the configuration name.
IncrSchema	This value is provided when you configure in the MetaCube Agent Administrator. Indicates the prefix to be used when MetaCube Aggregator creates incremental aggregate files.
Login	The last user ID used to login.
MetamodelSchema	Identifies the owner/schema of the metadata tables.
DSSSystemName	The name of the DSS System accessed at login.
ConnectDatabase	The database vendor. (2=MS Access; 5=Informix database)
ConnectionString	The name of the ODBC Data Source.
MetamodelFile	The name of the file that stores a local copy of the metadata for the DSS System.
PDQPriority	When PDQ Priority is enabled, the number indicating the priority for PDQ queries.
DataSkip	Indicates whether to use the Informix Data Skip feature for queries. (0=don't use Data Skip; 1=use Data Skip; 2=use Informix server default)

[Explorer]	
MainWindowState	Changed dynamically by Explorer based on window state (maximized, sized, or minimized) when application is closed.
[Engine]	
EnabledExtensions	Name(s) of all Explorer snap-in modules currently enabled.
DisabledExtensions	Name(s) of snap-in modules previously enabled, now disabled.
[Agents]	
CenterTo	Indicates whether dialog boxes center in relation to the monitor or the Agent Administrator main window.
AutoUpdateOnSubmit	Indicates that Agent Administrator updates the job queue automatically when a new job is submitted.
DependOn	Indicates that the Dependencies dialog box displays all jobs or only jobs that are pending. (AllJob or Pending)
[Excel]	
UseSampling	Indicates that sampling is being used.
Accuracy	A numeric value between 1 and 100 indicating the sample table that the MetaCube analysis engine should access to process a query. A higher Accuracy value indicates a larger Sample table should be accessed if one is available.
SlowQueryWarning	Indicates the threshold value above which the Slow Query Warning is displayed for queries.
MaxTotalFetches	The maximum number of rows that MetaCube will retrieve for a single SQL statement.
[Fonts]	
Name	Name of screen font for variable text in Agent Administrator.
Size	Size of the screen font for variable text in Agent Administrator.
Bold	Indicates whether variable text in Agent Administrator is bold. (False=not bold; True=bold)
Italic	Indicates whether variable text in Agent Administrator is italic. (False=not italic; True=italic)
Underline	Indicates whether variable text in Agent Administrator is underlined. (False=no underline; True=underline).

Index

A

- active users
 - in Secure Warehouse 5-5
- Agent Administrator 4-1 to 4-11
- administrative tasks 4-11
- controlling MetaCube Scheduler 4-11
- defining parameters 4-11, B-1 to B-5
- defining system messages 4-11
- described 1-7, 4-3
- managing the job queue 4-5 to 4-9
- starting 4-5
- submitting jobs 4-10
- user interface 4-4
- Aggregate Recommendations report 6-18
- aggregate tables 2-7, 2-12
 - candidate
 - specifying number for Warehouse Optimizer 6-12
 - creating 2-5, 2-10
 - recommended
 - viewing details of 6-18, 6-19
 - recommending
 - basis for 6-12
 - how many to recommend 6-17
 - memory needed 6-5
 - with Warehouse Optimizer 6-16
 - registering
 - in metadata 6-19
 - with Warehouse Optimizer 6-19
 - stages of optimization

- choosing with Warehouse Optimizer 6-10
 - use in optimizing MetaCube performance 6-3
 - validating 6-6
- Alert jobs 4-10
- analysis file
 - for Warehouse Optimizer 6-10
- architecture of the MetaCube system 1-8 to 1-10
- three-tier 1-8
- two-tier 1-10
- audit data
 - analyzing to recommend
 - aggregates 6-5, 6-16
 - collecting 6-17
 - defined 6-16
 - importance in aggregation 6-4
 - viewing 6-16
 - weight in aggregate
 - recommendation 6-17
- authorizing user access
 - in Secure Warehouse 5-8
 - to Secure Warehouse and Warehouse Manager 2-7

B

- base combinations
 - of dimension elements 6-7

C

- cache file
 - for Warehouse Optimizer 6-5
- candidate aggregates

- specifying number for Warehouse Optimizer 6-12
- collecting audit data 6-17
- combinations
 - of dimension elements
 - calculating for Warehouse Optimizer 6-7
- configuring
 - an aggregate analysis
 - choosing data source 6-11
 - full 6-11
 - partial 6-11
 - selecting analysis type 6-11
 - specifying number of candidates 6-12
 - with Warehouse Optimizer 6-11
 - memory for aggregate recommendation 6-5
 - MetaCube preferences C-1
 - metacube.ini file C-1
- connect string
 - stored as connection information 5-6
- connection information
 - connect string 5-6
 - database type 5-6
 - for users in Secure Warehouse 5-6
 - metaschema 5-6
 - user ID 5-6
- cost
 - for fact table
 - setting for Warehouse Optimizer 6-4
 - in Warehouse Optimizer
 - of query 6-14, 6-18
 - of storage 6-13, 6-14, 6-18
- Cost Graph 6-19

D

- data
 - audit 6-4
 - collecting 6-17
 - defined 6-16
 - viewing 6-16
 - weight in aggregate recommendation 6-17

- density
 - importance in aggregation 6-3
- marts
 - defined 1-3
 - on user queries 6-4
- sources 2-4, 2-7
 - choosing for Warehouse Optimizer 6-11
 - creating 2-7
- data warehouses
 - defined 1-3
 - maintaining 2-12
 - setting up 2-1 to 2-12
 - authorizing user access 2-4, 2-9
 - creating aggregate tables 2-5, 2-10
 - creating metadata 2-4, 2-7
 - creating sample tables 2-5
 - creating secure users 2-4, 2-7
 - defining public filters and queries 2-4, 2-8
 - designing a schema 2-3, 2-5
 - generating metadata tables 2-4, 2-6
 - installing MetaCube 2-3, 2-6
 - overview 2-3
 - populating data warehouse tables 2-3, 2-6
 - starting MetaCube Scheduler 2-5, 2-10
- tables for 2-3
- database
 - logins
 - changing 3-8
 - roles
 - assigning in Secure Warehouse 5-10
- type
 - stored as connection information 5-6
- deleting jobs
 - in job queue 4-8
- dependencies
 - for jobs in job queue
 - changing 4-8, 4-9
- displaying contents
 - of job queue 4-8
- DOS utility
 - for Web Publisher jobs A-1

- possible
 - components A-2 to A-3
 - sample commands A-3
- DSS Systems 2-4, 2-7
 - authorizing user access 2-4, 2-9, 5-3, 5-8
 - creating 2-7

F

- fact table costs
 - setting for Warehouse Optimizer 6-4
- filters
 - public 2-4, 2-8
- folder structure
 - creating 2-4, 2-9
- Full Aggregate jobs 4-10
- full analysis
 - of a DSS System
 - with Warehouse Optimizer 6-11

G

- generating
 - combinations for aggregate tables
 - reducing time involved 6-12
 - costs of queries
 - in Warehouse Optimizer 6-13
 - reducing time involved 6-13
 - graph of aggregate recommendation data 6-19
- groups of users
 - managing in Secure Warehouse 5-10

H

- HTML output
 - of a Web Publisher job A-1

I

- identifying users
 - to be managed in Secure Warehouse 5-7

- inactive users
 - in Secure Warehouse 5-5
- Incremental Aggregate jobs 4-10

J

- job queue
 - displaying contents 4-8
 - how jobs are processed 4-6
 - information provided about jobs 4-6
 - managing with Agent Administrator 4-5
 - modifying jobs 4-8
 - refreshing 4-6
- Job Set jobs 4-10
- jobs
 - Alert 4-10
 - Full Aggregate 4-10
 - Incremental Aggregate 4-10
 - Job Set 4-10
 - Operating System 4-10
 - Post Incremental Aggregates 4-10
 - Sample 4-10, 6-6
 - SQL non-Select 4-10
 - SQL Select 4-10, B-5
 - submitting with Agent Administrator 4-10
 - Web Publisher 4-10, A-1

L

- ListView pane 5-4
 - of Secure Warehouse 5-4

M

- memory requirements
 - for aggregate recommendation in Warehouse Optimizer 6-5
- MetaCube
 - analysis engine
 - introduced 1-4
 - registry 5-6
 - installation 2-3, 2-6
 - preferences
 - configuring C-1

- software components 1-4
 - Agent Administrator 1-7, 4-1 to 4-11
 - Explorer 1-4
 - MetaCube analysis engine 1-4
 - MetaCube for Excel 1-4
 - MetaCube Scheduler 3-1 to 3-9
 - Scheduler 1-6
 - Secure Warehouse 1-6, 5-1 to 5-11
 - SQL Optimizer 1-5
 - Warehouse Manager 1-5
 - Warehouse Optimizer 1-7, 6-1 to 6-21
 - Web Explorer 1-4
- system architecture 1-8 to 1-10
 - three-tier 1-8
 - two-tier 1-10
- system overview 1-1 to 1-10
- MetaCube Agents 1-6, 2-10
- MetaCube Explorer
 - introduced 1-4
- MetaCube for Excel
 - introduced 1-4
- MetaCube Scheduler 3-1 to 3-9
 - controlling with Agent Administrator 4-11
 - described 1-6, 3-3
 - processing jobs in queue 4-6
 - starting 2-5, 2-10
 - starting in UNIX 3-4
 - checking status 3-5
 - skipping prompts 3-4
 - starting in Windows NT 3-7
 - as a Windows NT service 3-7
 - configuring ODBC 3-8
 - stopping 3-6, 3-8
- MetaCube SQL Optimizer
 - introduced 1-5
- metacube user 2-6
- MetaCube Web Explorer
 - introduced 1-4
- metacube.ini file C-1 to C-3
 - configuring memory for aggregate recommendation in Warehouse Optimizer 6-5
 - turning on performance log for Warehouse Optimizer 6-20
- metadata 1-5, 2-4, 2-7

- registering aggregates in 6-19
- tables
 - generating 2-4, 2-6
- metapub user 2-8, 5-9
- metaschema
 - stored as connection information 5-6
- modifying jobs
 - in the job queue 4-8

O

- ODBC
 - configuring for MetaCube Scheduler 3-8
 - database logins 3-8
 - datasources 3-8
 - Startup options 3-9
 - datasources 3-7, 5-6
 - changing 3-8
- Operating System jobs 4-10
- overview
 - of MetaCube system 1-1 to 1-10
 - of process for setting up data warehouse 2-3

P

- parameters B-1 to B-5
 - defining with Agent Administrator 4-11
 - described B-1
 - in QueryBack jobs B-2
 - in SQL jobs B-5
 - user-defined B-4
- partial analysis
 - of a DSS System
 - used to speed computing of storage costs 6-13
 - with Warehouse Optimizer 6-11
- performance log
 - in Warehouse Optimizer 6-20
- Post Incremental Aggregates jobs 4-10
- priority
 - for jobs in job queue
 - changing 4-8, 4-9

- processing time
 - for aggregate registration 6-20
- prompts
 - skipping while starting MetaCube Scheduler 3-4
- public filters and queries 2-4, 2-8

Q

- queries
 - public 2-4, 2-8
- Query Audit Detail report 6-16
- query cost
 - in Warehouse Optimizer of recommended aggregates 6-14, 6-18

R

- recommending aggregates
 - in Warehouse Optimizer 6-16
 - analyzing audit data 6-16
 - configuring memory 6-5
 - how many to recommend 6-17
 - tasks involved 6-16
- refreshing the job queue 4-6
- registering aggregates
 - in Warehouse Optimizer 6-19
- registry
 - for MetaCube engine 5-6
- reports in Warehouse Optimizer Aggregate
 - Recommendations 6-18
 - Cost Graph 6-18, 6-19
 - Query Audit Detail 6-16
- ROOT folder 2-9
- row count
 - used for fact table cost 6-4

S

- sample commands
 - for Web Publisher DOS utility A-3
- Sample jobs 4-10, 6-6
- sample tables 2-7, 2-12
 - creating 2-5

- used with Warehouse Optimizer to speed computing of storage costs 6-6
- saving results
 - in Warehouse Optimizer 6-10
- schedule of execution
 - for jobs in job queue changing 4-8, 4-9
- Scheduler 1-6
 - MetaCube 3-1 to 3-9
 - controlling with Agent Administrator 4-11
 - described 1-6, 3-3
 - starting 2-5, 2-10
 - starting in UNIX 3-4
 - starting in Windows NT 3-7, 3-8
 - stopping 3-6, 3-8
 - Windows NT or Windows 95 A-1
- schema 2-3, 2-5
- secure users 2-4, 2-7, 2-10, 5-9, 6-19
- Secure Warehouse 5-1 to 5-11
 - described 1-6, 5-3
 - granting access to 5-9
 - logging on the first time 5-9
 - managing groups of users 5-10
 - managing snap-ins 5-11
 - process of using 5-7
 - authorizing user access 5-8
 - identifying users 5-7
 - limiting availability of jobs for users 5-8
 - specifying how user queries are processed 5-9
 - user connection information 5-6
 - user interface 5-4
 - active and inactive users 5-5
 - List View pane 5-4
 - TreeView pane 5-4
- snap-ins
 - managing with Secure Warehouse 5-11
- software components
 - of MetaCube system 1-4
 - Agent Administrator 1-7, 4-1 to 4-11
 - Explorer 1-4
 - MetaCube analysis engine 1-4
 - MetaCube for Excel 1-4
 - MetaCube Scheduler 3-1 to 3-9

- Scheduler 1-6
- Secure Warehouse 1-6, 5-1 to 5-11
- SQL Optimizer 1-5
- Warehouse Manager 1-5
- Warehouse Optimizer 1-7, 6-1 to 6-21
- Web Explorer 1-4
- SQL non-Select jobs 4-10
- SQL Select jobs 4-10
 - use of parameters B-5
- starting Agent Administrator 4-5
- starting MetaCube Scheduler
 - in UNIX 3-4
 - checking status 3-5
 - skipping the prompts 3-4
 - in Windows NT 3-7
 - configuring ODBC 3-8
- starting Warehouse Optimizer 6-10
- status of jobs in job queue
 - changing 4-9
- stopping MetaCube Scheduler
 - in UNIX 3-6
 - in Windows NT 3-8
- storage costs
 - in Warehouse Optimizer 6-14
 - of recommended aggregates 6-18
 - estimating 6-14
- storing results
 - in Warehouse Optimizer 6-10
- submitting jobs
 - with Agent Administrator 4-10
- synchronizing users
 - in Secure Warehouse 5-7
- system messages
 - defining with Agent Administrator 4-11

T

- tempview
 - used in Warehouse Optimizer for storage cost computation 6-15
- three-tiered architecture
 - of MetaCube system 1-8
- TreeView pane 5-4
 - of Secure Warehouse 5-4

two-tiered architecture
of MetaCube system 1-10

U

updating aggregate
recommendations
in Warehouse Optimizer 6-21
user access
to DSS Systems 2-4, 2-9, 5-3
user connection information
and Secure Warehouse 5-6
connect string 5-6
database type 5-6
metaschema 5-6
user ID 5-6
user ID
stored as connection
information 5-6
user interface
Agent Administrator 4-4
Secure Warehouse 5-4
active and inactive users 5-5
ListView pane 5-4
TreeView pane 5-4
Warehouse Optimizer 6-9
users
authorized for Secure
Warehouse 2-4, 2-7
creating 2-7
authorizing access
to DSS Systems 2-9
in groups
managed in Secure
Warehouse 5-10

V

validating
aggregates 6-6

W

Warehouse Manager
introduced 1-5
Warehouse Optimizer 2-10,
6-1 to 6-21

calculating combinations 6-7
configuring the analysis 6-11
described 6-3
full analysis 6-11
generating costs 6-13
introduced 1-7
partial analysis 6-11
performance log 6-20
recommending aggregates 6-16
analyzing audit data 6-16
configuring memory 6-5
how many to recommend 6-17
tasks involved 6-16
registering aggregates 6-19
rerunning 6-21
setting fact table costs 6-4
stages of 6-10
starting 6-10
storing results 6-10
updating aggregate
recommendations 6-21
user interface 6-9
Web Publisher A-1
DOS utility A-1
possible
components A-2 to A-3
sample commands A-3
jobs 4-10
HTML output A-1
Windows NT service 3-7

Symbols

_whocach_cc.mwo file 6-5

