Introduction to New Features

MetaCube ROLAP Option

Version 4.x

Informix Dynamic Server
Oracle Database
Red Brick Warehouse

Version 4.2
April 1999
Part No. 000-5381

Published by INFORMIX® Press

Informix Corporation
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Table of Contents

In This Guide

This guide introduces the changes for all MetaCube 4.x releases. The purpose of this guide is to inform existing MetaCube users of modifications to user interface or functionality that may not be described in the printed manuals. Online help for MetaCube applications is updated to reflect all new features for the 4.x releases.

Documentation for 4.x releases emphasizes online help. The online help systems include the following components:

- **Help Topics:** a complete online help system that contains “how to” topics and procedural information on using the product.
- **Context sensitive help:** a help system that allows users 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 or within a dialog box displays help specific to that feature or dialog box.
- **Help on Help:** contains instructions for using an online help system.

Printed manuals for the 4.x releases contain overview and conceptual information about the features and functions of the MetaCube products. Procedural information no longer appears in printed manuals and is contained in the help systems for each MetaCube application.

The *MetaCube Explorer User’s Guide* contains many tutorials on the use of the various features of Explorer. Almost all chapters contain one or more short tutorial. To do the tutorials requires access to the MetaCube demonstration database distributed with the product.

The *MetaCube Data Warehouse Administrator’s Guide*, a new book for MetaCube 4.0 release, provides an overview of the tasks a data warehouse administrator performs to create and maintain a data warehouse. It discusses the MetaCube administrative tools for data warehouse administrators. It is not updated for subsequent 4.x releases; new information pertaining to the tools for data warehouse administrators is included in this guide.

New Features for Release 4.0

Most changes in Release 4.0 arise from MetaCube's new architectural flexibility. MetaCube runs, as it always has, in a traditional two-tier, client/server configuration, but it can now also operate in a three-tier configuration.

Release 4.0 introduces several new MetaCube applications:

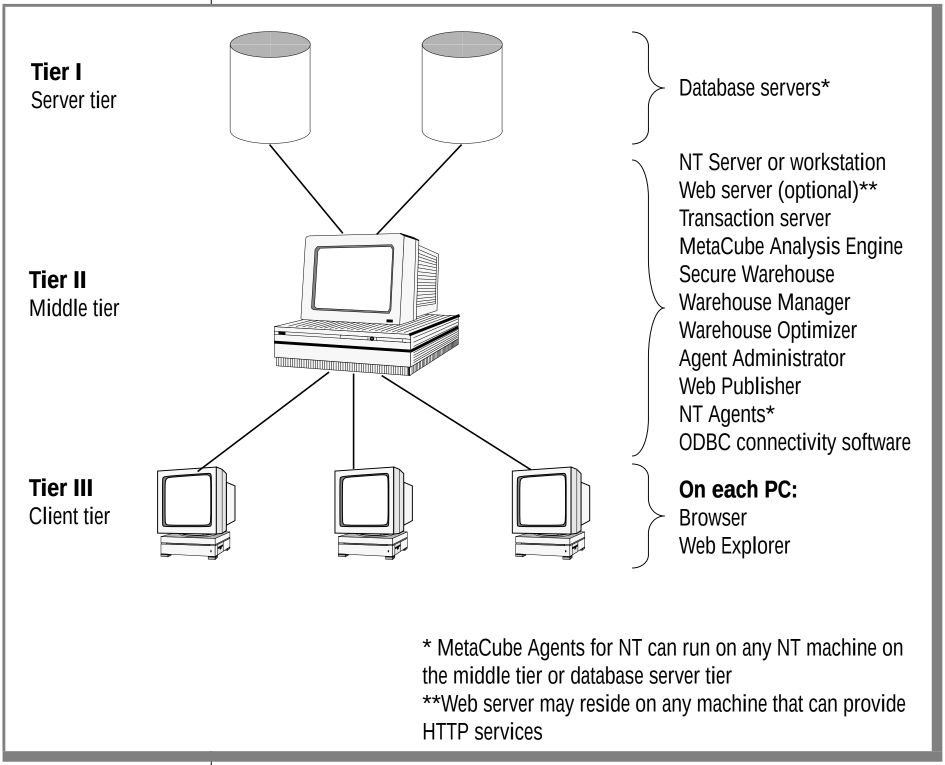
- Secure Warehouse incorporates system-wide security features that allow administrative control of user access to the database.
- Explorer comes in two versions—a traditional client/server implementation and Web Explorer, which runs inside a Web browser. Both applications employ the same user interface, which has been redesigned for this release.
- SQL Optimizer allows non-MetaCube SQL-generating query tools to access a MetaCube data warehouse system.

Some existing MetaCube applications have been improved. These include Warehouse Optimizer, Agent Administrator, MetaCube's Visual Basic application programming interface (API), and the MetaCube SDK for Snap-Ins for C++ programmers.

System Architecture

MetaCube supports a traditional two-tier client/server architecture and a new three-tier architecture, which allows users to run MetaCube Web Explorer through a Web browser. Figure 1 shows a typical three-tier architecture.

Figure 1
Typical Three-Tier Architecture



* MetaCube Agents for NT can run on any NT machine on the middle tier or database server tier
**Web server may reside on any machine that can provide HTTP services

In a three-tier 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 act as a Web server, although that is not required; the Web server can run anywhere in the network.

With a three-tier architecture, the MetaCube Analysis Engine resides on the middle tier rather than on the PCs of individual end users. Thus, one MetaCube Analysis Engine serves many MetaCube Explorer clients concurrently.

In addition to the MetaCube Analysis Engine, all the MetaCube administrative tools used to create and manage a data warehouse run on the middle tier. Those components are:

- Secure Warehouse
- Warehouse Managerr
- Agent Administrator
- Web Publisher
- Warehouse Optimizer

The MetaCube Agents can run on either the middle tier or the database server platform.

Server Tier

Database servers constitute the server tier of the three-tier architecture. MetaCube Agents can optionally run on the server tier rather than the middle tier.

Client Tier

The client tier configuration depends on the system architecture.

Client Tier in a Two-Tier Architecture

In a two-tier configuration, the client tier consists of Client / Server Explorer and the MetaCube Analysis Engine. This configuration is the client / server configuration used in previous MetaCube releases. However, one client-tier machine should be used for administrative purposes. All MetaCube administrative tools, such as Secure Warehouse, Warehouse Manager, and Agent Administrator, should be installed there.

Client Tier in a Three-Tier Architecture

In a three-tier architecture, the client tier typically consists of personal computers equipped with Web browsers from which users run MetaCube Web Explorer. One or more MetaCube Analysis Engines running on the middle tier support these client machines.

In some situations, the client tier may consist of personal computers running MetaCube Explorer or MetaCube for Excel. Those clients are served by one or more MetaCube Analysis Engines running on the middle tier. Using MetaCube Explorer or MetaCube for Excel in a three-tier architecture shifts the computation burden to the middle tier, which may be desirable for sites where client machines are not capable of handling intensive processing loads. In a three-tier configuration supporting only Client / Server Explorer or MetaCube for Excel, a Web server is not necessary.

Secure Warehouse

MetaCube Secure Warehouse introduces a security paradigm for the MetaCube system. With Secure Warehouse, a data warehouse administrator explicitly grants user permissions to access a DSS System. In this way, an administrator can control a user's column- and hierarchy-level access to the database. In addition, the administrator can place mandatory filters on a user's queries, restricting the user's row-level database access.

For example, with Secure Warehouse, a user may be granted access to a DSS System that provides only summary sales data. By applying a mandatory filter, a user could be restricted to querying sales data from a particular region, such as the East, even though the DSS System may contain summary data for the entire country.

Unauthorized users confront system restrictions in different ways. If not granted access to a DSS System through Secure Warehouse, a user may be able to start a MetaCube application but cannot connect to the database. If the user has connected to the database but attempts to query restricted data, the query fails. A user of Web Explorer cannot even access the MetaCube application until that user has been added to the collection of users managed in Secure Warehouse and a set of default connection properties has been defined for that user.

Secure Warehouse provides some additional security features. The data warehouse administrator can restrict the date and time at which a user's queries to the database can execute. The administrator can also control the level of system resources available to the user by setting PDQ Priority.

Where to Install Secure Warehouse

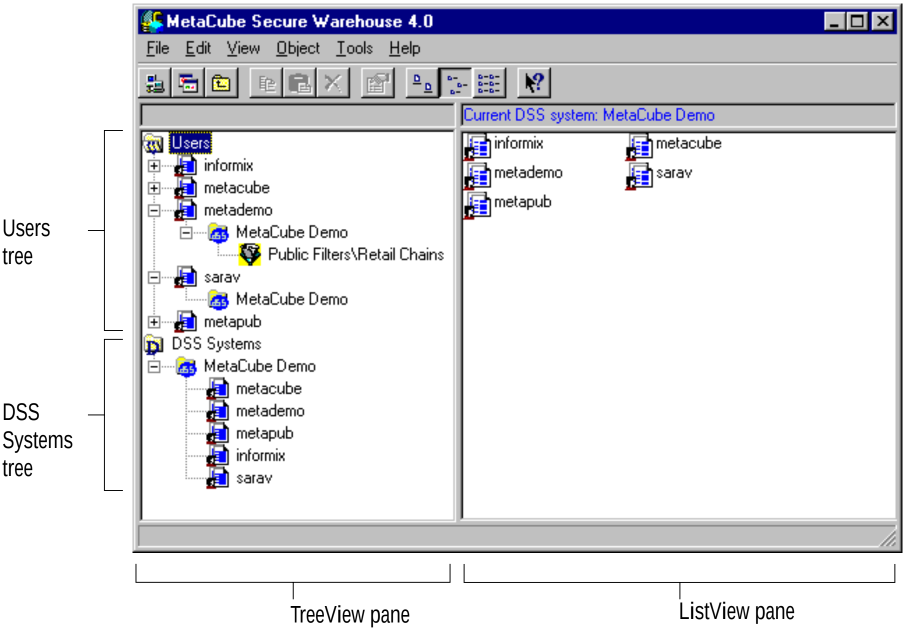
Where Secure Warehouse is installed depends on the system architecture. In a traditional two-tier, client / server architecture, Secure Warehouse and all the other MetaCube administrative tools, such as Agent Administrator and Warehouse Manager, should be installed on one PC. In a three-tier architecture, Secure Warehouse must be installed on every middle-tier machine running the MetaCube Analysis Engine.

Installation procedures for Secure Warehouse are described in the *MetaCube Installation and Configuration Guide*.

The Secure Warehouse User Interface

The MetaCube Secure Warehouse main window is shown in Figure 2 on page 6. It consists of two panes, the TreeView and the ListView. The TreeView shows two tree structures: the Users tree provides information on a per-user basis, and the DSS Systems tree provides information a per-DSS-System basis. The ListView shows the contents of the folder last selected in the TreeView.

Figure 2
Secure Warehouse
Main Window



Explorer

In Release 4.0, there are two versions of MetaCube Explorer:

- **Client/Server Explorer**, an updated version of MetaCube Explorer, typically runs in a two-tier architecture where Explorer and the MetaCube Analysis Engine run on the same client PC. An ODBC driver, which establishes a connection to the database, also runs on that PC. Client / Server Explorer can also run in a three-tier architecture, communicating with a MetaCube Analysis Engine on the middle tier using DCOM (Distributed Component Object Model) connectivity software.
- **Web Explorer**, a new feature in Release 4.0, runs in a Web browser and requires a three-tier configuration (see "System Architecture" on page 3). When a user accesses an internal MetaCube Web page (the location of which is site-specific) and invokes Web Explorer for the first time, all necessary software for Web Explorer is downloaded to the user's PC. The connection to the MetaCube Analysis Engine and the database is established, and the application is launched. For all future sessions, the user must still access the MetaCube Web page, which establishes a database connection, but no additional software is downloaded. Web Explorer saves workbooks on the user's local PC.

Functionally, Web Explorer and Client / Server Explorer offer the same features and employ the same user interface, with the exception of the toolbar and the menu bar.

The options on the Web browser menu bar in which Web Explorer is running do not apply to MetaCube. Menu bar options apply to the Web browser itself. Therefore, all Web Explorer options are accessed from the toolbar.

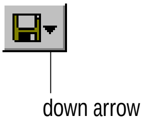


Figure 3
Web Explorer Toolbar Button with Down Arrow

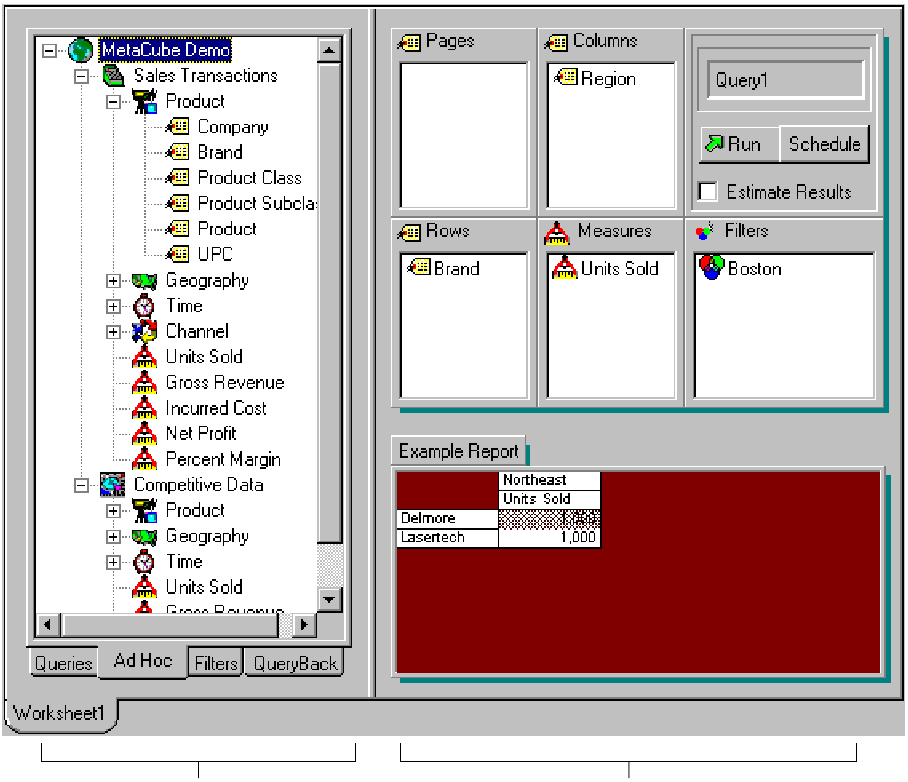
Some buttons on the toolbar have pull-down functionality, denoted by a down arrow on the button, as shown in Figure 3.

Using Web Explorer, a user can open one workbook at a time. With Client / Server Explorer, several workbooks may be open simultaneously.

Client / Server Explorer and Web Explorer interface and functionality for 4.x releases are described in detail in the *MetaCube Explorer User's Guide* and in the online help for the two applications.

New Explorer User Interface

Both versions of Explorer are redesigned to require fewer mouse clicks to create queries, access and run stored queries, apply and create filters, and submit and manage QueryBack jobs.



	Northeast
	Units Sold
Delmore	1,000
Lasertech	1,000

Tab page area

Query definition area

Figure 4
Redesigned Explorer Main Window

As shown in Figure 4, Explorer's main window is divided into two sections: a tab page area and a query definition area

Tab Page Area

The tab page area of the Explorer window contains four tab pages:

- The Queries tab page displays folders that contain saved queries. Click this tab to run or work with pre-defined queries.
- The Ad Hoc tab page (shown in Figure 4 on page 8) displays the DSS System hierarchy with icons representing the attributes and measures that make up the current DSS System. Click this tab to define a query. All data sources for the DSS System are displayed in the Ad Hoc tab page, making it easy to incorporate objects from multiple data sources into a single query.
- The Filters tab page displays folders that contain filters. Click this tab to create, apply, or work with filters.
- The QueryBack tab page displays jobs that have been submitted for background processing. Click this tab to work with QueryBack jobs.

Query Definition Area

The query definition area consists of drop boxes, the control area, and the Example Report area.

Drop Boxes

As shown in Figure 5, the Explorer window provides five drop boxes.

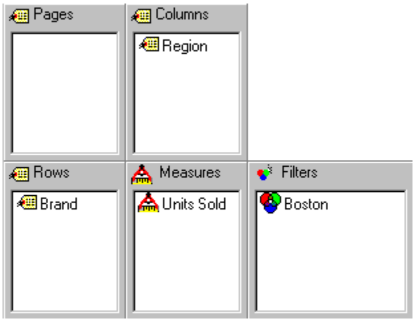


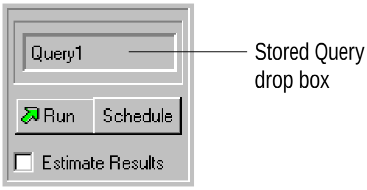
Figure 5
Main Window Drop Boxes

- **Attribute drop boxes for Rows, Columns, and Pages.** When creating a new query, a user can double-click attribute icons in the Ad Hoc tab page or drag them to drop boxes. In this way, a user can define the query and specify an initial format for the resulting report. To format multipage reports in Query Mode, drop an attribute icon into the Pages drop box.
- **Measures drop box.** Double-clicking or dragging a measure icon to this drop box specifies the data (that is, the measures) to be returned by the query.
- **Filters drop box.** To apply a filter to a query, click the Filters tab and then double-click or drag a filter icon to this drop box.

Control Area

As shown in Figure 6, the control area contains features for running queries.

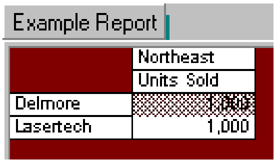
Figure 6
Control Area



- **Stored Query drop box.** When using the Queries tab, double-click or drag the icon for a stored query to the stored query drop box. The name of the stored query displays in the drop box, and the components of the query are displayed in their appropriate drop boxes.
- **Run button.** After populating the drop boxes in the query definition area, click this button to run a query.
- **Schedule button.** After populating the drop boxes in the query definition area, click this button to schedule a QueryBack job. Once a QueryBack job is scheduled, its icon appears in the QueryBack tab page. To retrieve the results of a QueryBack job, click the QueryBack tab and use the status list to obtain query results.
- **Estimate Results box.** Check this box before running a query to use Explorer’s Sampling capabilities.

Example Report Area

The Example Report area displays a prototype report that changes dynamically if the user repositions icons in the drop boxes. The Example Report area functions the same as it has in earlier versions of Explorer.



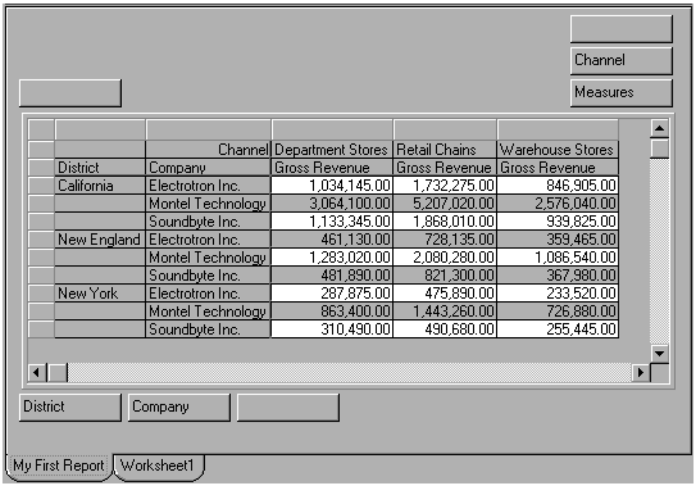
	Northeast
	Units Sold
Delmore	1,000
Lasertech	1,000

Figure 7
Example Report Area

Explorer Reports

The following improvements to the Explorer report screen facilitate its use:

- Scroll bars display automatically when viewing a report.
- Headings of a report can be frozen so they remain visible while scrolling horizontally or vertically.
- Report headings display attribute names.



	Channel	Department Stores	Retail Chains	Warehouse Stores
District	Company	Gross Revenue	Gross Revenue	Gross Revenue
California	Electrotron Inc.	1,034,145.00	1,732,275.00	846,905.00
	Montel Technology	3,064,100.00	5,207,020.00	2,576,040.00
	Soundbyte Inc.	1,133,345.00	1,868,010.00	939,825.00
New England	Electrotron Inc.	461,130.00	728,135.00	359,465.00
	Montel Technology	1,283,020.00	2,080,280.00	1,086,540.00
	Soundbyte Inc.	481,890.00	821,300.00	367,980.00
New York	Electrotron Inc.	287,875.00	475,890.00	233,520.00
	Montel Technology	863,400.00	1,443,260.00	726,880.00
	Soundbyte Inc.	310,490.00	490,680.00	255,445.00

Figure 8
Explorer Report

Pivot handles exhibit some new visual qualities; for example, they change color when a user drags them. Despite the superficial changes, pivot handles function exactly the same as in previous releases of MetaCube.

Shortcut Menus

Shortcut menus, displayed by right-clicking an object or feature in Explorer, have been enhanced in Release 4.0. Listed below are various actions for which a relevant right-click shortcut menu is available.

Action	Right-click shortcut menu options available
Place an attribute icon in a drop box	■ Remove (from the drop box) ■ Rename (the attribute for this query) ■ Sort (options for attributes) ■ Subtotals (for attributes in a break report) ■ New / Edit Custom Comparison (for comparing disparate attributes) ■ Buckets (for customized groupings of attribute values) ■ Edit Definition (for changing internal information about the attribute)
Place a measure icon in a drop box	■ Remove (from the drop box) ■ Rename (the measure for this query) ■ Calculation (to apply a MetaCube measure calculation) ■ Format (to specify format properties for a measure) ■ Stoplighting (for visually highlighting measure data) ■ Edit Definition (for changing internal information about the measure)
Place a filter icon in a drop box	■ Remove (from the drop box)

(1 of 3)

Action	Right-click shortcut menu options available
Right-click a filter icon on the Filters tab page	■ Cut (the filter definition to the clipboard) ■ Copy (the filter definition to the clipboard) ■ Paste (the filter definition in the clipboard to the current location) ■ Delete (the filter icon) ■ Rename (the filter icon) ■ Create a New Filter (to begin the process of creating a new filter definition) ■ Edit (the filter definition) ■ Make / Clear Default (to assign / clear a default filter designation)
Right-click a stored query icon on the Queries tab page	■ Cut (the stored query definition to the clipboard) ■ Copy (the stored query definition to the clipboard) ■ Paste (the stored query definition in the clipboard to the current location) ■ Delete (the stored query icon) ■ Rename (the stored query)
Right-click a folder icon on the Queries tab page or Filters tab page	■ Delete (the folder, if it is empty) ■ Rename (the folder) ■ Create New Filter (in the Filters tab page only)
Right-click the bottom of a worksheet or the worksheet tab	■ Query Mode / Results Mode (switches between Explorer's two display modes) ■ New Worksheet (to open a new blank worksheet) ■ Copy Query (to open a new worksheet and place a copy of the current query in it) ■ Rename Worksheet (to rename the current worksheet) ■ Clear Worksheet (to clear all icons from the drop boxes of the correct worksheet) ■ Delete Worksheet (to delete the current worksheet)

(2 of 3)

Action	Right-click shortcut menu options available
Right-click in a report	■ Error Margin (after right-clicking a cell containing data retrieved from a sample table, displays the margin of error for that data item) ■ Report Properties (to set report properties for the current report) ■ Report Information (to display detailed information about the report) ■ Subtotals (after right-clicking a pivot handle containing the name of the attribute for which to calculate subtotals) ■ Freeze Headings (to prevent column and row headings from scrolling off the screen in a large report) ■ Clear Measure Sorts (to clear sorting applied to measures in Results Mode)

(3 of 3)

Other Features

With some minor exceptions, the dialog boxes that allow access to Explorer functions and options have not changed from previous releases.

Explorer charts look the same as in previous releases. Although the sorting function is no longer available, all other chart functions remain the same as in previous MetaCube releases.

SQL Optimizer

MetaCube SQL Optimizer is a new product in Release 4.0 and is described in detail in the *MetaCube SQL Optimizer User's Guide*. SQL Optimizer is an application programming interface (API) that allows non-MetaCube packaged or custom SQL-generating query tools to access a MetaCube Analysis Engine and thus a MetaCube data warehouse.

SQL Optimizer can be accessed either through an ODBC interface or the MetaCube data interface (MDI). It receives standard SQL statements, as generated by a query tool or application, and restructures the SQL so that it is optimized for a MetaCube data warehouse's aggregate and sample tables.

Through the use of SQL Optimizer, a non-MetaCube query tool can be easily converted to run with the MetaCube Analysis Engine. This conversion can yield a significant improvement in query-processing time for end users. It also allows users of other query tools to access the full complement of MetaCube applications.

Warehouse Optimizer

Warehouse Optimizer features tighter memory usage and faster processing. It also offers the following new capabilities.

- **Combinations generated Automatically.** Generating dimension element combinations was previously a separate stage in warehouse optimization. In Release 4.0, this process is performed automatically when the user confirms the configuration settings.
- **Maximum and minimum settings on dimensions.** When performing a partial analysis, users can now 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. By limiting the number of dimensions in candidate aggregates, users decrease the number of possible candidate aggregates and therefore decrease the time required for storage cost generation.
- **Estimate costs.** Users can choose whether to estimate storage costs in the cost-generation stage. With estimate costs, Warehouse Optimizer submits queries for the aggregate candidates made of base combinations—all dimension element combinations consisting of one or two elements—and estimates costs for all remaining combinations. This greatly reduces the time needed to generate costs.
- **Memory swapping in the recommendation stage.** Users can allocate the memory that Warehouse Optimizer will use during its aggregate recommendation stage by setting a value in the **metacube.ini** file. If enough memory is not allocated for all combinations to be read into memory at one time, Warehouse Optimizer creates a cache file to hold combinations and swaps them into memory from the cache file.

- **Performance log.** Warehouse Optimizer can now create a performance log, which can be used to view information on the number of combinations processed and the processing time needed for the cost generation and aggregate recommendation stages. The performance log can be turned on and off by setting a value in the **metacube.ini** file.

Important: *Aggregate tables cannot be created using Warehouse Optimizer. Instead, use Agent Administrator to submit a job that creates aggregate tables based on the aggregate table definitions registered in metadata by Warehouse Optimizer.*

For detailed information about Warehouse Optimizer, refer to online help for the application and the *MetaCube Data Warehouse Administrator's Guide*.

MetaCube Agents and Agent Administrator

MetaCube Agents and MetaCube Agent Administrator offer several new features. MetaCube Agents and Agent Administrator are described in detail in the *MetaCube Data Warehouse Administrator's Guide* and in the online help for Agent Administrator.

- **Displaying jobs in the Job Queue dialog box.** MetaCube Agent Administrator offers new methods to sort and filter information displayed in the Job Queue dialog box. Users can:
    - arrange jobs in the Job Queue dialog box by clicking the header for the category of information to be sorted. For example, by clicking on the Job Name header, all jobs are sorted in ascending order by job name. Click again and they are sorted in descending order.
    - define filters, based on job type, that control which jobs are displayed in the Job Queue dialog box. For example, a filter might display only jobs types equal to SQL **Select** jobs.
- **Alerter Agent and the Alert job.** An Alert job, which uses the Alerter Agent, sends an email message when a specified SQL SELECT statement returns one or more rows. Alert jobs are particularly useful for notifying administrators of an exception condition.
- **Split full aggregates.** The Full Aggregate job now allows specification of a strategy for splitting a job into pieces to optimize the system resources it needs. When the Full Aggregate job completes, it assembles those pieces into a single aggregate table.

MetaCube Analysis Engine

In Release 4.0, the MetaCube Analysis Engine can operate on a middle tier. By locating the MetaCube Analysis Engine on centralized, powerful machines, client applications can be deployed on machines that are less powerful. The three-tier architecture also helps to minimize network traffic. The MetaCube Analysis Engine can also utilize the features of Microsoft Transaction Server to perform load balancing and optimization.

Visual Basic API

Changes have been made to the Visual Basic application programming interface to the MetaCube Analysis Engine. Refer to the *MetaCube Application Programmer's Manual* for complete details on the MetaCube OLE Automation Interface.

- **New object classes.** The following new classes of object are used to support security restrictions introduced in MetaCube Secure Warehouse:
    - ❑ User
    - ❑ Users
    - ❑ DSSSystem
    - ❑ DSSSystems
    - ❑ AvailableDSSSystems
- **Support for object type variables.** Using MetaCube 4.0 and Visual Basic or Visual Basic for Applications, programmers can declare MetaCube-specific object type variables and then safely create new instances of the MetaCube object classes. This approach to creating objects yields multiple advantages, including faster processing time, type safety, and function checking during the coding process.
- **New connection paradigm.** In previous releases, the MetaCube Analysis Engine controlled connection information in the metacube.ini file. In Release 4.0, connection information is managed by each MetaCube application, although connection information is still stored in the **metacube.ini** file. Connection information for Web Explorer users is managed in Secure Warehouse.

- **Changes to existing object classes.** New properties and methods have been added to MetaCube's existing classes of object, and some existing properties and methods have also been modified. The following list shows all new or changed properties and methods in Release 4.0:

  Filter.FullPathName

  Folder.FullPathName

  Metabase.Configuration
  Metabase.ConnectDatabase
  Metabase.ConnectString
  Metabase.CreateNew
  Metabase.DatabaseDBSpaces
  Metabase.DatabaseRoles
  Metabase.DatabaseUsers
  Metabase.DataSources
  Metabase.DeleteMetamodel
  Metabase.MetamodelNames
  Metabase.MetaSchema
  Metabase.Password
  Metabase.RemoteConnect
  Metabase.Role
  Metabase.Save
  Metabase.SaveAs
  Metabase.SuppressDialogs

  Query.AsynchLastError
  Query.AsynchRetrieve
  Query.AsynchRetrieveStatus
  Query.CancelAsynchRetrieve

MetaCube SDK for Snap-Ins

MetaCube SDK for Snap-Ins allows C++ programmers to write a MetaCube Snap-In, a custom measure calculation for MetaCube Explorer or MetaCube for Excel. The *MetaCube SDK for Snap-Ins Programmer's Manual* documents the MetaCube SDK Extension Wizard and the MetaCube SDK for Snap-Ins.

The following member functions and data members in the MetaCube SDK for Snap-Ins have been added or modified:

CDb::GetColumns
CDb::GetDataSources
CDb::GetTables
CDb::GetUsers
CDb::SelectStmt

CParseNode::m_number_val

Folder::GetFullPathName

Metabase::CreateNew
Metabase::DeleteMetamodel
Metabase::Get_m_configuration
Metabase::GetUniqueID
Metabase::OpenQueryStorage
Metabase::Save
Metabase::SaveAs

In addition, minor changes have been made in the process of using Microsoft Developer Studio to debug a MetaCube Snap-In.

Installation Procedures

Because of new MetaCube architecture and functionality options, the installation process is more complex than in previous releases. Updated procedures reflect the choice of a two- or three-tier architecture, the availability of MetaCube Web Explorer, and the introduction of security measures.

Administrators must make several strategic decisions before installing software. They must decide whether users should run Client/Server Explorer or Web Explorer. If they choose to run MetaCube Web Explorer, the MetaCube system must be configured for three tiers. If they choose Client/Server Explorer, the MetaCube system can be configured for two tiers, but if client machines are not especially powerful, it may be more efficient to use a three-tier configuration and shift the processing burden to more powerful PCs or servers on the middle tier.

If a site uses Web Explorer, configuration of a Web server is necessary. Web Explorer is invoked from an HTML-formatted Web page. A Web site must accommodate the Web pages MetaCube uses to install and run Web Explorer. Furthermore, clients' Web browsers may also require configuration to accommodate Web Explorer.

To use MetaCube 4.0, new security features for users must be defined using Secure Warehouse. Because of MetaCube's new focus on system security, additional procedures are necessary, during the installation process—first to grant access to MetaCube's administrative tools, such as Secure Warehouse, Warehouse Manager, and Agent Administrator, and then to define user access to the database. Moreover, for every user running Web Explorer, default connection information must be defined.

Besides installing the MetaCube software, some third-party software is required if a site uses a three-tier architecture.

Release 4.01

This release adds support for the Netscape Web browser for MetaCube Web Explorer. There are no changes to functions or capabilities in this release.

Release 4.02

This release adds support for Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options for the MetaCube product suite. The release also contains enhancements and improvements to existing features.

Sampling

Sampling is implemented on a query-by-query basis. The user interface for Client/Server Explorer and Web Explorer allows the setting of Sampling Accuracy for each query. The Sampling Accuracy setting for a query is saved with the query so that it can be applied again when the query is rerun. Reports that contain sampled data are flagged by a label that says `Sampled Data` in the lower right corner.

When connected to Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, the MetaCube Sampling feature takes advantage of that database server's ability to retrieve sampled data; no separate sample tables are required.

Sometimes, the MetaCube Analysis Engine can retrieve results faster and more accurately using aggregate tables rather than sample tables. If that is the case, the report returns aggregated data and is labeled `Aggregate Data` in the lower right corner.

Refer to the *MetaCube Explorer User's Guide* for descriptions and illustrations of the Explorer user interface and reports derived from sampled data.

Display of Measure Icons

The appearance of the DSS System hierarchy display in Explorer is specified and controlled using Warehouse Manager. An enhancement for Release 4.02 affects how attribute and measure icons display. By default, attribute icons display at the top of the tree in Explorer's Ad Hoc tab page, followed underneath by measure icons. This scheme can be reversed, so that measure icons display above the attribute icons.

The *MetaCube Explorer User's Guide* provides illustrations of how the display of the DSS System hierarchy can be changed.

Enhanced Index Support

Support for the GK (generalized-key) index is implemented for data warehouses stored in Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options. Although these enhancements are designed specifically for that database server, they are compatible with all database servers.

To support the GK index feature with Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, the tables in the data warehouse must not contain duplicate keys. To accommodate this, specification of dimension tables using Warehouse Manager is enhanced. Separate dimension element tables (DETs) may be specified. These tables contain the unique values associated with a given dimension and, because neither they nor the dimension table they relate to have duplicate values in the key column, a GK index may be applied to these tables.

The *MetaCube Data Warehouse Administrator's Guide* describes two methods of specifying metadata for dimensions. One method, available in previous releases of MetaCube, specifies metadata for tables in which key columns contain non-unique data. The second method, available with Release 4.02, specifies metadata for tables to which GK indexes may be applied (using DETs). The *MetaCube Data Warehouse Administrator's Guide* includes diagrams comparing the snowflake schemas for DET and non-DET implementations.

Visual Basic API

MetaCube's OLE Automation API is modified to support the software enhancements in Release 4.02 of MetaCube products. New properties are added for the FactTable, Query, and DimensionElement classes of object. Changes related to support of Dimension Element Tables are made for the Dimension and DimensionElement classes.

The Version 4.1 *MetaCube Application Programmer's Manual* includes information on the latest changes to the MetaCube OLE Automation Interface.

New Features for Release 4.1

This section provides information on changes and enhancements to the MetaCube product suite for Release 4.1.

The most significant enhancement for this release is MetaCube support for Oracle databases. MetaCube Agents for Oracle run in a Windows NT environment. To accommodate changes required to run in an Oracle environment, user interfaces for some applications are changed.

User interfaces have also changed to accommodate other enhancements in MetaCube Release 4.1.

MetaCube Support for the Oracle Database

A MetaCube data warehouse may stored in an Oracle database and queried and managed by the MetaCube suite of applications. Modifications are incorporated into the following MetaCube products to support Oracle databases:

- Client/Server Explorer, Web Explorer, and MetaCube for Excel
- Secure Warehouse
- Warehouse Manager
- Agent Administrator

The following Release 4.1 applications cannot be used with Oracle databases:

- Warehouse Optimizer
- SQL Optimizer

When configuring to connect to a data warehouse, a database type parameter must be specified. MetaCube Release 4.1 supports MetaCube Agents for Oracle running in a Windows NT operating system environment.

Secure Warehouse

Because MetaCube supports more than one database, users are configured to connect to a particular type of database. In Secure Warehouse, database type information for active users is reflected in the User Properties dialog box. To manage permissions and properties for users who connect to various databases, database type information must be specified when switching connections in Secure Warehouse.

User properties for the Informix database options PDQ Priority and Data Skip cannot be individually set by users. Using Secure Warehouse, a data warehouse administrator controls these settings. They remain in effect until changed again through Secure Warehouse.

A Release 4.1 enhancement in Secure Warehouse allows a data warehouse administrator to set user properties for many users at a time. By highlighting a group of user icons in the ListView panel of Secure Warehouse, changes or new settings for user properties—such as default DSS System, query auditing, PDQ Priority and Data Skip for Informix databases, QueryBack permissions, mandatory filters, and others—apply to all users selected.

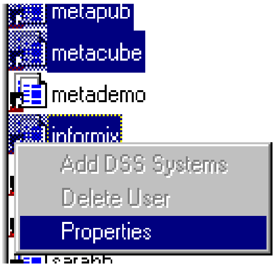


Figure 9
Setting User Properties for Many Users at OneTime

In addition, multiple users may be added to a DSS System at one time.

MetaCube Secure Warehouse is discussed in the *MetaCube Data Warehouse Administrator's Guide*.

Agent Administrator

MetaCube Agent Administrator features improved flexibility for running operating system jobs when using MetaCube Agents for Windows NT. Ordinarily, operating system jobs execute on the server where the MetaCube Agents are running. This is known as a *local* operating system job.

However, operating system jobs originating with MetaCube Agents running on a Windows NT PC can be redirected to execute on a remote UNIX host. This is known as a *remote* operating system job. To redirect an operating system job to a remote UNIX host, the host name as well as user login and password must be provided. Agent Administrator uses this information to send the operating system job to the appropriate UNIX server to execute.

Incremental aggregate jobs can be run for both Informix and Oracle databases. To specify characteristics for the aggregate tables built during the incremental aggregate process, Agent Administrator provides options that are database-specific.

In the job queue display, a status of Deleting may appear after deleting a job from the queue. This indicates that the job is still running; when it completes, it is removed from the job queue.

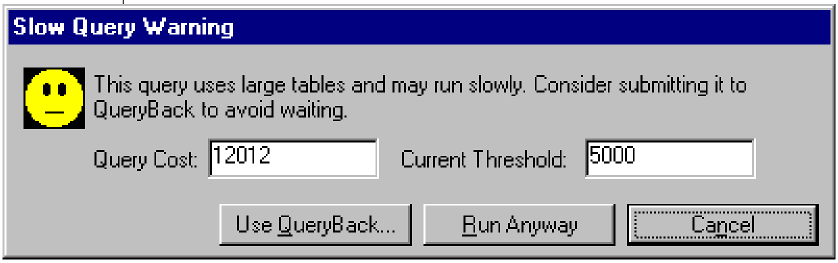
MetaCube Agent Administrator is discussed in the *MetaCube Data Warehouse Administrator's Guide*.

MetaCube Query Applications

Some changes to the MetaCube query applications—Client / Server Explorer, Web Explorer, and MetaCube for Excel—affect report printing, format of reports in Excel, and Snap-Ins for measure calculations.

Slow Query Warning

The slow query warning, displayed when queries will take some time to run, contains information to help users decide whether to continue to run a report, submit it to QueryBack, or cancel.



This query uses large tables and may run slowly. Consider submitting it to QueryBack to avoid waiting.

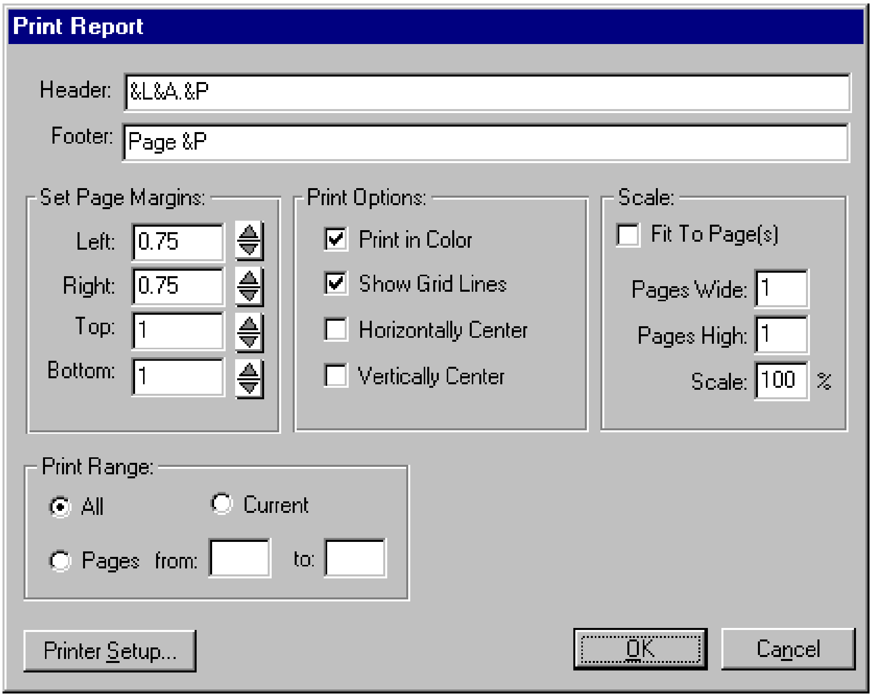
Query Cost: 12012 Current Threshold: 5000

Figure 10
Slow Query Warning

Query Cost is the actual number of rows in the data table to be queried. Current Threshold is a configured value that triggers the display of the slow query warning. By comparing the two values displayed, users can make a judgment as to how long the query might run.

Printing Reports

Improved page printing in Explorer allows better control of how reports appear on a printed page. Users can scale printing of larger reports to “squeeze” more information onto a printed page; this allows MetaCube to accommodate longer or wider reports and limit the number of pages required for print. In addition, multipage report printing options allow selection of a single page, a page range, or the entire report for printing.



Print Report

Header: &L&A.&P

Footer: Page &P

Set Page Margins:
- Left: 0.75
- Right: 0.75
- Top: 1
- Bottom: 1

Print Options:
- ☑ Print in Color
- ☑ Show Grid Lines
- ☐ Horizontally Center
- ☐ Vertically Center

Scale:
- ☐ Fit To Page(s)
- Pages Wide: 1
- Pages High: 1
- Scale: 100 %

Print Range:
- ● All ○ Current
- ○ Pages from: to:

Figure 11
Print Report Dialog Box

Figure 11 shows the Print Report dialog box with new options for scaling a report to the printed page, printing page ranges, and page numbering for multipage reports. For more information on printing options, refer to online help for Explorer and the *MetaCube Explorer User’s Guide*.

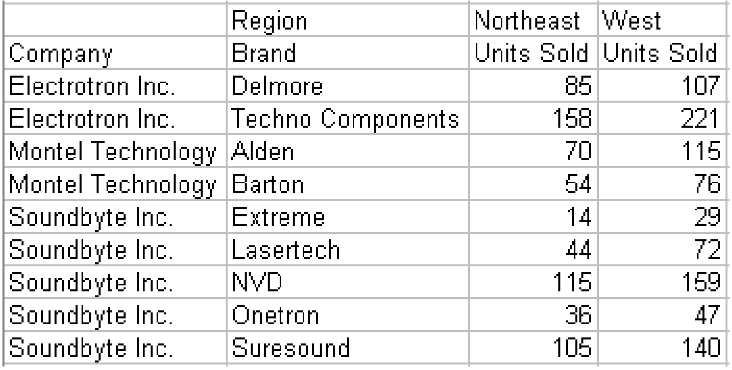
Excel Spreadsheet Headings

When working with Excel spreadsheets, it can be useful to include headings for each row of the spreadsheet. When manipulating data in the spreadsheet, labels on each row assure that information is always identified.

When using MetaCube for Excel to retrieve data, a MetaCube menu option is causes heading labels to appear on each row of the spreadsheet, even though the labels are identical.

This spreadsheet format can be set on a query-by-query basis.

Similarly, users may specify in the Preferences dialog box that duplicate headings should be inserted into the Excel spreadsheet when exporting an Explorer report. When the report appears in Excel, every row is labeled by a row heading. An example of an Excel spreadsheet with duplicate row headings is shown in Figure 12.



	Region	Northeast	West
Company	Brand	Units Sold	Units Sold
Electrotron Inc.	Delmore	85	107
Electrotron Inc.	Techno Components	158	221
Montel Technology	Alden	70	115
Montel Technology	Barton	54	76
Soundbyte Inc.	Extreme	14	29
Soundbyte Inc.	Lasertech	44	72
Soundbyte Inc.	NVD	115	159
Soundbyte Inc.	Onetron	36	47
Soundbyte Inc.	Suresound	105	140

Figure 12
Excel Spreadsheet with Duplicate Headings

Snap-Ins for MetaCube Query Applications

In addition to the default measure calculations provided with every version of Explorer, MetaCube Release 4.1 includes three files that allow users to perform additional calculations. These files are:

- **Mathematical.mcx**
- **BoolAnalysis.mcx**
- **Statistics.mcx**

These files are MetaCube Snap-Ins that can be added to the MetaCube Analysis Engine. Information on how to add Snap-Ins so that measure calculations are available for Explorer or MetaCube for Excel users is contained in the online help for those applications. For a three-tier configuration, MetaCube Snap-Ins are added to the MetaCube Analysis Engine running on the middle tier accessed by Web Explorer users.

Detailed information on measure calculations and, specifically, the calculations performed by the Release 4.1 Snap-Ins is contained in the *MetaCube Explorer User's Guide* and the *MetaCube for Excel User's Guide*.

Visual Basic API

The following properties are added to the FactTables, Aggregates, and Samples classes of object in the MetaCube OLE Automation API for the MetaCube Analysis Engine.

Property	**Description/Example**
DefaultCost	Long integer. The default cost of the corresponding table. Read-only property. C = MyFactTable.DefaultCost
DefaultRowCount	Long integer. The default row count of the corresponding table. Read-only property. R = MyFactTable.DefaultRowCount

Print Manuals for Release 4.1

Listed below are the titles and a short explanation of all the manuals contained in the documentation set for the MetaCube 4.1 Release.

- *Introduction to New Features*. This manual.
- *MetaCube Explorer User's Guide*. Written for people responsible for analyzing data about their company's business. It describes the features of MetaCube Explorer and MetaCube Web Explorer for querying a MetaCube data warehouse.

- *MetaCube for Excel User’s Guide*. 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 to obtain spreadsheet or PivotTable reports.
- *MetaCube Data Warehouse Administrator’s Guide*. Revised for Release 4.1; written for the data warehouse administrator. It describes the overall process of developing a data warehouse and it introduces the tools for managing a data warehouse—MetaCube Secure Warehouse, MetaCube Warehouse Manager, MetaCube Agent Administrator, including Web Publisher, and MetaCube Warehouse Optimizer.
- *MetaCube Application Programmer’s Manual*. Written for a programmer who will implement 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, MetaCube Web 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 C++ code for customized measure calculations.
- *MetaCube SQL Optimizer User’s Guide*. Describes how to use the MetaCube SQL Optimizer for connecting non-MetaCube 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*. Describes how to install and configure the MetaCube software components on both the server and on PCs.

New Features for Release 4.2

This section provides information on changes and enhancements to the MetaCube product suite for Release 4.2.

The most significant enhancement for this release is MetaCube support for Red Brick Warehouse. MetaCube Agents for Red Brick Warehouse run on a Windows NT environment. MetaCube applications allow specification of Red Brick as the database type. Other changes in MetaCube applications implement support for Red Brick Warehouse; see "Features Related to Red Brick Warehouse Support" on page 41.

In addition, MetaCube supports a set of Red Brick extensions to SQL, referred to as RISQL functions, that are used specifically when querying a Red Brick data warehouse. MetaCube Warehouse Manager allows specification of calculated measures that use RISQL functions; see "Use of RISQL Functions" on page 39. RISQL extensions are described in Red Brick documentation; see the *SQL Reference Guide* for RISQL syntax and the *SQL Self-Study Guide* for examples and usage.

Release 4.2 includes enhanced support for semi-additive measures—measures that track events or occurrences—and time-balanced measures—used to store non-cumulative or beginning and ending data. Warehouse Manager allows specification of semi-additive and time-balanced measures. See "Semi-Additive and Time-Balanced Measures" on page 42.

Oracle Database Support

MetaCube Release 4.2 supports MetaCube Agents for Oracle running in a UNIX operating system environment.

SQL Optimizer, which allows non-MetaCube SQL-generating query applications to run against a MetaCube data warehouse, is supported for data warehouses stored in Oracle databases running in a Windows NT environment.

Database-specific versions of SQL Optimizer are:

- **mcifmxdr.dll**—for Informix Dynamic Server
- **mcoradr.dll**—for Oracle databases

These MetaCube drivers are installed into the System or System32 folder.

The filename of the database-specific driver you will use must be specified in the **metacube.ini** file, in the [GENERAL] section, as shown in this example.

```
[GENERAL]
.
.
.
InformixPassThroughDriver=iclit09a.dll
OraclePassThroughDriver=sqora32.dll
```

This example shows the names of the Informix and Oracle passthrough drivers.

Warehouse Optimizer Index Recommendation

Warehouse Optimizer is enhanced as follows:

- Performs aggregate recommendations for data warehouses stored in an Oracle database
- Recommends an optimal indexing strategy that takes advantage of the GK (generalized-key) index feature for Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options.

Important: *Index recommendation is only available for Informix databases and* MetaCube *schemas created with release 4.02 and later. GK (generalized-key) index recommendation is only available for tables with primary keys. When recommending indexes for Informix Dynamic Server without the Advanced Decision Support and Extended Parallel options or* MetaCube *schemas created with release 4.01 and prior, the option to recommend GK indexes cannot be used.*

Optimizing Performance With Indexes

Database indexes work very similarly to index entries in the back of a book. For instance, you create an index, called **Prod_name_ix**, which includes the **Product_name** column in the **Product** dimension table. Each unique value in the **Product_name** column has an entry in the **Prod_name_ix** index with a pointer to any row in the **Product** dimension table that contains an instance of that value.

If you run a query requesting the names of all the customers who purchased Come N Get It Dog Food (a **Product_name** value) and the amount of dog food purchased, without creating the **Prod_name_ix** index, the MetaCube Analysis Engine searches every row of the **Product** dimension table for matches. If, however, you run the same query after you have created the **Prod_name_ix** index, the MetaCube Analysis Engine looks up Come N Get It Dog Food in the index and follows the pointers to find the rows that contain that value. This means that the only rows accessed by the MetaCube Analysis Engine are the ones that contain information requested in the query, greatly decreasing processing time.

However, when an index is used to scan a table, the records are not scanned sequentially. This increases the time needed to process the query. Therefore an index will not be used unless it saves a substantial amount of time. For instance an index might be used if only a small number of records must be scanned, as in the preceding query example. For an index to be used in processing a query, each column included in the index must have a corresponding, highly selective filter in the query.

Warehouse Optimizer analyzes *audit data*—details of the information requested in user queries (especially filter information)—to determine the best indexing strategy for your data warehouse. To help you make an informed decision about index creation, Warehouse Optimizer displays information on the indexes it recommends. Once you choose indexes, Warehouse Optimizer can create the index.

Warehouse Optimizer recommends the following types of index:

- Local index

  In a local index the column included in the index and the rows the index points to are in the same table. For instance, you might have a local index, called **Brand_id_ix**. The index contains an entry for each unique value in the **Brand_id** column of the **Product** dimension table. Each entry in the index points to the rows in the **Product** dimension table that contain that value.

  MetaCube query optimizer can utilize more than one local index for each table included in a query. Combinations of single column local indexes can be used to process the same queries as multicolumn local indexes, providing the flexibility to process more queries than a multicolumn local index. Therefore, only single column local indexes are recommended by Warehouse Optimizer.

- GK (generalized-key) index (only supported for Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options)

  There are two types of GK index recommended by Warehouse Optimizer. These are the most useful in optimizing MetaCube queries.

  - ❑ FCJ **(foreign column join) index**. In this index, the column included in the index is in a table that is related to the indexed table by a foreign key column (join column). The index pointers refer to values in the indexed table. For instance, you might have a **Sales** fact table with a foreign key column, **Product_id**. **Product_id** is the primary key column in the **Product** dimension table. There is a column called **Brand_name** in the **Product** dimension table. You might have an FCJ index, called **GK_brand_ix**, that indexes the **Brand_name** column of the **Product** dimension table and points to rows that contain the associated **Product_id** values in the **Sales** fact table.
  - ❑ MTJ **(multitable join) index**. This index is similar to an FCJ index, but the unique column values can come from *multiple* tables, with each table being related to the indexed table by join columns.

    MTJ indexes are very powerful, but they can not be used to process as many different queries as FCJ indexes.

    Since MTJ indexes are expensive to maintain and usually require large storage space, they are recommended only in cases where frequently run queries involve a set of filter combinations.

  MetaCube query optimizer can utilize more than one GK index for a single query, but, unlike local indexes, only one GK index can be applied to each table included in the query.

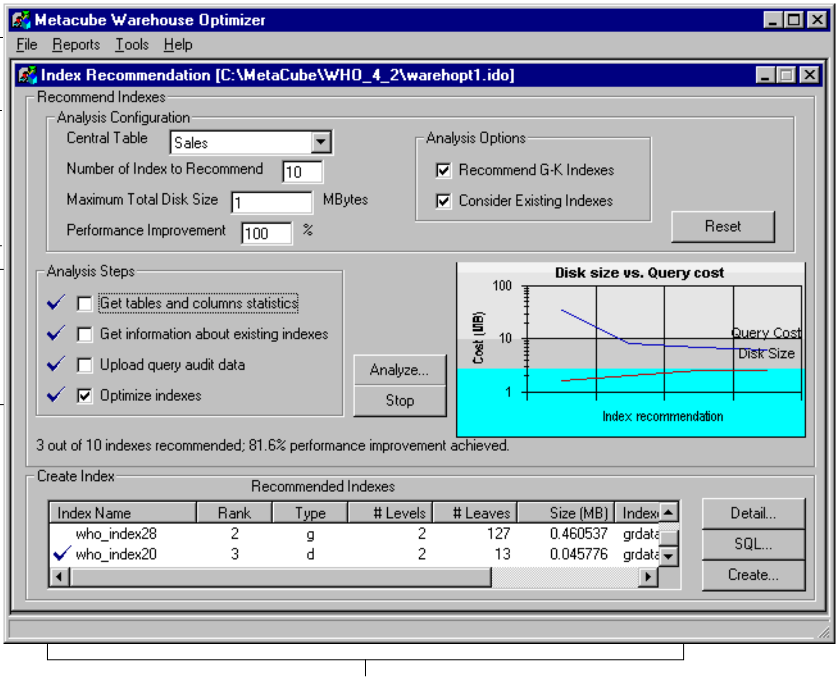
All indexes are stored in a B-tree structure. An index can have one of the following types of pointer:

- **Bitmap**. A bit is associated with each row of the indexed table, indicating if the row contains the index entry value. Bitmaps are used for indexes that have few unique column values.
- **Rid (rowid)**. Each index entry has a corresponding list of rowids that points to the rows in the indexed table that contain that value.

For more information on indexes refer to the *Administrator's Guide for Informix Dynamic Server* and the *Performance Guide for Informix Dynamic Server*.

After connecting to the database, the **Index Recommendation** option in the **Analysis** menu allows you to access the Warehouse Optimizer index analysis feature. The Index Recommendation window is shown in Figure 13.

Figure 13
Index Recommendation Window



Central Table: Sales

Number of Index to Recommend: 10

Maximum Total Disk Size: 1 MBytes

Performance Improvement: 100 %

☑ Recommend G-K Indexes

☑ Consider Existing Indexes

✓ □ Get tables and columns statistics

✓ □ Get information about existing indexes

✓ □ Upload query audit data

✓ ☑ Optimize indexes

3 out of 10 indexes recommended; 81.6% performance improvement achieved.

	Index Name	Rank	Type	# Levels	# Leaves	Size (MB)	Index
	who_index28	2	g	2	127	0.460537	grdata
✓	who_index20	3	d	2	13	0.045776	grdata

Menu bar

Analysis Configuration group box (displays information on the analysis configuration)

Analysis Steps group box (displays information about the analysis stage)

Recommend Indexes
(displays details of recommended indexes)

Using Warehouse Optimizer to Recommend Indexes

Index recommendation consists of three main stages:

- Configuring the analysis
- Recommending indexes
- Creating indexes

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 with the **.ido** filename extension; you can reopen this file at any time to continue, complete, or change the analysis.

Configuring the Analysis

You must specify values for the following aspects of the analysis:

- Choose a central fact table
- Set parameters for index recommendation
- Select analysis options

Choose a Central Fact Table

When you select a central fact table on which to run the analysis, Warehouse Optimizer includes in the analysis the fact table and all the surrounding dimension, dimension element, attribute, and aggregate tables.

Set Parameters for Index Recommendation

There are three parameters for index recommendation:

- Number of indexes to recommend
- Maximum disk space to use for index storage
- Performance improvement desired

When you open a new index analysis file, Warehouse Optimizer displays default values for the following:

- Number of indexes to recommend
- Maximum total disk size in MBytes
- Performance improvement

These defaults can be changed. When you run the index analysis, Warehouse Optimizer stops recommending indexes when it reaches one of these values.

Select Analysis Options

There are two analysis options available:

- **Recommend** GK **indexes.** GK indexes are supported only for Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, MetaCube schemas created with release 4.02 and later, and tables with primary keys. This option is disabled for databases other than Informix Dynamic Server 8.x. To recommend indexes for MetaCube schemas created with release 4.01 and prior, the option to recommend GK indexes cannot be used.
- **Consider existing indexes.** If you choose to consider existing indexes, Warehouse Optimizer recommends new indexes based on how the data warehouse performs when it uses the existing indexes. If you do not select this option, Warehouse Optimizer does not take into account the existing indexes and may recommend indexes that already exist.

Recommending Indexes

There are several steps involved in recommending indexes:

- Get table and column statistics
- Get information about existing indexes
- Upload query audit data
- Optimize indexes

Get Table and Column Statistics

Warehouse Optimizer queries the database to get the following statistics:

- The size of each fact, dimension, dimension element, attribute, and aggregate table, based on the number of rows and the size of each row in each table
- The number of distinct values in each column and the minimum and maximum values for each numeric column

Get Information About Existing Indexes

Warehouse Optimizer queries the database to get the following information about existing indexes:

- For local indexes, information about index components

- For GK indexes, the definition of the index is parsed to retrieve information about the index components involved in joins

Upload Query Audit Data

For information on query audit data refer to the *MetaCube Data Warehouse Administrator's Guide*.

Optimize Indexes

Warehouse Optimizer reconstructs the SQL for the audited queries to obtain the selected components (from the SELECT clause), the join and filter components (from the WHERE clause), and the tables (from the FROM clause) included in the query. Warehouse Optimizer analyzes all of this information to generate plausible indexes and estimates the size of each plausible index.

Warehouse Optimizer then determines if each plausible index can be applied to the audited queries. An index can be applied to a query if the cost of accessing the table through the index is less than the cost of scanning the entire table. Warehouse Optimizer then compares the original query cost for each audited query to the query cost if that particular index is created.

Warehouse Optimizer calculates the indexes that achieve the greatest improvement in query performance and produces a list of index recommendations based on the parameter constraints you provided in analysis configuration.

Creating Indexes

The list of recommended indexes includes the following detailed information about each index:

- **Index Name**—the name of the index
- **Rank**—the place the index holds in the ordered list of indexes; rank 1 provides the greatest performance improvement
- **Type**—lists one of the following types:
    - **D**—local rid (row id)
    - **d**—local bitmap
    - **G**—GK rid (row id)
    - **g**—GK bitmap

- **# Levels**—the number of levels in the B-tree (GK indexes only)
- **# Leaves**—the number of leaves in the B-tree (GK indexes only)
- **Size (**MB**)**—the amount of disk space, in megabytes, needed to store the index
- **Indexed Table**—the table to which the index points
- **% Improvement**—the percentage of improvement in query performance

Warehouse Optimizer displays the following information about any index in the list:

- **Index Components**—the tables and columns whose unique values make up the index entries
- **Name**—the name of the index; this name can be changed
- **Rank**—the place the index holds in the ordered list of indexes
- **Index Type**—lists one of the following types:
  - **D**—local rid (row id)
  - **d**—local bitmap
  - **G**—GK rid (rowid)
  - **g**—GK bitmap
- **Index on Table**—the table to which the index points
- **Index Size (in Bytes)**—the amount of disk space, in bytes, needed to store the index
- **Total Disk Size for Indexes (in Bytes)**—the amount of disk space, in bytes, needed to store this index and all higher ranking indexes
- **Averge Query Cost**—the average cost to run a query based on the creation of this index and all higher ranking indexes; this cost is the average number of rows accessed to perform a query

From the list of recommended indexes, you specify the indexes you want to create. Although you can randomly select indexes from the list of recommendations, you should select indexes in rank order, since the performance improvement listed for each index is based on the assumption that all higher ranking indexes are built. If, for example, you choose only the indexes ranked 1 and 3, the performance improvement listed for the index ranked 3 is no longer accurate because it is based on the index ranked 2 being created.

The index creation fails if one of the following conditions exists:

- There is an existing index with the same name
- There is an existing index with the same definition
- For a GK index, a component table does not have a primary key and Warehouse Optimizer cannot alter the table to include a primary key
- For a GK index, a component table or the indexed table is not static and Warehouse Optimizer cannot alter the table to make it static

Once you create indexes, these are indicated when you view Warehouse Optimizer. You may reselect indexes from the recommendations at any time.

Updating Recommendations

You might want to update the recommendations made by Warehouse Optimizer from time to time. For instance, after collecting new query audit data, rerunning Warehouse Optimizer index recommendation updates the recommendations based on the data stored in the audit tables. Warehouse Optimizer only reruns the analysis stages you choose and any dependent analysis stages. In the case of uploading new query audit data, Warehouse Optimizer does not need to update table and column statistics or information on existing indexes, so it skips these stages.

Important: Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options *requires static tables for GK indexes. If you update your data, you must re-create any GK indexes. Local indexes are updated automatically when new data is uploaded.*

Update your index recommendations with Warehouse Optimizer to keep your data warehouse running as efficiently as possible.

Red Brick Warehouse Support

Red Brick Warehouse can be queried and managed using the MetaCube suite of products. When connecting, MetaCube users can specify Red Brick as the database type. Using Secure Warehouse, data warehouse administrators may assign user permissions and DSS Systems for a Red Brick Warehouse. Agent Administrator also creates aggregate and sample tables for a Red Brick Warehouse. Table create options for Red Brick Warehouse are supported for building incremental aggregate tables.

Use of RISQL Functions

An important aspect of using Red Brick Warehouse is the ability to use the Red Brick SQL extensions, known as RISQL functions.

When connected to Red Brick Warehouse, MetaCube supports the use of the RISQL functions. Warehouse Manager displays the RISQL functions in the Aggregations tab of the Measure editing pane.

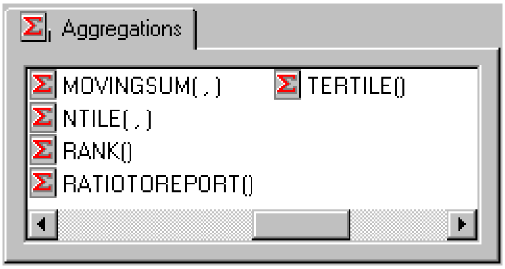


Figure 14
RISQL Functions in the Aggregations Tab

Using Warehouse Manager, RISQL functions may be used to specify MetaCube calculated measures in the Measure editing pane. RISQL function names are viewable in the Measure editing pane only when connected to Red Brick Warehouse.

This example shows the generalized syntax for a calculated measure that uses a RISQL function:

```
RISQL_FUNCTION(AGGREGATE_FUNCTION(COLUMN('column_name')))
```

In this example:

RISQL_FUNCTION is the name of a Red Brick RISQL function.

AGGREGATE_FUNCTION is a standard SQL aggregation function, such as SUM.

column_name is the name of the column containing the measure data on which the calculation is performed.

When specifying a calculated measure using a RISQL function, you must use the name of a column, as displayed in the Columns tab of the Measure editing pane; the column name definition uses the keyword COLUMN. Do not use the representation of a measure, which uses the keyword FACT.

Features Related to Red Brick Warehouse Support

Some interface changes in Secure Warehouse reflect Red Brick Warehouse support. A data warehouse administrator may assign user access and user permissions for DSS Systems in a Red Brick Warehouse. Also, the Switch Connection function allows administrative users to change the current connection to a Red Brick Warehouse connection.

Agent Administrator provides the following Red Brick Warehouse-specific options for running incremental aggregate jobs:

- Maximum Segments (required)
- Maximum Rows / Segment (required)
- Additional Options (optional: up to 10 characters)

These options can be set when running an incremental aggregate job and can be viewed in the Job Queue dialog box for a Red Brick Warehouse incremental aggregate job submitted for processing.

For Red Brick Warehouse users who wish to specify an OTHER bucket as a custom bucket to apply to a query, the Red Brick Warehouse cross join feature must be enabled. In the rbw.config, enable the cross join feature as shown in this example:

```
#Enable cross join
#OPTION CROSS_JOIN {OFF | ON}
OPTION CROSS_JOIN ON
```

Custom buckets are discussed in the *MetaCube Explorer User's Guide*.

Agent Administrator

A new feature of Agent Administrator allows an administrator to specify the amount of time to wait when refreshing the job queue status page or requesting specific information on a single job's status. When many agent jobs are currently executing, availability of job status information may be negatively impacted. Specifying a wait time for retrieving status information causes Agent Administrator to retry the request for the specified number of seconds before returning an error that the table containing status information is unavailable.

Semi-Additive and Time-Balanced Measures

With Release 4.2, the MetaCube analysis engine supports the use of time-balanced measures in a data warehouse. In addition, support for semi-additive measures defined using the aggregate function COUNT(DISTINCT) is enhanced.

Semi-Additive Measures

Semi-additive measures track unique occurrences. For example, a fact table might contain the names all the products sold for a given time period. To obtain a count of the individual products sold (not a count of the total number of all products), a semi-additive measure is used. In a query, a semi-additive measure causes generation of SQL that identifies and counts unique occurrences, eliminating duplicates.

When using semi-additive measures, it is important that dimension table hierarchies support the various elements for which unique counts are desired. For example, to count occurrences by time periods, such as week, month, or quarter, the fact and aggregate tables must differentiate the data entries by the relevant time period.

When measures are semi-additive, they cannot be aggregated using the SUM( ) function. Instead, data for a semi-additive measure is summed by unique values. For example, if an aggregate table contains monthly data and a semi-additive measure is defined for customer ID, customer ID is part of the GROUP BY clause and totals for each unique customer ID are stored in the aggregate.

In Warehouse Manager, semi-additive measures are specified with the aggregate function COUNT(DISTINCT) using the following syntax:

```
COUNT(DISTINCT COLUMN('column_name') )
```

In this example, *column_name* identifies the column holding the measure data to be counted for unique values.

Prior releases of MetaCube support measures specified using the COUNT(DISTINCT) aggregate function; however these measures could not be specified for aggregate tables. With Release 4.2, semi-additive measures—that is, measures defined using the COUNT(DISTINCT) aggregate function—may be included in aggregate tables.

Time-Balanced Measures

A time-balanced measure is a type of measure that tracks balances, such as inventory figures or bank balances. Time-balanced measures identify specific starting or ending values.

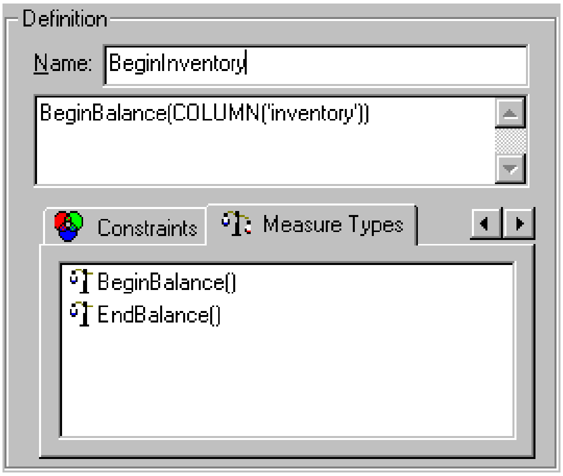
A query containing a time-balanced measure must include at least one attribute from the time dimension. The time attribute indicates the time period for which the beginning or ending balance is to be retrieved. In the time dimension table, itself, the time period codes must occur in ascending order; that is, earlier dates must have lower code values and later dates must have higher code values.

If a query contains more than one attribute from the time dimension, the time-balanced measure value is returned for the lowest level time attribute, regardless of the order of the attributes in the query. For example, if both month and year attributes appear in a query, the time-balanced measure is retrieved for the month.

Time-balanced measures are not summed by time dimension. For all other dimensions they are additive and can be summed. For example, the month-end inventory cannot be calculated by summing all inventory figures for the month; however, the current inventory for all products can be calculated by summing the individual inventory values for each product.

Important: *Time-balanced measures should not be included in aggregate tables; only additive measures should be specified and included in aggregate tables. Also, time-balanced measures will not give meaningful results from sample tables.*

In Warehouse Manager, a new tab in the Measure editing pane, called Measure Types, allows specification of time-balanced measures.



Name: BeginInventory

BeginBalance(COLUMN('inventory'))

Figure 15
Time-Balanced Measure Specifiers in the Measure Types Tab

BeginBalance( ) is used to specify a measure that returns the value associated with the first date in the specified time period; for example, the first day of a month.

```
BeginBalance(COLUMN('inventory'))
```

In this example, the measure retrieves the inventory value stored on the first date of the time period specified by the time attribute included in the query; it might be the first day of a month.

EndBalance( ) functions in a similar way and is used to specify a measure that returns the value associated with the last date in the specified time period—for example, the last day of a quarter.

To use the BeginBalance( ) or EndBalance( ) specifier, data must exist for the exact date specified; that is, there must be a valid data entry in the data table for the beginning or ending date of the time period. Beginning and ending date codes are retrieved from the time dimension and the corresponding data value is retrieved from the fact table. If the date does not exist in the data table or if measure data is missing, `NULL` is returned in the results.

Demonstration Database

The MetaCube demonstration database is enhanced to illustrate the use of semi-additive measures. For Red Brick Warehouse, several measures defined using RISQL functions are included in the demonstration database. Completed QueryBack jobs are included in the QueryBack Tab page.

An inventory record is included in the database that tracks beginning and ending inventories for every individual store by product. In the Sales Transactions data source, semi-additive measures that retrieve beginning and ending balances are defined using Begin Balance( ) and End Balance( ) specifiers for the inventory data.

One stored query, Count Distinct Product Brand City, uses a semi-additive measure that identifies the unique count of products sold, by city for all brands. This query retrieves results from an aggregate table, Sales_agg10, that is included in the MetaCube demonstration database for this release.

Formula Snap-In

An additional measure calculation snap-in file is provided. The name of the file is **formula.mcx** and you or your data warehouse administrator may snap it into the MetaCube Analysis Engine.

Important: *Unlike other calculated measures, you cannot use any of the measures included in the formula as the basis for the formula. For example, in the demonstration database, for a formula based on Units Sold, use any other measure, for example, Gross Revenue, as the measure on which to build the formula. Because of this, you will need to edit the name of the formula measure, since it will reflect a measure that is unrelated to the values calculated.*

The Formula snap-in allows you to specify any calculation that can be notated as a formula using the arithmetic operators. You may indicate precedence using parentheses and the formula may use constants.

Any measure in the DSS System may be used in the formula, regardless of the data source; however, any measure specified in the formula also must be included in the query itself.

When specifying a formula, you provide two parameters:

- the formula itself
- the format used to display the results of the calculation in the report

Both parameters are enclosed in double quotes. For example, in the demonstration database, this formula:

```
"(Units Sold - Units Sold_1) / Units Sold_1"
```

where Units Sold is contained in the Sales Transactions data source and Units Sold_1 is contained in the Competitive Data data source, produces a result that indicates the difference in unit sales compared with the competitors. Adding the absolute function, this formula produces a result that indicates whether sales are greater than or less than the competitor expressed as a percentage:

```
"abs(Units Sold - Units Sold_1) / Units Sold_1*100"
```

You must enter the numeric format for the formula manually; that is, the Format button in the Measure Calculation dialog box does not function for this snap-in. Enclose the format string in double quotes.

These functions are supported for formulas:

abs	Absolute value
sin	Sine
arcsin	ArcSine
sinh	Hyperbolic sine
cos	Cosine
arccos	ArcCosine
cosh	Hyperbolic cosine
tan	Tangent
arctan	ArcTangent
tanh	Hyperbolic tangent
DegToRad	Degrees to Radian
sqrt	Square root
log	Log base 10
ln	Natural log

Print Manuals for Release 4.2

Listed here are the titles and a short explanation of new manuals released with MetaCube Release 4.2.

- *Introduction to New Features*. This manual.
- *MetaCube Installation and Configuration Guide*. Describes how to install and configure the MetaCube software components on both the server and on PCs.

Other manuals included in this MetaCube release 4.2 are described in the section "Print Manuals for Release 4.1" on page 28.

Index