Introducing DataQuant
IBM DataQuant for z/OS and Multiplatforms
Version 1 Release 2

Introducing DataQuant

IBM
This edition applies to Version 1 Release 2 of IBM DataQuant for z/OS, program number 5697-N64. It also applies to program number 5724-R90, which applies to Version 1 Release 2 of the following products: IBM DataQuant for Multiplatforms; IBM DataQuant for DB2 Warehouse Starter Edition; and IBM DataQuant for DB2 Warehouse Intermediate Edition. This information applies to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GC19-1254-12.

© Copyright IBM Corporation 2007.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© Rocket Software Inc. 2007, 2014
Contents

About this information ........................................ v i i
Service updates and support information ....................... v i i
How to send your comments ...................................... v i i

Chapter 1. Summary of changes ................................ 1

Chapter 2. DataQuant at a glance ................................ 9
What is DataQuant? .............................................. 9
The business value of DataQuant ............................... 10
DataQuant in the real world .................................... 12
DataQuant for WebSphere versus DataQuant for Workstation ..................................................... 13

Chapter 3. Robust business intelligence function ............... 15
Multiplatform data access ...................................... 15
Queries .......................................................... 15
  Relational queries ......................................... 16
  OLAP queries ............................................... 18
  Prompt hierarchies ........................................ 20
  Analytical queries ........................................ 20
  Forecasts .................................................... 21
Reports ......................................................... 22
  Visual reports ............................................ 23
  Standard tabular reports ................................. 27
  Saving formats as templates .............................. 27
  File formats and delivery ............................... 27
Dashboards ..................................................... 28
Analysis functions ........................................... 30
  Arithmetic functions .................................... 31
  Color functions .......................................... 31
  Conversion functions ................................... 32
  Data formatting functions ............................... 33
  Date and time functions ................................. 34
  Hierarchical functions .................................. 35
  Information functions .................................. 36
  Logical functions ....................................... 36
  Math and trigonometric functions ....................... 37
  Measured functions ...................................... 38
  Security functions ...................................... 38
  Spatial functions ........................................ 39
  Statistical functions ................................... 39
  Text functions .......................................... 40
  Translation functions .................................. 41
Data editing functions ...................................... 41
Application development ..................................... 41
Procedures ...................................................... 42

Chapter 4. Ease of use ......................................... 43
Zero-coding architecture ...................................... 43
Role-based user interfaces .................................. 43
Customizable workspaces .................................... 45
Support for virtual data sources ............................... 45
Intuitive navigation ......................................... 45
Graphical, drag-and-drop object development ................ 46
Safety net for database modifications ......................... 46

© Copyright IBM Corp. 2007
About this information

IBM® DataQuant is a business intelligence tool available on z/OS®, Linux, UNIX, Windows, and iSeries® platforms. You can use DataQuant to quickly and easily query, format, and edit data or develop visual reports (for example, charts, graphs, or maps), interactive dashboards, and information portals.

These topics are designed for those evaluating DataQuant, or for users who would like a comprehensive overview of product features and functions before beginning to use the product.

Always check the DB2® and IMS™ Tools Library page for the most current version of this publication:

www.ibm.com/software/data/db2imstools/db2tools-library.html

Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the following web page:


How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other documentation, use either of the following options:

• Use the online reader comment form, which is located at:
  http://www.ibm.com/software/data/rcf

• Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of your product, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
Chapter 1. Summary of changes

Several enhancements have been added to DataQuant Version 1.2.

These features enable users to add virtual data sources, schedule jobs, distribute visual reports and visual dashboards in multiple languages, view dashboards using Flash, access legacy QMF™ for Windows APIs, assign expiration schedules to dashboards and reports, and view favorites and most frequently used objects in a personal view. Support for Object REXX has been reintroduced, and the WebSphere® interface has been redesigned to provide a more rich-client user experience.

Enhancements in DataQuant Version 1.2 Fix Pack 20

- New database support
  DataQuant supports connections to Teradata database.

- Database-based security for Oracle-based repositories
  DataQuant provides the ability to apply database-based security option for repositories that are hosted on Oracle databases.

- Using the Secure Sockets Layer (SSL) protocol
  DataQuant provides the ability to use LDAP over SSL security option for repositories.

- Using DB2 special registers
  DataQuant provides the capability to edit database tables using DB2 special registers.

- Filtering table columns in the Prompted Query editor
  DataQuant provides the possibility to filter the table columns in the Add Columns dialog of the Prompted Query editor. This allows to find a specific column easily and include it to the query results.

- TIMESTAMP format of dates in the object tracking tables for DB2
  The TIMESTAMP format is supported for dates together with the numerical format in the RSIBOBJ_ACTIVITY_DTL and RSIBOBJ_ACTIVITY_SUMM tables.

Enhancements in DataQuant Version 1.2 Fix Pack 19

- Improved cell formatting for query result sets exported to Excel
  This enhancement ensures that the query result set is exported to Excel correctly if each Excel formula starts with the equals sign in the query.

Enhancements in DataQuant Version 1.2 Fix Pack 18

- Improved cell formatting for data exported from DataQuant to Excel
  This enhancement addresses data characterization issues that can occur when exporting data from DataQuant to Excel. Prior to this enhancement, it was possible that character data from the database might contain string patterns that had a special meaning to Excel. When this occurred, Excel processed the string sequence by flagging the cell with a warning symbol that the string might be formatted incorrectly. This enhancement ensures that character columns arrive in Excel with the Format Cells> Number set to Text.

- Capability to control display options for the Progress dialog
A new parameter has been added to the General preferences window. The new parameter, called **Always run in background**, provides users with the option to hide or display the Progress dialog from view for long running operations.

- **Improved result set scrolling**
  Logic has been added to enhance the scrolling behavior of the results grid in DataQuant for WebSphere. With this enhancement when users scroll the result grid, DataQuant for WebSphere requests data from the server and displays results dynamically. This enhancement has been applied to query results, quick reports, and forecasts.

- **Context menu support for opening database tables in the Visual Query Editor**
  With this enhancement users can open database tables (from any data source) in the Visual Query Editor. A Visual Query Editor option has been added to the context menu so that when a user right-clicks while in the Repository Explorer view they now have the option to select *Open with > Visual Query Editor*.

- **Option to retain query sort order added**
  A new variable controlling whether to retain a user defined sort order for queries has been added to Global Variables preferences window. The new global variable is named **DSQQW_QUERY_PRESERVE_SORT**. If users set this variable to 0, the sort order is not preserved for the query. If the user sets this variable to 1, the query sort order is preserved and is applied whenever the query is run.

### Enhancements in DataQuant Version 1.2 Fix Pack 17

- **Export multiple results set**
  DataQuant includes the capability to export queries that contain multiple result sets.

- **Export to Excel**
  DataQuant includes the capability to export query result sets to an Excel data sheet.

- **New schedule options**
  DataQuant includes the capability to specify hourly schedule for cache resource limits, scheduled tasks, and expiration schedules.

- **Table height limit**
  DataQuant includes the capability to set a table height limit in Flash rendering mode.

- **Table editor in DataQuant for WebSphere**
  Database tables can be edited in DataQuant for WebSphere.

- **ScrollBar automode**
  The Texbox, EmbeddedScene, and EmbeddedContent objects support “Auto” ScrollBar mode. In this mode, both scrollbars appear only if the content of a component exceeds the size of this component.

- **Ability to specify SMTP servers for the whole application**
  DataQuant includes the capability to define parameters for commonly used SMTP servers once for the whole application.

- **Ability to add custom JavaScript functions to visual projects**
  DataQuant includes the capability to use custom JavaScript functions in visual projects with the help of JavaScript Modules.

### Enhancements in DataQuant Version 1.2 Fix Pack 16

- **Customized repository caching**
DataQuant provides users with the capability to implement custom cache validation settings for content cache, server metadata cache, capabilities, structure cache, and table metadata cache.

- **Read-only visual dashboards**
  DataQuant includes the capability to compile read-only versions of your visual dashboards in a project and make them available to users.

- **Support for MySQL parser**
  DataQuant supports the MySQL parser.

- **Single sign-on preference setting enhancements**
  DataQuant includes new preference settings for single sign-on.

- **Drill-down paths**
  DataQuant includes drill-down path functionality. With drill-down paths you can “drill down” or navigate to several different levels of query results. Each level is represented by a different display mode chart, and each display mode chart can be fully customized.

- **Forecasting capabilities**
  DataQuant includes a new forecasting capability. You can use forecasts to help predict future values based on historical results. Forecasting helps you better prepare for changes in economic or competitive conditions by incorporating an analysis of time series historical data.

- **Visual queries**
  DataQuant includes the capability to construct visual queries using Structured Query Language (SQL) statements.

- **Analytical query capabilities**
  DataQuant supports analytical queries. With analytical queries you can combine data from multiple queries (from the same or differing data sources) into one result set. You can add multiple queries and tables to the analytical query tree structure, generating a comprehensive result set that encompasses data from many different sources.

- **Prompt hierarchies**
  DataQuant includes a new type of object called a prompt hierarchy. Prompt hierarchies are a unique type of object that are stored independent of a given query. This allows them to be used by multiple query objects. A prompt hierarchy can present a hierarchy of values, with unlimited levels of detail.

**Enhancements in DataQuant Version 1.2 Fix Pack 15**

- **Repository caching**
  DataQuant supports repository caching. With repository caching configured, the system saves the content of the repository (dashboards, queries, visual reports and so on) to the memory on the workstation client or on the web server (if you are using DataQuant for WebSphere). Subsequently, the next time you access the repository objects, information is retrieved from memory (without sending a request to the repository), thereby reducing the retrieval time.

- **New preference for configuring 24-hour date and time rollover conversion**
  DataQuant includes a new preference setting that allows the user to specify whether to allow Java™ to convert and express 24-hour rollover time as the next day in result sets and visual projects.

- **Additional support for Google Maps**
  DataQuant now supports
  GoogleMapPolyline, GoogleMapPolygon, and GoogleMapObject object types.
• **Active directory single sign-on support** DataQuant now supports Active Directory (AD) single sign-on. Single sign-on is a characteristic of access control that uses centralized authentication servers for determining whether a user’s sign-on can be used to access multiple applications and systems.

**Enhancements in DataQuant Version 1.2 Fix Pack 14**

• **Support for DB2 temporal data in the query diagram and prompted query views**
  DataQuant now supports DB2 temporal data management technology in its query diagram and prompted query views.
  A new **Time Period** button has been added to the Prompted Query and Diagram Query editors. Users can use the button to add a “FOR” specification to a query to support tables that have been configured for temporal data.

• **Support for RUNTSO command**
  Users can also now issue the RUNTSO command to invoke QMF for TSO as a DB2 for z/OS stored procedure. The procedure name is **Q.DSQQMFSP**. Upon invocation, users pass the name of a query or procedure to run on QMF for TSO. Output returns to the workstation in a result set. Up to 21 result sets can be specified for a QMF procedure.

• **Automatic generation of dashboard and report URLs**
  A new Web Link wizard has been added for creating web links to DataQuant objects that can be opened directly in web browsers.

• **Additional reformatting capability for SQL queries**
  A new capability has been added for specifying any integer between 10 and 1000 for the line width of SQL queries. DataQuant uses hardcoded line width of 79 characters. This enhancement enables users to circumvent the way DataQuant reformats a query’s text.

• **Enhanced refresh capabilities for data-driven components**
  A new event action has been added for refreshing a component in a visual dashboard by invalidating a query cache. This event action reruns the query and the results are displayed in the refreshed component.

• **New Launch LOB event action**
  A new capability has been added for opening LOB data from an object of a visual dashboard. The new Launch LOB event action allows you to select an object of a visual dashboard based on the table containing LOB data and display this data. This action can be applied only to the objects that are contained in the data template.

• **JavaScript API support for visual dashboards embedded in a web page**
  A new JavaScript function has been added that allows users to set dashboard global parameters when a dashboard is open in HTML IFRAME via the **GetObject** command.

• **New global variable (DSQEC_RUN_MQ)**
  A new global variable named **DSQEC_RUN_MQ** has been added to control the manner by which statements in a multi-statement query are run.

• **New global variable (DSQAO_HOME_WORKSPACE)**
  A new global variable named **DSQAO_HOME_WORKSPACE** has been added. This new global variable contains the key to the user’s home workspace if it exists.

**Enhancements applied to DataQuant Version 1.2**

• Virtual data sources
Virtual data sources allow you to create virtual tables that can be edited and manipulated by users without altering the original tables' content. In prior versions, users developing dashboards, queries or reports were required to understand the underlying physical tables and views from which their queries drew data. With the introduction of virtual data sources, administrators can now optionally shield their users from the complexities of the underlying database structures, and provide content developers with a simplified data model against which content can be developed. Virtual data sources work by introducing a metadata layer which mediates between an administrator-defined, virtual data source and the underlying data sources that contain the physical tables and views.

This approach provides the following key benefits:

- Users are shielded from the complexities of the underlying data model. Obscure column names and table names can be replaced with human-readable variants and complex table joins can be represented by a single virtual table, much like a database view.

- Administrators can define multiple virtual tables in a single virtual database, each of which draws data from one or more tables within differing data sources.

  For example, one could add three tables from DB2® and two tables from Informix®. To end users, a given virtual data source acts as a single database, allowing end users to author queries against all tables contained within it, despite their underlying sources residing in differing data sources.

- Users are insulated from database schema changes. Database administrators can readily make changes to the underlying database schemas without affecting existing dashboards, queries, or reports, since the virtual data source schema remains unchanged.

**Job scheduler**

DataQuant introduces the ability to schedule a procedure object to be executed on a periodic, recurring basis or at a specific date and time. A DataQuant procedure can perform one or more sequential steps, such as run queries, apply report templates, export reports to PDF files, or send reports via e-mail.

Jobs can be scheduled to run at either of the following locations:

1. Local workstation: DataQuant automatically creates a job with the operating system scheduler, under the profile of the logged on user. All execution takes place at the end user's workstation.

2. DataQuant for WebSphere: DataQuant schedules a job for execution by DataQuant for WebSphere's new built-in job scheduler. In this mode, the job is executed by DataQuant for WebSphere itself, fully independent of the workstation that scheduled it. Using this approach, multiple users may schedule work to be performed on an ad-hoc or periodic basis, without the need to leave their workstations powered up and logged on.

**Support for multi-language visual reports and dashboards**

DataQuant's visual reports and interactive dashboards provide an excellent means of disseminating your business data across the enterprise. For global companies, this often means deploying solutions across differing language regions. While DataQuant is translated into over 20 languages, any text that you add to your reports and dashboards remains in its original form, regardless of region.

DataQuant now includes the ability to author reports and dashboards that will present localized text that matches the region of the user. For example, you can now author a report or dashboard in German that also contains your text in...
French. In this example, German users viewing your report or dashboard see German text, whereas French users accessing the very same solution see your French text. This new feature allows you to define as many ‘string tables’ as you wish within a given visual project, each of which can be assigned to a given regional language.

- **Support for Flash-based dashboards**
  In prior versions, DataQuant dashboards could be deployed using pure HTML, PDF, or optionally via a lightweight browser Applet. DataQuant now introduces the ability to deploy dashboards using Adobe's Flash player. The Flash-based option provides a user experience that is more comparable to a rich application since the dashboard is assembled by the workstation and many dashboard interactions can take place without the need to communicate back to the web server.

- **Introduction of backward-compatible Win32 API**
  DataQuant offers three forms of application programming interfaces (APIs), allowing DataQuant functionality to be optionally embedded within custom applications and third-party software. These include Java APIs, Web Service APIs, and Command Library (command line) APIs.

  Often times the relationships between different tables’ columns is not readily apparent to a given user. Users who are not familiar with a particular set of tables may require additional information about these relationships before they can author join queries, taking up valuable production time. ER diagrams allow experienced administrators to predefine all of the possible column relationships that a user might need to set up their join conditions. At any time, users can view graphical representations of all of these predefined relationships and use them to pick and choose what join conditions they need to retrieve their data. This shortens the learning curve for novice users and streamlines the query authoring process dramatically.

  This new feature introduces a set of Win32 APIs that provide access to DataQuant functionality using a set of functions that are compatible with the former QMF for Windows APIs. This allows in-house applications that were developed using the QMF Win32-based APIs to operate with the newer Java-based DataQuant solution.

- **Data expiration schedules in reports and dashboards**
  In prior releases, DataQuant’s visual reports and dashboards retrieved data from the database when the solution is accessed. For example, a sales report returns sales data that is current at the time of opening the report.

  DataQuant’s new data expiration schedules allow report and dashboard developers to define the point at which new data should be fetched from the database. Expiration schedules can be defined in terms of a time span (such as so many seconds, minutes, hours, and days) or as a regular interval (such as every day at noon or every first Monday of each month). In addition, each query in a report or dashboard can be assigned a distinct expiration schedule.

  This new feature can be utilized in many ways, such as:
  - Defining a sales report that always displays data for the current month. The data is fetched once only, after being accessed for the first time after the new data is available. Subsequent access to the report is satisfied by data held in the report’s data cache. Copies of the report can also be saved against a given month, creating a repository of report snapshots on a monthly basis.
  - Defining a dashboard that refreshes each query according to the underlying validity of the data. For example, book authors never expire whereas transactional records are valid for so many minutes. By pooling the data...
caches at the web application server, concurrent user access to dashboards results in virtually no incremental loading on the underlying data sources that power them.

- **Enhanced DataQuant for WebSphere user interface and capabilities**
  In prior versions, DataQuant for WebSphere's user interface offered a subset of workstation features but using a distinct HTML interface.
  DataQuant's Eclipse-based user interface offers a productive and increasingly familiar environment for both technical and non-technical users alike. DataQuant for WebSphere now provides web users with a user interface that is virtually identical to the workstation product. This retains the benefits of a pure-HTML solution without losing the productivity of a desktop application. Application toolbars, context menus, dialogs, and views remain common between both products, allowing for the seamless transition between each of the solutions.

- **New Personal view**
  DataQuant now includes an optional 'Personal' view that is docked above the workspace or repository explorer in both the workstation and web products.
  This new view offers the following facilities:
  - A 'Recently Used' folder that lists and provides direct access to tables, queries, dashboards, reports, and procedures recently accessed by the user.
  - A 'Favorites' folder within which users can collect their 'favorite' items (tables, queries, reports, or dashboards). Items are added by dragging and dropping or by right-clicking on objects and selecting 'Add to favorites'.
  - A 'Startup' folder that allows users to define items that should be opened whenever the application is launched. Items added to this folder (using the same mechanism as above) are automatically opened and displayed when the user logs into the application.

- **Support for Open Object REXX**
  QMF for TSO/CICS users often use the REXX programming language to incorporate additional programming logic within procedures and forms (classic reports). Prior versions of QMF for Windows supported a number of REXX statements using IBM Object REXX. However, this REXX product was subsequently discontinued and donated to the open source community.

---

1. Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software. Eclipse provides extensible tools and frameworks that span the software development lifecycle, including support for modeling, language development environments for Java, C/C++, PHP and others, testing and performance, business intelligence, rich client applications and embedded development. A large, vibrant ecosystem of major technology vendors, innovative start-ups, universities and research institutions and individuals extend, complement and support the Eclipse Platform. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the Eclipse projects. Full details of Eclipse and the Eclipse Foundation are available at [www.eclipse.org](http://www.eclipse.org).
Chapter 2. DataQuant at a glance

This topic provides an overview of the features and functions that are available in DataQuant and the business value that these functions provide.

What is DataQuant?

DataQuant is a powerful but easy to use business intelligence tool that streamlines all aspects of delivering your business data across the enterprise.

One of your most important assets is the operational data that forms the backbone of your business. Many people at different organizational levels need access to this data, but they might have different security levels as well as different uses for that data. You cannot afford to invest in tools that are difficult to use and maintain, and that require large numbers of staff to keep them running smoothly. A product’s time to value and return on investment affects your organization in a very real way. Excellent time to value can mean the difference between gaining insight you can quickly act upon and getting mired in the details of simply isolating, interpreting, and presenting just the data you need, right when you need it.

DataQuant provides excellent time to value and excellent return on investment by delivering a robust set of query, reporting, dashboard, and analysis functions that are packaged in a very easy-to-use interface. Although many analytical tools require extensive programming and deployment time, DataQuant requires no programming skills to query, format, and edit data or develop visual reports (for example, charts, graphs, or maps), interactive dashboards, and information portals. DataQuant includes a rich graphical palette, as well as dozens of out-of-the-box charts, controls, and graphical objects that you can drag and drop to quickly create dashboards and reports.

Although the cost and complexity of implementing reporting solutions often result in one-size-fits-all design, DataQuant stands out in the business intelligence market because it allows you to control and customize queries, reports, and dashboards to precisely fit the operational needs of specific individuals or groups across your enterprise.

DataQuant consists of the following components:

- DataQuant for Workstation\(^2\), an Eclipse-based\(^3\), cross-platform, rich workstation application that runs on Linux, Solaris, and Windows
- DataQuant for WebSphere, an application-server-based solution that extends key DataQuant function to enterprise users by using a “thin client,” browser-based deployment model

\(^2\) Throughout this information, the IBM DataQuant client for Workstation environments is referred to as DataQuant for Workstation, and the IBM DataQuant client for WebSphere Application Server is referred to as DataQuant for WebSphere.

\(^3\) Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software. Eclipse provides extensible tools and frameworks that span the software development lifecycle, including support for modeling, language development environments for Java, C/C++, PHP and others, testing and performance, business intelligence, rich client applications and embedded development. A large, vibrant ecosystem of major technology vendors, innovative start-ups, universities and research institutions and individuals extend, complement and support the Eclipse Platform. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the Eclipse projects. Full details of Eclipse and the Eclipse Foundation are available at [www.eclipse.org](http://www.eclipse.org).
The basic architecture for these components is shown here:

![DataQuant architecture diagram]

For additional information about product configuration and installation prerequisites, see Appendix B, “Prerequisites and installation information,” on page 69 and Installing and Managing DataQuant.

The business value of DataQuant

DataQuant provides a comprehensive array of business intelligence capabilities that are both powerful and easy to use.

As you evaluate business intelligence solutions, consider the following areas:

Coverage of function across the most commonly performed business intelligence tasks

DataQuant provides robust function across the entire breadth of key business intelligence tasks, including query, reporting, analysis, visual design, and application development.

Having all of these functions in a single product makes the solution easier to install, configure, maintain, update, tune, and support. You never have to worry about how to integrate key functions that are spread across multiple components. DataQuant allows you to rapidly develop and deploy executive dashboards, information portals, and interactive data visualization solutions to both workstation and Web users across your enterprise. It allows ad-hoc development of OLAP analytics, SQL queries, tabular reports, visual reports, pivot tables, and data analysis views. It includes a built-in table editor for experienced users and database administrators, and supports procedure objects, which help you automate
repetitive tasks such as importing data, and generating and distributing reports to a series of e-mail users. Support for most popular database management systems is built-in.

For more information about specific DataQuant functions, see Chapter 3, “Robust business intelligence function,” on page 15.

Ease of use

Built-in wizards and other intuitive user assistance make DataQuant easy to learn and use. Using DataQuant to develop content reduces your administration costs and unburdens your IT and support staff so that they can design thoughtful solutions that are tailored to your operational needs rather than spend time teaching less-technical staff.

For more information, see Chapter 4, “Ease of use,” on page 43.

Ease of administration

DataQuant simplifies the administration process by providing you with the ability to manage the product by using either a Web connection or a live workstation. This ability is particularly useful because you can make changes, install updates, troubleshoot, and provide user support from anywhere in the world.

Administration interfaces are intuitive and designed around the tasks that administrators normally perform, which allows you to easily search for and manage object metadata and create and share objects with others for maximum efficiency.

For more information, see Chapter 5, “Administration,” on page 49.

Security features

The rich security infrastructure of DataQuant is broad enough to handle the needs of both large and small organizations, yet flexible enough to accommodate your exact security requirements. You can assign security permissions on a group basis, which allows you to set up a role-based security scheme, or you can tailor permissions to individual users. The flexible security infrastructure of DataQuant allows you to customize not only the query and reporting environment, but the content within that environment as well. Security permissions are so granular that you can control access to individual database columns.

For more information, see Chapter 6, “Security,” on page 51.

Resource governing and usage auditing

DataQuant provides comprehensive built-in resource limits and governing capabilities, so that you no longer have to worry about runaway queries that consume a disproportionate amount of database and system resources. You can set resource consumption limits by user or group, which allows role-based versus individual restrictions to be created. Activity logs provide built-in usage auditing capabilities, including several ready-to-use reports, so that you have a snapshot of what each user is doing on the system at any given time.

For more information, see Chapter 7, “Performance and tuning,” on page 55.

Scalability

A scalable solution paves the way for your organization to grow and change without limiting your options. The throughput capabilities of
DataQuant can handle both large and small data demands, and its ease of use and intuitive design ensure that users of all skill levels can be productive immediately.

For more information, see Chapter 8, “Scalability,” on page 57.

Integration with your day-to-day operations and current product set

Although corporate mergers and acquisitions can strengthen an organization's market position, they often leave the IT staff wondering how to pick up the pieces and make everything work in harmony. DataQuant offers access to any data server that has a JDBC driver interface, which allows you to access virtually any data anywhere in your organization regardless of where it is stored.

DataQuant is part of the DB2 Warehouse solution and offers export capability and API support that allows it to blend seamlessly with both IBM and third-party products.

For more information, see Chapter 9, “Integration with existing tool sets,” on page 59.

Flexibility, customization, and personalization of interfaces and objects

Today's workplaces have become more global than even a few years ago, so they need to accommodate staff who work at various times throughout a 24-hour day. DataQuant allows flexible scheduling of queries and reports as well as the ability to work offline, which allows staff to make the most of every moment, wherever they are.

The visual design environment of DataQuant is so powerful precisely because it is so flexible. If you can imagine it, you can design it with DataQuant. Workspaces can be personalized to fit user or group needs and tasks. Access to objects can be tightly controlled so that users see only those objects for which they have the appropriate security permissions.

For more information, see Chapter 10, “Flexibility and customization,” on page 61.

DataQuant in the real world

DataQuant makes collecting, sorting, and expressing business data much more efficient and effective.

Many organizations today collect a significant amount of enterprise data that is related to their operations, including sales and profitability data by territory, and new business in the pipeline that has not yet been closed. Although these organizations might have good systems in place to record operational data, they remain challenged by the task of providing tailored access to this data across all levels of the organization. They might find that collecting the data is easy, but delivering that data to the people who need it in the format that they need to do their jobs remains the most challenging task faced by the IT group. Tools that go beyond traditional query and reporting are key, because they not only meet the diverse needs of the user base, from power users to executives, they also provide a means of quickly delivering customized reports to meet the day-to-day demands for ad-hoc business data.

DataQuant addresses these needs in the following ways:

- Executives have access to key corporate information through DataQuant dashboards, which summarize operational data across the entire enterprise.
Given a comprehensive overview of their corporate information, executives have convenient access to daily performance, and can easily drill down into specific areas of the business that are flagged as problematic.

- Dozens of customized business reports, which can be developed in minutes rather than days, provide instant access to job-specific operational data across all layers of the company. For example, inventory managers can review distribution and supplier reports, and sales managers can review the daily performance of their respective sales teams. Sales executives can run the same reports, yet see performance across all sales teams, leveraging the ability of DataQuant to tailor report content based on the type of user.

- Business analysts can use the visual design environment of DataQuant for Workstation to craft ad-hoc visual reports that perform analysis on the real-time business data that flows through the company’s transactional databases. With the ability to simultaneously correlate this data with their data warehouses, analysts can spot trends well before the monthly data is closed out and committed to the data warehouse.

### DataQuant for WebSphere versus DataQuant for Workstation

DataQuant includes a rich client (DataQuant for Workstation) and a thin client (DataQuant for WebSphere) component.

The component you use depends on business and technical requirements specific to your situation. While all administrative functions (and a lot of the user functions) available in DataQuant for Workstation are also available in DataQuant for WebSphere, there are some noteworthy differences.

The following list highlights functionality available in DataQuant for Workstation only.

**Note:** Most of these functions provide users the capability to design more robust reporting and dashboard solutions and enhance product usability.

- **Enhanced visual design capabilities:**
  DataQuant for Workstation includes a visual designer for both reports and dashboards. With little or no coding experience, users can employ the visual designer to enhance the appearance and usability of reports and dashboards. Users can graphically place charts, selectors, and controls on a report canvas or dashboard design editor. Users can embed static content such as text, graphics, hyperlinks, and supporting information (data-driven or static). These elements can also be embedded in traditional tabular reports or interactive dashboards.

- **Accessibility conformance:**
  DataQuant for Workstation conforms to accessibility standards and guidelines, including robust support for keyboard shortcuts.

- **Bidirectional data transformation:**
  DataQuant for Workstation supports bidirectional (BIDI) data transformation. BIDI refers to supporting data processing of text in two directions, right-to-left (RTL) and left-to-right (LTR). By including BIDI support, DataQuant for Workstation is able to correctly display text.

- **Use of the host operating system scheduler:**
  DataQuant for Workstation includes support for using the host operating system’s scheduler, in addition to the scheduler provided with the product. This allows for additional flexibility when scheduling tasks such as generating and distributing managed reports. With DataQuant for WebSphere, users are not able...
to use the host operating system scheduler. DataQuant for WebSphere users must use the scheduler that is included with the product.

- Enhanced drill-down editing capabilities:
  DataQuant for Workstation provides enhanced capabilities, such as zoom editing and column dragging. The zoom editing presents the user with the capability to zoom in for a more detailed view of data on a scene or on a data point in a layout.

- Customizable perspectives and views:
  DataQuant for Workstation provides the capability to customize workbench views. Users can move views around within the workbench, docking them in different locations to suit their preferences.
  Additionally, DataQuant for Workstation users can customize the workbench by adding and removing views from the various perspectives. This allows users to establish standardized views and perspectives that suit their design needs.

- Document editing capabilities:
  DataQuant for Workstation provides the capability to edit data in documents. For example, users can set font size and style as well as add color to text in documents.

- Support for REXX programming language:
  DataQuant for Workstation supports the REXX programming language, so advanced "power users" can develop and implement scripts that address complex and repetitive tasks and functions.

- Targeted printing capabilities:
  DataQuant for Workstation supports targeted printing. Using DataQuant for Workstation, users can print a single document of interest. For example, users can perform a **File > Print** operation on a specific report or query result and get the output for that single piece of data.

- Seamless procedure-run EXPORT command:
  DataQuant for Workstation provides seamless support for procedure-run EXPORT commands. Users can run procedures to export data in various formats directly to their machines. DataQuant for WebSphere users can also use procedures to run EXPORT commands, but the process requires the user to download and save data through a download wizard.
Chapter 3. Robust business intelligence function

IBM DataQuant offers a single streamlined package that includes multiple query methods, flexible yet comprehensive visual design and presentation options, dashboard functions, and application development interfaces across a variety of platforms.

When the key business intelligence functions that are used by the majority of your users are not all contained in a single product, the time and money that you spend implementing the entire solution can increase exponentially. Each product has a lifecycle of its own, which results in additional hours and expense to order, install, configure, deploy, maintain, tune, and support. Time to value improves with a simpler implementation model that reduces your reliance on special consulting services or support to keep the solution running smoothly.

Multiplatform data access

You can configure DataQuant access to the following types of relational and multidimensional data sources. Data access can be shared, personal, network-based, or Web-service-based.

- DB2 running on any platform (including z/OS)
- Informix®
- OLAP services included in DB2 Warehouse Version 9 or later (all editions)
- Other MDX-based OLAP servers with support for XMLA connectivity
- Virtually all other JDBC-compliant data sources, including Oracle and SQL Server

Support for JDBC-compliant data sources includes support for all JDBC-level data types.

Objects that are created with DataQuant are stored in an object repository, which is a centralized storage area in which users save their queries, forms, reports, and other metadata objects.

Additionally, DataQuant for WebSphere supports any Web application server (on any platform) that is capable of hosting Java-based Web applications deployed through the use of EAR or WAR files. DataQuant for WebSphere can be accessed by virtually any JavaScript-capable Web browser running on any platform, and has been formally validated against the following browsers:

- Internet Explorer Version 7 (or later)
- Mozilla Firefox Version 3 (or later)

Queries

DataQuant supports both relational and OLAP queries.

After the query is created, you can save it to:

- A file
- A workspace, which is a personalized collection of metadata objects and data sources that are frequently accessed
• A QMF catalog (see http://www.ibm.com/software/data/qmf/ for more information about QMF)

You can assign a descriptive name to a query and provide descriptive comments so that other users can easily recognize the purpose of the query. Queries can be shared and reused in this way, and administrators can create queries that can be run by users without any additional modification, intervention, or training. All data elements that a query might need to access can also be assigned intuitive names and additional comments so that everyone can easily recognize which data objects within a particular query are relevant for their operational needs and modify the query if necessary.

Queries can be embedded within grids, tables, or charts with surrounding checkboxes, lists, combos, and other elements that provide an intuitive means of selecting available parameters.

After you create and run the query, you can:
• Format, group, and aggregate the results
• E-mail the results directly from the File menu
• Export the results to a database or a file in one of the following formats:
  – HTML
  – PDF
  – CSV
  – Ixf
  – dBase III
  – XML
  – WQML
  – TXT
• Use the results to create:
  – A tabular report, as explained in “Standard tabular reports” on page 27
  – A visual report (for example, a chart, graph, or map), as explained in “Visual reports” on page 23
  – A dashboard, as explained in “Dashboards” on page 28

DataQuant also provides full large object (LOB) support, which allows you to retrieve, view, edit, and export LOB data. Because retrieval and manipulation of this type of data can be very resource-intensive, DataQuant allows you to restrict how users interact with LOB data.

**Relational queries**

DataQuant offers many ways to build relational queries, providing all users with a productive means of accessing data, regardless of their varying levels of skill.

To create a query, you can:
• Use the diagram interface
  This method allows you to create simple or analytical queries by using a drag-and-drop mechanism. Even complex joins are made easy by “wiring” the tables together visually. The following figure shows this interface.
Use the prompted query interface
This facility leads you through the process of creating a relational query by using a graphical interface rather than manually typing SQL text.

Write your own SQL statements
The SQL editor is colorized so that you can clearly see certain types of statements and relationships. The content-assist function displays a list of SQL keywords or object names that are appropriate for your position in the SQL statement. You can also display parameter hints for the SQL functions that you use in the query if you do not know the full syntax or use of the function.

Use the Draw Query wizard to build a query visually
This method is available in DataQuant for Workstation and allows you to create an SQL query by specifying the tables to query and the query type, which can be a SELECT, UPDATE, or INSERT query. The SQL statement is then built automatically and references the names and data types of all the columns in the selected table or tables.

With all of these methods, DataQuant generates and displays the SQL statements that will be issued to the database. In this way, users can gradually learn SQL if they want to. Users can switch between the SQL, prompted, and diagram views of a query to obtain more information or to continue building the query by using a different method.

DataQuant supports the full range of SQL functions, including stored procedures as well as analytical queries that are nested within one another (also called subqueries), where multiple passes of the data are required. Subqueries can be implemented in one of the following ways:

- A given query can embed the output of an independent query (for example, to drive the parameter values of the original query).
For example, suppose that query A returns a list of customers. This query can then be used to supply a parameter set to query B, which might perform a task such as generating an invoice list on a per-customer basis. Thus, query A retrieves a subset of information and query B returns data based on the outcome of the subset.

- Visual elements in a report or dashboard can be authored to nest layouts within other layouts.

The queries that drive a given layout automatically act as subordinate queries that deliver information based on the output of the queries that embed them. For example, a given query might return product market regions in the form of geospatial data. A geospatial region can be configured to embed a bar chart that displays monthly sales for that region. Furthermore, each bar on the chart can be configured to embed a product-mix pie chart for each month. By defining this presentation model once, data presentation is expanded into multiple entities when the queries are executed. In this example, the query would produce a particular number of geographical regions, each with its own bar chart, and each with distinct product mixes on each bar.

DataQuant provides you with full control over how you want to handle missing or null data points in the query results. With DataQuant, you can:

- Remove missing or null values at the SQL level by using database-specific functions, such as Value(Column, 0). DataQuant has a built-in SQL parser and knowledge of all DB2 functions.
- Handle missing or null values by using DataQuant's own expression language. All of DataQuant's expressions are available in the built-in expression designer, which offers the functions by category and automatically enters the function text for you. For example, the function expression Coalesce(Column, 'N/A') turns all null values into the text “N/A”. Values can also be handled with a conditional expression that outputs different results depending on the column value. For example, the following expression uses either the column value or “Empty” if the column is null:
  
  \[
  \text{if (Column=null, 'Empty', Column)}
  \]

OLAP queries

DataQuant supports multidimensional analysis through the use of OLAP queries.

OLAP queries are multidimensional queries that can present your cube data in a wide variety of views. You can use the OLAP query wizard to easily create a new OLAP query, or open an existing query and modify it to obtain different views of the cube data. Drill-down, drill-up, and drill-across operations are also supported, as explained in "Data drilling capabilities" on page 19.

As with relational queries, DataQuant offers multiple methods of constructing multi-dimensional queries. You can browse available cube models and construct queries by graphically dragging and dropping dimensions and measures into the query structure. Experienced users can open the Multidimensional Expressions (MDX) editor and directly enter or edit the query's underlying MDX statements. You can also switch between the MDX and graphical editors, optionally building the query by using a blend of both methods.

You can format OLAP query results directly in the query results editor window, which provides many options, including the ability to:

- View summary information for a measure or dimension
- View only totals for a measure or summarized dimension
• View detailed data for a summarized dimension
• Resize columns
• Change the font or format for a measure or dimension

**Data drilling capabilities**
By using DataQuant, you can zoom in or out to find just the data you need at the level of detail that is appropriate for the situation, without ever leaving the current page.

You can drill down through an unlimited number of levels of detail and graphical entities, exposing underlying information with each click. Drill-downs can be performed by using the traditional “navigate to new page” method or by using DataQuant’s information zooming facility, which reveals more information as you zoom closer to the graphical layout without moving from the page itself (much like Internet map sites reveal more information as the user zooms in). For example, suppose that a page displays a scatter graph of course grades versus course duration. When zoomed in a level, the scatter graph data points could change to text labels that present the course name, lecturer, and number of attendees on the scatter graph itself.

Clicked items carry the full context to the destination object or page. For example, by clicking on a pie slice that is embedded in a bar chart that is embedded on a regional map, data that characterizes region, sales month, and product ID can be automatically transferred to the destination object or page. After the data is transferred to the object or page, that data can be used to drive the drill-down charts and graphs that provide more information about the item. All graphical entities can be assigned drill-down actions so that you can customize what a particular part of a visual report or dashboard will do when it is clicked or dragged.

You can also toggle between flat and graphical views of the data. These views support both detailed and high-level analyses. Additionally, linear regression capabilities support planning, budgeting, analyzing, and forecasting activities.

**Multidimensional analysis**
OLAP cube data can be displayed in both the workstation and Web interfaces of the product and can be directly incorporated into visual reports and dashboards by using DataQuant’s OLAP-aware charts, graphs, and dimension slicers.

You can drill up and drill down through a user-defined set of dimensions and levels by using the built-in OLAP explorer. You can then create new or open pre-existing OLAP queries of interest in DataQuant's graphical OLAP designer. The OLAP designer allows you to drag and drop dimensions and measures of interest into your queries and immediately see the results of your selections. No coding knowledge is required to browse and construct OLAP queries. DataQuant also provides access to the same data manipulation options provided by relational queries, including:
• Conditional formatting (changing the appearance of cells based on numeric conditions)
• The ability to bookmark and restore particular formatting and layout options and preferences for recall in subsequent sessions

You can apply OLAP result sets to standard layouts in the following steps:
1. Select a measure to be charted over a given dimension at a given dimensional level. For example, you can chart sales over time at the quarter level. Unassigned dimensions in the query return all values by default.

2. Use the OLAP slicer to obtain a hierarchical list of a designated dimension. The slicer provides a drop-down list that allows you to select all values or just certain values in a dimension.

3. Connect one or more OLAP slices to dimensions associated with a layout's OLAP query. Content authors can associate an OLAP slicer object with a layout that is driven by an OLAP query by “wiring” the two together. When the two are connected, the wizard presents the list of dimensions for the layout's query and allows you to connect the slicer object to one of them.

By using DataQuant, you can also easily constrain a dimension by using preset parameters rather than having to manually select each desired attribute in the set. You can preconfigure OLAP queries to use specific dimensions, and optionally filter levels within those dimensions (for example, you could include only north and east regions in a report rather than north, east, south and west).

Prompt hierarchies
Prompt hierarchies allow you to select one or more variable values from a flat or hierarchical list of options.

One way to add user interaction to a query is to add a variable that prompts you to specify a specific value. For example, when you run a query against staff data, you are prompted to specify what department data is to be returned in the query result set. However, what if you wanted to be able to display specific departments but exclude others? Taking it further, what if you wanted to then be able to specify which job types in those departments to display? You could create a prompt hierarchy that breaks your single query down by department first, and job type second. When you run the query, you are presented with a tree of options that lets you specify only the specific rows of data that you want to display.

Prompt hierarchies are distinct DataQuant for Workstation objects that are stored independent of a given query, allowing them to be used by multiple query objects. Prompt hierarchies draw their option list from a query, for example “select region from sales territories”. In addition, a prompt hierarchy can present a hierarchy of values, with unlimited levels of detail. For example, “select region, country from sales territories”. In this case, the list of options would be broken down by ‘Americas’, ‘Europe’ and ‘APAC’, with USA, Canada and Mexico listed under ‘Americas’ and so on. You can then select values at both levels of the prompt, perhaps selecting ‘APAC’ in its entirety and ‘Americas’ but not selecting Mexico.

Analytical queries
Analytical queries allow you to combine data from multiple queries from the same or differing data sources into one result set.

There may come a time when you must be able to draw data from several different sets of data, some of which might be stored in different data sources. Querying each set of data one at a time can be time consuming and ultimately adds unnecessary time and confusion to your query management process. Using analytical queries, you can combine query results from multiple queries that span different data sources into one query result set. You can add any number of queries in any number of combinations to your analytical query, allowing you to...
author one comprehensive result set that retrieves exactly the data that you need. There are a variety of query types that you can incorporate into an analytical query structure tree:

- **Append Query** - Append queries combine query results from two queries into one query result set. You can specify how the columns will be paired up across the two result sets.

- **Join Query** - Join queries join the result sets from two queries into one unified result set by joining one or more columns together, much like a join operation in SQL.

- **Crosstab Query** - Crosstab queries provide a cross-tabulation of one or more columns, such as a sum, average, min, max, computed over two or more tabulation columns. For example, tabulating the sum of product sales across the region and month.

- **Normalize Query** - Normalized queries "normalize" the text returned in a query result set. Normalizing query text facilitates query processing, such that executions of the same query with different parameters can be easily compared with one another and aggregated together.

- **Conditional Grouping Query** - With Conditional Grouping queries you can create aggregated summaries of result set data based on conditional expressions.

- **Column Filter Query** - With Column filter queries you can add a calculated column before or after existing columns and enter filtering functions.

Multiple queries and tables can be added into the analytical query structure tree to generate a comprehensive result set that encompass data from many different sources. The different options of each query type allow you to customize how this data is displayed.

**Forecasts**

Forecasts are projections of future data based on historical time series data.

With forecasting capabilities, you can make projections of future values based on past values. Using forecasts, organizations can prepare for changes in economic or competitive conditions by analyzing time series historical data to predict performance and future trends. For example, in a supply chain, if the forecast demand matches the actual demand then significant efficiencies can be achieved in terms of production, distribution, and return.

DataQuant forecasts use various predictive methods based on mathematical algorithms that model the future demand based on time series historical data that can be sourced from queries and tables containing date or time columns. The overall objective is to choose a time series method that produces a best fit model of past values, by identifying existing patterns in the data and projecting the model into the future to generate the forecast.

The following methods can be used to forecast future values:

- If the time series is relatively stationary with no overall tendency to fluctuate at one part of the series as compared to another part of the series, then Moving Average, Weighted Moving Average, or Single Exponential Smoothing provide the best fit model.

- If the time series has a trend with a consistent upward or downward movement over time, then Double Exponential Smoothing provide the best fit model.
• If the series has a trend and seasonality with a pattern of peaks and troughs that repeat themselves over a time-frame of usually less than or equal to a year, then **Holt-Winters method** provide the best fit model.

• If the series has a trend, seasonality and cyclicity with a pattern of peaks and troughs that repeat themselves over an extended time-frame usually greater than a year, then the **Multiplicative Decomposition method** provide the best fit model.

• If the series displays none of the above, then **Neural Networks** be used to mathematically fit the historical data.

• If there are theoretical reasons to indicate that the data should follow a clear mathematical function, then one of the **Curve fitting methods** can be used.

• In addition to the above methods, the forecaster is also able to manually adjust any predicted values based on the forecaster’s knowledge and any external events.

As most new users discover, the ability to quickly plot and compare each forecast method is a major feature of DataQuant forecasts. However, a forecaster’s knowledge and experience help to reduce the possibilities and consequently provide greater confidence and reliability in the forecast.

---

**Figure 3. A forecast that predicts future revenue.** This figure shows forecasted revenue totals for three months beyond the known historical data.

**Reports**

From standard tabular reports to visual reports, which allow you to chart, graph, and map your data, DataQuant challenges and extends the traditional notion of reporting, making your communication and presentation more precise, effective, and powerful.
In addition to traditional page-based reports, DataQuant includes built-in dashboard capabilities so that you can see live operational data presented in a variety of creative combinations. (For more information, see “Dashboards” on page 28.)

Visual reports
DataQuant has an extremely flexible visual design environment for both reports and dashboards.

With no coding knowledge, you can give life to your ideas with nearly any visual format that you can imagine. Based on the award-winning Eclipse graphical editing framework, DataQuant’s visual designer allows you to graphically place charts, selectors, and controls on the report canvas and embed static content such as text, graphics, hyperlinks, and supporting information (data-driven or static). These elements can also be embedded in traditional tabular reports or interactive dashboards.

DataQuant’s dynamic object properties provide full control of the data elements, down to the individual items in charts, which allows you to control their appearance and behavior as a function of the data and user interactions. The intuitive user interface allows designers to simply drag and drop visual elements into charts, graphs, and maps, then size and position them and manipulate their format, color and options through the use of a properties sidebar.

Visual formatting
You can control many visual attributes in a report.

Some of these attributes apply only to tabular reports, whereas others also apply to visual reports and dashboards. You can make the following types of changes:

• Change column names and add headers and footers, page breaks, and numbering
• Select font, script, style (boldfacing, underlining, and other styles), size, and color separately for heading and data elements
• Control background and foreground color
• Change alignment properties as a whole or for individual columns
• Specify text wrapping and numeric separators

You can specify unique format attributes for each column's header cell, for each detail cell in every column, for each break-point summary cell, or for the column's final summary cell. Format options vary depending on whether you selected a column that contains character, numeric, date, or time data.

DataQuant provides preview capabilities so that you do not waste time going back and forth between your results and the options.

Conditional formatting and expressions
Conditional formatting lets you customize the way DataQuant processes and displays your data.

No one knows your business needs better than you. That's why DataQuant for Workstation allows you to include conditional formatting in queries, reports, and dashboards. Conditional formatting allows you to specify conditional expressions that control both the display of the data as well as the behavior of a query, report, or dashboard element (what it does when clicked or changed, based on the
underlying data). For example, you can use conditional formatting to highlight in red any fields that indicate year-end sales that have fallen below a particular figure.

DataQuant’s expression language offers broad support for:

- Built-in mathematical and statistical functions, such as Sum(), Average(), and StdDev()
- Numeric operators that can be part of arithmetic operations or comparisons (for example, +, -, >, <, <=, >=)
- Logical and relational operators
- References to other columns that are returned by the query (for example, ColumnA + 3 * ColumnB)

**Calculation functions**
Calculation expressions can be used to define columns and add depth to your reports.

With DataQuant, you are not limited only to the discrete set of columns as they are defined in the tables that you are querying. You can also add calculated columns to your reports, in which you create a column of data that is based on an expression that you define and then add that column to the query results. Calculation expressions can use data from other columns, as well as constants and variables. DataQuant is also capable of displaying percentage or unit changes between values in the same row of a report, so that the display can reflect the granularity in the expressions that you define.

**Grouping, summary, and aggregation functions**
DataQuant allows you to group, aggregate, and summarize data in queries and reports.

For example, a sales report can summarize performance by department or it can average commissions across the company as a whole or by salesperson, sales account, department, manager, or almost any other column in the report. The following grouping functions are available:

**Top group**
Placing a column in a top group displays related values of that column on the horizontal axis. This function is also known as pivot or across capability. You can define multiple columns as top groups.

**Side group**
Placing a column in a side group aggregates data down the left side of the query results or report by the selected column.

**Count**
The Count function creates a summary row at the bottom of the query results that shows the number of values in the column.

**First**
The First function creates a summary row at the bottom of the query results that shows the first value in the column.

**Last**
The Last function creates a summary row at the bottom of the query results that shows the last value in the column.

**Maximum**
The Maximum function creates a summary row at the bottom of the query results that shows the maximum value in the column.
Minimum
The Minimum function creates a summary row at the bottom of the query results that shows the minimum value in the column.

Sum
The Sum function creates a summary row in the query results that shows the sum of the values in the column.

Cumulative sum
The Cumulative sum function displays the values in the column as a cumulative sum, which is a rolling sum.

Average
The Average function creates a summary row that shows the average of the values in the column.

Standard deviation
The Standard deviation function creates a summary row that shows the standard deviation of the values in the column.

Percentage of group
The Percentage of group function identifies how each value in the group contributes to the whole; it displays the values in the column as a percentage of the group’s total, with a summary row that shows the sum of the percentages.

Percentage of total
The Percentage of total function displays the values in the column as a percentage of the column total, with a summary row that shows the sum of the percentages.

Cumulative percentage of group
The Cumulative percentage of group function displays the values in the column as a percentage of the group total. A summary row at the bottom of the query results shows the sum of the percentages.

Cumulative percentage of total
The Cumulative percentage of total function displays the values in the column as a percentage value of the group total and a percentage value of the column total. A summary row at the bottom of the query results shows the sum of the percentages.

Sorting options
Different sorting options let you customize the way report columns are displayed. DataQuant allows you to sort columns in ascending or descending order, or set up a multi-level sort, in which you can layer column data so that it sorts in ascending or descending order by the first column, then by the second column within the first, then by the third column within the second.

Creating charts, graphs, and maps
DataQuant provides a graphical palette that contains over 20 default charts and graphs.

The graphical palette includes:
- Line charts
- Bar charts
- Pie charts
- Column charts
- Scatter charts
Customizing graphics

DataQuant has a rich graphical palette, which contains dozens of graphical primitives that allow content authors to build their own custom charts if one of the default chart types is not suitable.

Primitives, or primitive objects, are graphic objects such as text, lines, shapes, alignment panels, and pictures. For example, a floating quartile graph can be assembled by using a chart frame, axes, a data rectangle and three straight lines that are tied to quartile functions that operate against the data that was supplied by the query.

The following primitive objects are provided:

- **Label**
  Displays text, with or without a rectangular frame

- **Line**
  Displays a straight line segment that connects two points

- **Arrow**
  Displays a straight line segment with an arrowhead on one endpoint

- **Double arrow**
  Displays a straight line segment with an arrowhead on each endpoint

- **Polyline**
  Displays a series of line segments that connect vertices

- **MultiPolyline**
  Displays one or more unique polyline objects that are connected or disjointed

- **Polygon**
  Displays a series of line segments that connect vertices

- **MultiPolygon**
  Displays a series of one or more disjointed polygon objects

- **Rectangle**
  Displays a rectangle

- **Round rectangle**
  Displays a rectangle that has rounded corners
**Ellipse**
Displays a standard ellipse with foci aligned with the x- or y-axis.

**Picture**
Displays one of the following image types:
- Joint Photographic Experts Group (.jpeg)
- Graphics Interchange format (.gif)
- Portable Network Graphics (.png)

**Alignment panel**
Groups multiple objects relative to a single point so that they all move in unison when based on a single location value.

**Standard tabular reports**
DataQuant allows you to apply ad-hoc or pre-configured tabular report formats to query results.

DataQuant’s tabular reporting facility puts the power of detail at your fingertips by allowing you to control the formatting at a very granular level as well as define your own layout options and conditional logic. In this way, you can control virtually every aspect of a report’s format.

Information can be:
- Data-driven, in which information that is derived from the underlying data is presented
  For example, you can develop an expense report that displays “Quota Met” or “Below Target” for various columns, depending on the values of “budget” and “actual” columns in the underlying query.
- Static, in which data values from the underlying query are represented directly on the report
- A mixture of the two

After a report is generated, you can either apply updates to the original object (if permission is granted) or save the changed report under a new name. If you choose the latter, the object is still linked to the original query or queries and automatically reflects any changes that are made to the originals.

**Saving formats as templates**
Formatting and presentation of data is often less dependent on individual preferences and more dependent on the operational needs of a particular department, role, or function within your organization. For this reason, DataQuant allows you to save your formatting options independently of the data, so that you can reuse formatting attributes against another set of data. In this way, role-specific and department-specific report formats can be developed and readily applied whenever you run a query that requires that format.

**File formats and delivery**
You can convert a tabular report to a chart or graph or edit the underlying form to further refine how it is displayed.

After you have generated either a tabular or visual report, you can:
- Print it
- Convert it to HTML format
Dashboards

A dashboard is an interface that integrates data from a variety of sources and provides a unified display of relevant and in-context information.

By using DataQuant, content developers can create graphical dashboards and database visualization applications that present either interactive or persistent data. Dashboards are typically designed to run interactively (though they are not required to do so) and provide a dynamic means of reviewing critical enterprise data. The data that is presented in a dashboard can be obtained from querying multiple data sources across the enterprise and displaying it in a wide range of visual images, including graphs, maps, charts, and custom graphics.

Unlike reports, which tend to contain a fixed amount of information, dashboards have the ability to deliver real-time information on demand, as needed by the dashboard user. For example, an executive might need to see an operational summary across all business units. Real-time color coding of data (for example, red, yellow, and green) draws the executive’s attention to areas of concern. Clicking on problematic areas immediately produces dynamic reports that reveal the information underlying each area of concern.

You create dashboards by using DataQuant’s built-in visual designer, shown in the following figure:
To create a dashboard:
1. Drag a dashboard object from the graphical palette and arrange it on the dashboard canvas.
2. Set appearance and font by using the Properties view.
3. Add queries and graphical objects (such as date selectors and charts) to your dashboard and define relationships between these objects by graphically wiring the objects together and specifying their relationships.

As you create your dashboard, DataQuant provides built-in ways to:
- Display query results
  You use layout objects or the List and Combo controls to display query results data. You can explore the different display options (explained in Getting Started with DataQuant) and decide which options best suit the kind of analysis you want to show.
- Capture user input
  You can capture user preferences by using standard user interface controls.
- Pass information to and from the dashboard
  You can use parameters to pass information that has been acquired from user actions. This information then affects the display of a dashboard, the content of a scene, or the execution of a query.
- Implement navigation features
  You can use navigation features to allow users to move around the dashboard to get to new information.
- Display and present information visually
You can drag and drop visual elements on the dashboard canvas and define relationships between them.

- Point to data sources

As you create your dashboard, you specify the data sources that are referenced in the queries that your dashboard uses.

After you create a dashboard, it can be directly viewed in both the workstation and Web-based applications, as shown in the following figure:

**Figure 5. A finished dashboard, rich with graphical elements that are perfectly tailored to the dashboard user’s purposes**

### Analysis functions

DataQuant's analysis functions help make sense of your data.

Report formatting and presentation alone mean nothing unless you are able to extract meaningful insight from your data. DataQuant’s analysis capabilities go beyond traditional reporting to help you truly gain the insight that you need to drive your organization forward. DataQuant is fundamentally designed for drill-downs in queries, reports, and dashboards and excels in this area, so that you
can drill down directly through the use of the graphical display and drill up to
analyze trends and see summary-level information. A huge repertoire of built-in
functions allows you to tailor reports and dashboards at an extremely detailed
level. The following topics provide more information about DataQuant's analytical
functions:

- "Arithmetic functions"
- "Color functions"
- "Conversion functions" on page 32
- "Data formatting functions" on page 33
- "Date and time functions" on page 34
- "Hierarchical functions" on page 35
- "Information functions" on page 36
- "Logical functions" on page 36
- "Math and trigonometric functions" on page 37
- "Measured functions" on page 38
- "Security functions" on page 38
- "Spatial functions" on page 39
- "Statistical functions" on page 39
- "Text functions" on page 40
- "Translation functions" on page 41

**Arithmetic functions**

Arithmetic functions calculate values based on the specified arithmetic function.

- **add()** Adds two numeric arguments.
- **divide()** Divides two numeric arguments.
- **mod()** Returns the modulus, or integer residue of a number following an integral
division by a second number.
- **multiply()** Multiplies two numeric arguments.
- **percent()** Divides number by one hundred.
- **subtract()** Subtracts two numeric arguments.
- **sum()** Returns the sum of an expression, calculated based on the contents of the
query results set.
- **unaryMinus()** Negates the numeric argument.

**Color functions**

Color functions help you customize visual presentation of reports and dashboard
objects.

- **blend()** Blends two colors by using a specified ratio.
- **blueval()** Returns the amount of blue in a color (0-255).
**brighten()**
Brightens a color by the specified amount.

**colorMap()**
Returns a color based on a numeric value that has been defined in a given color map.

**colorSeq()**
Returns a color based on an index value that specifies a wedge in a color sequence wheel.

**darken()**
Darkens a color.

**greenVal()**
Returns the amount of green in a color (0–255).

**hsl()**
Returns a color with the specified hue, saturation, and intensity (value in the range of 0 - 255).

**hue()**
Returns the hue of a color (0–255).

**luminance()**
Returns the luminance of a color (0–255).

**redVal()**
Returns the amount of red in a color (0–255).

**rgb()**
Returns a color with the specified amounts of red, green, and blue (color value in the range of 0 - 255).

**saturation()**
Returns the saturation of a color (0-255).

---

**Conversion functions**
Conversion functions convert the parameter value from its current data type to another data type.

**asDate()**
Converts object to date.

**asDateTime()**
Converts object to date and time.

**asTime()**
Converts object to time.

**boolToNum()**
Converts Boolean true and false to 1 and 0 respectively.

**boolToStr()**
Converts Boolean true and false to character strings of True or False respectively.

**formatNum()**
Converts a number to text using custom formatting.

**getX()**
Returns X coordinate of a given point.

**getY()**
Returns Y coordinate of a given point.

**hex()**
Returns the hexadecimal value of a given number.
indexOf()  
Returns a list of indices of elements from the second list that are contained in the first list. Indices are relative to where the element is positioned in the first list.

numToBool()  
Converts a number to a Boolean (true/false) value.

numToColor()  
Converts a number to a color.

numToStr()  
Converts a number to a string with default number formatting.

poly()  
Constructs a pointset.

polySet()  
Constructs a polygon set.

polySetToStr()  
Converts a polygon set to text.

polyToStr()  
Converts a pointset to text.

pt()  
Constructs a point from individual x-y values.

ptToStr()  
Converts a point value to text.

rwinding()  
Reverses the order of the points.

spoly()  
Converts a text value into a pointset.

spolySet()  
Converts a text value into a polygon set.

spt()  
Converts a text value into a point.

str()  
Converts an object to a text string.

strToBin()  
Converts a text string to a binary notation.

strToHex()  
Converts a text string to hexadecimal notation.

strToNum()  
Converts a string value into a numeric value.

strToTextSet()  
Converts an arbitrary number of explicit strings into a textset.

Data formatting functions
Data formatting functions reformat data values to and from database and operating system formats.

dbDateToUSLocale()  
Converts a date from database format to USA format.

dbDateToWin()  
Converts a date from database format to Windows format.
dbMoneyToUSLocale()
    Converts a money amount from database format to USA format.

dbMoneyToWin()
    Converts a money amount from database format to Windows format.

dbNumToUSLocale()
    Converts a number from database format to USA format.

dbNumToWin()
    Converts a number from database format to Windows format.

dateToDB()
    Converts a date and time from USA format to YYYY-MM-DD
    HH:MM:SS.SSSSS.

dateToDBInformix()
    Converts a date from USA format to Informix database format.

dateToDBInformixInt()
    Converts a date from USA format to Informix database format using
    specified interval settings.

dateToDBOracle()
    Converts a date from the operating system’s format to an Oracle database
    format.

moneyToDB()
    Converts a money amount from USA format to JDBC style.

winDateToDB()
    Converts the date from Windows format to database format.

winDateToDBInformix()
    Converts the Informix date from Windows format to database format.

winDateToDBInformixInt()
    Converts the Informix date from Windows format to database format and
    lets you specify interval settings.

winDateToDBOracle()
    Converts the Oracle date from Windows format to database format.

winMoneyToDB()
    Converts money from Windows format to database format.

winNumToDB()
    Converts the number from Windows format to database format.

Date and time functions

Date and Time functions return specific elements of date and time values.
date()   Returns the number that represents the specified date.
dateTime()   Returns the number that represents the specified date and time.
day()     Returns the day for the given date.
dayName()   Returns the name of the day for the given date.
dayOfWeek()   Returns the day of the week for the date.
dayOfYear()
    Returns the number of days since the start of the year.

hour()   Returns the hour for the given date and time.
jday()   Returns the Julian day for the given date.

microsecond()
    Returns the fractional second for the given date and time.

minute()  Returns the minute for the given date and time.

month()  Returns the month for the given date.

monthName()  Returns the name of the month for the given date.

now()  Returns the current date and time as a number.

quarter()  Returns the quarter for the given date.

second()  Returns the second for the given date and time.

time()  Returns the number that represents the specified time.
today()  Returns the current date as a number.

year()  Returns the year for the given date.

Hierarchical functions
Hierarchical functions will identify a parameter value's position in hierarchical
representations such as organization charts.

childCount()
    Returns the number of child data points below the current position in a
    hierarchical layout.

childDepth()
    Returns the depth of the current data point relative to the top-most data
    point in a hierarchical layout.

hasChildren()
    Returns whether or not the current item has child items in the hierarchical
    layout.

hasParent()
    Returns whether or not the current item has a parent item in the
    hierarchical layout.

hasSiblings()
    Returns whether or not the current item has sibling items in the
    hierarchical layout.

siblingCount()
    Returns the number of sibling items that contain the current item in the
    hierarchical layout.

siblingNumber()
    Returns the 0-based index of the current item relative to its sibling items at
    the same level in the hierarchical layout.
**Information functions**

Information functions return information on specified data objects.

- **accum()**
  
  This function computes the sum of values for a given numeric value, across a specific, contiguous group of rows in a query result set. The function can only be called within a data template.

- **coalesce()**
  
  Evaluates the specified arguments in order and returns the first one that is not NULL.

- **field()**
  
  Executes the specified query and returns from the specified query result set a textset containing all values for the specified column.

- **fieldValue()**
  
  Executes the specified query and returns the value for the specified field from the specified row.

- **groupRowNumber()**
  
  Returns the number of the row that contains the specified group.

- **isNull()**
  
  Tests whether or not the supplied parameter is null.

- **isSummaryRow()**
  
  Returns true if current query result set row is summary row.

- **pointCount()**
  
  Returns the number of data points contained in the current set of query results.

- **pointNumber()**
  
  Returns the zero-based index of the current row in the current set of query results.

- **stockImage()**
  
  Returns the image data from the stock images collection corresponding to the supplied name.

**Logical functions**

Logical functions return values based on logical operations performed on parameter values.

- **and()**
  
  Returns true if both numeric arguments are true.

- **equal()**
  
  Returns true if two numeric arguments are equal.

- **greaterOrEqual()**
  
  Returns true if the first numeric argument is greater than or equal to the second numeric argument.

- **greaterThan()**
  
  Returns true if the first numeric argument is greater than the second numeric argument.

- **if()**
  
  Evaluates a condition and returns the corresponding true or false expression.

- **@if()**
  
  Tests for specific values within an expression and returns the results.
lessOrEqual()
   Returns true if the first numeric argument is less than or equal to the
   second numeric argument.

lessThan()
   Returns true if the first numeric argument is less than the second numeric
   argument.

not()
   Returns true if the numeric argument is not true.

notEqual()
   Returns true if two numeric arguments are not equal.

or()
   Returns true if either numeric arguments are true.

Math and trigonometric functions
Math and trigonometric functions calculate values based on the specified
mathematical function.

abs()
   Returns the absolute value of a given number.

acos()
   Returns the inverse cosine in degrees.

acosh()
   Returns the inverse hyperbolic cosine in degrees.

actn()
   Returns the inverse cotangent in degrees.

asin()
   Returns the inverse sine in degrees.

asinh()
   Returns the inverse hyperbolic sine in degrees.

atan()
   Returns the inverse tangent in degrees.

atan2()
   Returns the inverse tangent, derived from the supplied opposite (y) and
   adjacent (x) values.

atanh()
   Returns the inverse hyperbolic tangent in degrees.

ceil()
   Returns the closest integer greater than or equal to the supplied number.

cos()
   Returns the cosine of the supplied angle.

cosh()
   Returns the hyperbolic cosine in degrees.

cot()
   Returns the cotangent of the supplied angle.

degrees()
   Converts radians to degrees.

deg()
   The result is expressed in degrees.

exp()
   Returns e, raised to the power of a given number. The value e equals
   2.718281828459045, the base of natural logarithms. As such, this function
   can also be considered as providing inverse natural logarithms.

fac()
   Returns the factorial of a supplied number.

floor()
   Returns the closest integer that is less than or equal to the supplied
   number.

int()
   Rounds the supplied number to the nearest integer.

ln()
   Returns the natural logarithm of a given number.
log() Returns the logarithm of a number with a specified base.

log10() Returns the base-10 logarithm of a given number.

max() Returns the largest number in an explicit series.

min() Returns the smallest number in an explicit series.

pi() Returns the value of pi.

power() Raises a given number to a supplied power.

radians() Converts degrees to radians.

rand() Returns a random value, uniformly distributed between lower and upper limits.

randInt() Returns a random value between 0 and 32,767.

randn() Returns a random value, with a Gaussian Normal distribution.

round() Returns the closest integer.

sign() Returns the sign of the value, reflecting whether or not it is negative, zero or positive.

sin() Returns the sine of the supplied angle.

sinc() Computes the SINC value, normalized to 1.

sinh() Returns the hyperbolic sine in degrees.

sqrt() Returns the square root of the supplied number.

tan() Returns the tangent of the supplied angle.

tanh() Returns the hyperbolic tangent in degrees.

Measured functions
Measured functions expresses a given value as a specified unit of measure.

cm() Specifies a number in centimeters.

in() Specifies a number in inches.

mm() Specifies a number in millimeters.

pixels() Specifies a number with a description of pixels.

points() Specifies a number with a description of points.

removeUnits() Removes units from specified measured value.

Security functions
The isSecurityLevel() function specifies the name of the security list that will be used to tailor visual report or visual dashboard content based on the security level of the user.
**Spatial functions**

Spatial functions are used when mapping spatial data.

- **centerPoint()**
  Determines the center point of a polygon by a pointset.

- **moveCenter()**
  Centers the pointset.

- **rotate()**
  Rotates a pointset clockwise.

- **rotateInPlace()**
  Rotates a pointset by the specified angle.

- **scale()**
  Scales the pointset relative to a reference location.

- **scaleInPlace()**
  Scales the pointset relative to its center point.

- **translate()**
  Translates the pointset.

- **xspan()**
  Computes the width of a bounding rectangle encompassing the supplied pointset.

- **yspan()**
  Computes the height of a bounding rectangle encompassing the supplied pointset.

**Statistical functions**

Statistical functions are used to perform standard statistical analysis on supplied parameter values.

- **aveDev()**
  Returns the average deviation.

- **average()**
  Returns the average.

- **chi2()**
  Computes the chi-square probability function with specified degrees of freedom.

- **comb()**
  Returns the number of distinct items from a collection of items.

- **kurt()**
  Returns the kurtosis (4th moment) of a numeric query column.

- **largest()**
  Returns the largest value.

- **linReg()**
  Performs a linear regression.

- **perm()**
  Returns the number of permutations of a given size from a collection of a larger size.

- **skew()**
  Returns the skew of a series about the mean (3rd moment).

- **smallest()**
  Returns the smallest value in a numeric query column.
stdDev()
Returns the standard deviation.

variance()
Returns the variance.

Text functions
Text functions format textual data or return specific information on supplied data sources.

concat()
Concatenates two text strings into a single string.

decrypt()
Decrypts the supplied encrypted message.

encrypt()
Encrypts the supplied plaintext message.

filterTextSet()
Returns filtered textSet according filter pattern. Pattern can contain SQL LIKE special characters: '%', and '_'.

filterTextSetRegExp()
Returns filtered textSet according filter pattern. Pattern should be in JAVA REGEX syntax.

find()
Returns the index of the first occurrence of the supplied substring in a supplied string, starting the search from a given location in the string.

findNoCase()
Returns the index of the first occurrence of the supplied substring in a supplied string, starting the search from a given location in the string.

format()
Returns a string whose content is formatted based on a supplied expression.

insert()
Inserts substring into the string at the indicated offset.

left()
Extracts the leftmost n characters from a larger text string.

length()
Calculates the number of characters in the supplied string.

lineCount()
Returns the number of lines of text that are separated by carriage returns and/or line feeds.

lineText()
Returns a specific line of text from the supplied textset.

lower()
Converts the supplied text to lower case.

ltrim()
Returns the string with all leading blanks removed.

mid()
Extracts a portion of text from a specified position for a specified number of characters from a larger text string.

repeat()
Returns the given string repeated a given number of times based on the value of the second parameter.
reverse()
Reverses the order of the characters in the supplied text.

right()  Extracts a portion of text from the right.

rtrim()  Returns the string with all trailing blanks removed.

space()   Creates a string of blanks with the number of blanks in the string based on the value of the parameter.

substitute()  Finds and replaces a specified occurrence or all occurrences of a string of text.

trim()     Removes the white space from both the left and right of the supplied text.

upper()    Converts the supplied text to upper case.

Translation functions
Translation functions allow you to retrieve and manipulate different translation tables in visual dashboards.

getAvailableTranslationTables()
Returns a list of available translation tables.

Data editing functions
DataQuant's flexible design allows you to edit tables using the built-in Table Editor.

The Table Editor allows you to add, delete, and change entire rows and also supports changes to individual cells within a table. You can choose to apply your changes immediately after each edit or batch multiple edits into a single transaction.

Application development
DataQuant provides the capability to incorporate product functions and features into applications that you already have in place.

For example, you can provide access to DataQuant queries and reports from within a broader, in-house Web application.

You can incorporate DataQuant functions into existing applications in three ways:

• Embed DataQuant queries, reports, and dashboards within Web applications, portlets, or custom Web pages by using DataQuant for WebSphere

• Use Java APIs and Web service APIs to incorporate DataQuant features into Java-based workstation and Web applications

• Automate DataQuant functions by using the Command Library Interface, which provides a way to perform operations outside of the user interface (for example, you can create an external, custom application that generates and exports reports). You can use the command interface to save time and resource by batching query and report operations.
Procedures

In addition to its API functions, DataQuant provides built-in procedure development capabilities.

A procedure is an object that enables you to perform multiple functions with a single RUN command. Procedures can take advantage of sophisticated data and object management. For example, procedures can:

- Run queries
- Print reports
- Import data
- Export data
- Perform other functions

You can create, edit, and run procedures from both DataQuant for Workstation and DataQuant for WebSphere. Like other DataQuant objects, procedures are stored in the repository and are accessible through the DataQuant workspaces.
Chapter 4. Ease of use

DataQuant presents a comprehensive package of functions in a single, easy to use program.

Providing a robust set of functions is not enough in today’s business intelligence environment. Users at all levels within the organization must be able to quickly and easily retrieve and work with only the data that they need on a day-to-day basis, without time-consuming phone calls to IT staff for over-the-shoulder assistance. IBM DataQuant stands out in the business intelligence software market not only because of its robust function, but because this function is packaged in a very intuitive, easy-to-use interface.

Zero-coding architecture

DataQuant provides quick, off-the-shelf methods as well as more detailed ways to customize reports and group, aggregate, and arrange results in a data grid before generating a report.

DataQuant’s multiple query-building interfaces, explained in “Queries” on page 15, make it easy for all users to build the queries that retrieve just the data they need, regardless of SQL skills. By using the colorized SQL editor with content assistance and parameter hints, even users who are not familiar with SQL can develop a SQL query, and can switch to one of the other interfaces any time they need to.

Role-based user interfaces

DataQuant is specifically designed to accommodate the roles that are found in organizations that rely on business intelligence software. A perspective in DataQuant for Workstation provides the views, menus, and wizards that are commonly used by a particular type of business intelligence user.

The following perspectives are available:

**Administrator**

The Administrator perspective provides the views, menus, and wizards that enable a user with administrative privileges to create and maintain repositories.

**Visual designer**

The Visual Designer perspective provides the views, menus, and wizards that are used to create visual reports and dashboards.

**User**

The User perspective provides the views, menus, and wizards that are used to query data sources, access or construct reports, and view dashboards.

**BIRT report designer**

The BIRT report designer perspective provides the views, menus, and wizards that are used to run Eclipse BIRT based reports.

Users can switch between the four different perspectives according to how their security privileges are set. (For more information about setting security privileges, see Chapter 6, “Security,” on page 51.)
Context-sensitive views display the information that you need to work with when you need to work with it. Each perspective of DataQuant for Workstation initially displays the views that are most commonly used to perform specific functions. DataQuant for Workstation includes the following views:

**Properties view**
Displays the properties of the object that is selected or active in the editor

**Repositories view**
Displays the repositories that you have created

**Repository Explorer view**
Displays the contents of the repositories to which you have access
It also displays information about the references that one object has on other objects.

**Users and Groups view**
Displays the users and groups that you have created for your internal security structure

**Repository Connections view**
Displays the list of repository connections that you have available

**Search view**
Enables you to search for information within the workstation application

**Workspaces view**
Lists the workspaces that are available to your user ID
All of the data sources that you can access are contained in the workspaces that are presented.

**Personal view**
Displays recently used and favorite objects, as well as objects that open automatically upon startup.

**Outline view**
Displays an outline of the active object in the editor
The structural elements that are listed depend on the type of object that is open.

**Output view**
Displays informational and error messages that are generated while running DataQuant for Workstation

**Palette view**
Displays the objects that you can insert in a visual project

**Progress view**
Displays all progress, percentage of work completed, and cancellation messages

**Help view**
Displays help information

**Events view**
Displays navigation elements that you can apply to layout objects within a dashboard

**Project Explorer view**
Displays each element in a visual report or a dashboard
All views have their own menus. You can change the layout of a perspective by opening and closing views and by docking them in different positions in the main workstation window.

**Customizable workspaces**

Database administrators and analysts might want to see reports filed under the data sources that power them, whereas sales and marketing users might want to see reports organized by territory (a folder for each region, for example). DataQuant allows the same entities to be labeled and organized into differing taxonomies on a per-user and per-group basis.

Workspaces keep everything you need readily available and enable you to control access to repository information. Workspaces also allow users to work offline to help conserve resources. Initially, all users have access to the default workspace. Administrators can add data source information to the workspace based on the operational needs of a user or group as well as security privileges, then assign the appropriate users and groups that are allowed access to the workspace. Users can further refine the layout of their workspaces to suit their preferences and needs, which makes it easy for them to find and reuse just the objects they need to work with.

**Support for virtual data sources**

DataQuant includes support for a metadata layer that makes it easier for users to query enterprise data and build reports and dashboards that draw from it.

DataQuant virtual data sources allow administrators to optionally shield their users from the complexities of the underlying database structures, and provide content developers with a simplified data model against which content can be developed. Virtual data sources work by introducing a metadata layer which mediates between an administrator-defined, virtual data source and the underlying data sources that contain the physical tables and views. This streamlines content development by replacing obscure column names with easy to understand alternatives and representing complex table joins as a single virtual table. Also, administrators can define multiple virtual tables in a single virtual database, each of which draws data from one or more tables within differing data sources. To users, a given virtual data source acts as a single database, allowing users to author queries against all tables contained within it, despite their underlying sources residing in differing data sources. Finally, virtual data sources insulate users from database schema changes, allowing database administrators to make changes to the underlying database schemas without affecting existing dashboards, queries, or reports.

**Intuitive navigation**

Each workspace contains one or more object repositories, which organize metadata objects (queries, reports, and dashboards) as well as data sources into a hierarchy.

This hierarchy makes it easy to see relationships and find what you need. Within their individual repositories, users can customize object names and provide definitions so that other users can easily recognize the purpose of existing queries and reports and run them without the need for further training or assistance. DataQuant supports the full range of sharing permissions, from view only, to view/edit, to full control (which provides capability to view, edit, or delete an object).
Graphical, drag-and-drop object development

DataQuant offers an extremely intuitive, graphical drag-and-drop design environment for visual reports and dashboards.

DataQuant supports a wide variety of drag-and-drop mechanisms for report generation. For example, DataQuant’s graphical query diagram view allows less-experienced users to join tables by dragging and dropping related columns, effectively “wiring” them together. If users require additional assistance, they can use wizards to help them create SQL and OLAP queries.

The query designer allows you to drag and drop dimensions and measures of interest into queries and immediately see the results of those selections. After the query is run, query results can be manipulated using a drag-and-drop, fully interactive data-analysis grid. This grid supports adding calculated columns (for example, adding a Total column that sums three columns in the query results). You can also use the grid to rearrange columns and group and aggregate the data, including using pivot functions. Selecting from the graphical palette, you can also drag and drop charts, graphs, selectors and other graphical elements onto the canvas and wire data to them, as appropriate. You can then size and position each element and manipulate properties such as format, color, and options. This capability allows you to customize items such as fonts, legends, and logos.

Preview capabilities are available for queries, reports, and dashboards, which maximizes iterative development and thus reduces the overall time it takes to develop a particular object.

The Web client and desktop client environments share a common look and feel, which allows users to move from one interface to the other easily and reduces the need for additional training and its associated time and expense.

Safety net for database modifications

Security permissions prevent inadvertent modification or deletion of sensitive objects. Beyond setting security privileges, however, DataQuant provides users with a safety net when they attempt to make interdependent changes to items in reports and dashboards. Before a user changes an aspect of a dashboard or report, DataQuant lists the dependent objects that are affected by the proposed change.

Sample data to help users learn

Sample data (that includes dashboards, reports, and queries), is included in the product to help users learn how to author DataQuant objects. Cheat sheets for key tasks provide step-by-step instructions and make it easy to get started immediately. One of these cheat sheets leads the user through the process of loading the sample data. Additional cheat sheets explain how to create queries, reports and a dashboard that also use the sample data.

Built-in user assistance

Both DataQuant for Workstation and DataQuant for WebSphere provide online help for both users and administrators. The help is procedural as well as contextual and is linked directly to the application in both environments.
Wizards guide creation of queries and a colorized SQL editor with content assist and parameter hints makes it possible to develop SQL statements even without a full command of the language.
Chapter 5. Administration

From installation to applying regular maintenance to managing the day-to-day volume of user activity, IBM DataQuant administration is straightforward and efficient.

Administrator and designer interfaces

The Administrator and Visual Designer perspectives are interfaces that group together all of the objects and actions that administrators need to perform, providing fast access to everything you need to work with.

You can define and customize workspaces for your users by individual, group, or role and assign security permissions in this manner also, so that you never have to worry that users are accessing something that is beyond either their skill levels or the scope of their authorizations and privileges. For more information about DataQuant perspectives, see “Role-based user interfaces” on page 43.

Operational metadata

The day-to-day operations of a database administrator revolve around metadata. Information about who accessed what, why, and when, or what objects are consuming too much resource must be readily available to keep the organization moving forward. DataQuant provides metadata about objects, users, and groups in a format that is easy to understand and use, yet robust enough to meet stringent audit standards.

DataQuant provides metadata that contains information that you can use to analyze and report on your data operations. You can browse metadata for queries, reports, and dashboards as well as the database tables, views, and queries that are used by them. Default reports that track object usage and user activity are available, as explained in “Activity logs” on page 55.

You can search across the entire object repository for a specific query, report, or dashboard and obtain metadata about all objects and their interdependencies.

User and group metadata is drawn directly from an external LDAP source or, for those that do not have an LDAP-based directory server, from DataQuant's internal directory facility. All other data is available directly from within the centralized DataQuant repository and is accessible through both DataQuant for Workstation and DataQuant for WebSphere.

Sharing and reuse of objects

DataQuant’s emphasis on reuse allows you to deploy a consistent look and feel in reporting across your enterprise when you need it, which saves time and energy and reduces IT costs.

You can use database objects efficiently only if you have a system that helps you effectively organize, search, share, and reuse them. To allow you to spend more time gaining insight from queries, reports, and dashboards, and less time retrieving and formatting data, all query, report, and dashboard objects are essentially templates that are applied to the data at run time. You can save these templates to
a server for indexing and reuse the templates in different situations. For example, a report that was designed to output the salary history for the sales team can be reused to display salary history for technical support personnel in a different division. The report is not tightly bound to a given entity and can be applied to entirely different queries. (If aggregation has been used in the report, the new query must generate an equal or greater number of columns with the same data types.)

To allow for even greater flexibility, template queries, reports, and dashboards can be written generically, with variables as placeholders that are then replaced with actual values at run time. You can set substitution variables, whose values are used for a particular SQL query at run time, or global variables, which can be set and then used by several objects (for example, queries, reports, or procedures) for the duration of the entire DataQuant session. Variable values can be set to be retained between sessions or re-initialized to default values, entirely at the discretion of the content author or administrator.

Custom code can also be stored in the repository and reused in queries, reports, and dashboards.

Not only does DataQuant allow you to reuse objects and code, it also caches selection criteria for reuse in all available windows and dialogs, which allows you to reuse prior choices without retyping.
Chapter 6. Security

A very robust function set combined with the ability to easily query and format data calls for a highly flexible set of security features that is robust in its own right.

IBM DataQuant security is easy to set up and administer because it follows industry-standard protocols.

**Authentication methods**

DataQuant uses the industry-standard LDAP protocol for user authentication and, optionally, any data sources provided to each user.

LDAP-integrated security drives how administrators personalize workspaces, resource governing, object tracking, and user logon mappings for each data source that a user needs to access.

Single sign-on is also supported through third party and custom solutions. Authentication information can be passed from a third-party application to DataQuant, or DataQuant can be set up to automatically retrieve the HTTPS authentication information from the server.

**Administrator-controlled access**

The DataQuant interface allows you to make as much or as little viewable and available to your users as you want.

For example, you can configure DataQuant to display only those objects the user has the security permissions to view, or even limit the display of a database table to a subset of columns that you want the user to see.

Users can create and password-protect their own repositories so that they can navigate and easily manage the objects that they have access to.

**Granularity**

The full range of security permissions (view only, view/edit, or view/edit/delete) can be applied to every object in the DataQuant object repository, including individual database columns.

Because permissions are granted by administrators, users cannot inadvertently share content with other users who might not have the same access privileges. Administrators can control permissions at either a user or group level, thereby allowing role-based security structures. This role-based design is shown in the following figure:
Shared objects can also be personalized to display in different ways to different users. For example, a database table might show all columns to a faculty group, yet only show a subset of the columns to a student user group; a report can be structured so that it hides from the student user group a bar chart with cumulative test scores, yet displays it for members of the faculty.

**Integration with existing security architecture**

DataQuant is designed to work alongside firewalled networks without impacting your existing security architecture or complicating installation.

In addition, DataQuant interacts with other security solutions you may already have in place, such as RACF® on the zSeries platform.

**Security of Web clients**

DataQuant for WebSphere’s capability to operate over HTTPS connections ensures that client/server transmissions remain secure in all query, reporting, analysis, and dashboard operations.

---

*Figure 6. Assigning security permissions. DataQuant security can be role-based by group or tailored to individual users.*
Encryption support

DataQuant supports the encryption and decryption capabilities of the JDBC driver and underlying databases. Also, DataQuant offers its own encryption and decryption functions, which are derived from the Twofish algorithm, a license-free encryption algorithm that is publicly available. DataQuant's built-in encryption functions can perform additional security operations in both dashboards and reports, such as protecting data that was entered by dashboard users or decrypting information on a per-user basis.

Transactional data integrity

Because DataQuant data is maintained in standard database tables, critical information is preserved through the transactional system in place for the database server. In cases of system failure, data integrity is maintained at all times.
Chapter 7. Performance and tuning

DataQuant keeps track of system performance and lets you tune your resources for maximum efficiency.

Regardless of the scale of your database architecture, you need to understand and be able to track who is accessing what data and when. You also need the capability to ensure that queries or other operations don't consume too many database resources and bring your entire operational data system to a grinding halt when you can least afford it. IBM DataQuant has many built-in protective measures that help you tune your system for optimal performance.

Resource control functions

DataQuant includes a comprehensive resource governing facility that allows you to manage the load placed on your database servers.

Resource limits allow you to control a broad array of parameters, including:

- Maximum number of rows that a query can return
- Maximum number of bytes that a query can return
- Maximum number of database connections to be used for a given data source
- List of SQL verbs allowed (for example, you could prevent users from issuing DROP, ALTER, UPDATE and DELETE statements)
- Ability to retrieve large object (LOB) data
- Ability to edit database data
- Ability to export query results to external file formats
- Ability to save query results as a database table

To implement these resource limits, you define a collection of rules and restrictions, collectively known as a resource limit group. You then apply the limits either universally (for all users and all data sources at all times) or on a more limited basis that applies different limits to different users, groups, and servers depending on the time of day. For example, you can create rules that limit users to 1,000 rows of data per query during working hours and to 5,000 rows during non-working hours. You might also choose to apply more stringent rules to production servers and less stringent rules to test servers.

In addition to allowing you to control resource usage, DataQuant controls performance by automatically paging query data and fetches additional rows on demand as you browse through reports or query result sets.

Activity logs

Activity logs let you track all aspects of system usage.

The ability to determine who has accessed what information and when can be vital in the day-to-day operations for some organizations. An easy and efficient way to identify resource abuse or inefficient queries or reports is key. DataQuant contains built-in auditing mechanisms that help organizations track and run metrics on the
usage of queries (including what parameters were passed), reports, and dashboards on a per-user, per-group, or per-database basis. The activity log records the following information:

- The execution time of an object
- The initiating user
- The SQL text issued to the database
- The number of rows returned
- The number of bytes returned
- Various other run time parameters

If you see that resources are being used inefficiently, you can optionally use DB2 Query Patroller (in the Linux, UNIX, and Windows environments) or built-in database tuning facilities (such as those available in DB2 for z/OS, such as materialized query tables, or MQTs) to ensure that queries and reports meet your response-time standards.

Detailed tracking information is stored in standard database tables, which allows you to search for and extract data of interest and spot trends and problems quickly. The DataQuant administrator interface provides a wide variety of default performance and schedule summaries to get you started. You can customize these to meet specific needs or you can use DataQuant’s query and reporting interface to create precisely what you need. Default reports include:

- Performance summary (maximum time) by query
- Performance summary (average time) by query
- Performance summary (cumulative) for a given query
- Performance details for a given query
- Performance profile for a given user
- Performance summary (cumulative) for a given user
- Time-period performance summary (cumulative) by user
- Cancellation summary by user (user-cancelled queries)
- Cancellation details for a given user
Chapter 8. Scalability

IBM DataQuant scales to your operational needs on all levels, both in terms of the skill and knowledge level of your user base as well as the availability and performance demands of your day-to-day operations.

Skill levels

DataQuant supports users of any skill level through easy to use graphical interfaces and comprehensive help systems.

In most large organizations, only a small percentage of the staff has a working knowledge of database design and use. When relatively few people within an organization have the programming skills that are necessary to navigate their way around a database unassisted, database administrators and other IT personnel typically spend a disproportionate amount of time supporting users rather than designing and deploying thoughtful, effective solutions that meet the objectives of the organization.

By using DataQuant, users with little or no database knowledge can still be productive with graphical user interfaces that walk them through each step of building a query. Users with more sophisticated skills can query or update data directly through the SQL editor and grid- or form-based table editing interfaces. Both the query development tools and OLAP editor generate SQL statements and make the auto-generated SQL available to the user. In the case of relational queries, users can modify the SQL and return to the diagram or prompted-query interfaces for further refinements, which allows them to switch back and forth between the various development facilities.

DataQuant includes sample data accompanied by cheat sheets, which allow users with little or no knowledge to be productive quickly and go in and out of learning mode as they create what they need to do their jobs.

DataQuant’s rich security and personalization model tailors available content based on users’ security levels as well as their levels of expertise. For example, a technical user might want to see available content in a database-centric view, whereas non-technical users might want to see the same data presented in a business and role-specific form (for example, organized by operational hierarchy, such as a teacher’s reports that are organized by year and course). Complementing the user- and group-specific personalization, DataQuant also contains distinct perspectives that tailor the environment for users, database administrators, and content designers. See “Role-based user interfaces” on page 43 for more information about perspectives.

Users can subscribe to reports and can schedule their own reports, all without the assistance of an administrator. Users access reports through a shared DataQuant repository. The repository provides a report view that is tailored to the individual’s administrator-granted group membership (for example, a sales group versus a database-administrator group). Users can access their allocated reports at any time; they can also create their own schedule for report generation using DataQuant’s built in job scheduler, or an external scheduler, such as the Windows scheduler or cron on UNIX or Linux systems.
Throughput capability

DataQuant is designed to handle the throughput of the zSeries platform and can handle very large amounts of data.

Built-in resource governing functions, which are explained in Chapter 7, “Performance and tuning,” on page 55, allow you to ensure that resource loads remain within defined parameters as data flow scales to handle greater operational demands.
Chapter 9. Integration with existing tool sets

IBM DataQuant works with other business intelligence tools you already have in place, reducing the need for additional planning and programming and increasing your time to value in implementing a customized solution that fits your business needs.

How DataQuant fits into the DB2 Warehouse solution

DataQuant fits into the broader business intelligence solution offerings from IBM.

At the lower level, these offerings include extract, transform, and load (ETL) tools, as well as data quality and cleansing tools. DataQuant also works with custom analytical and performance management features that are offered by DB2 Alphablox. This wide variety of compatible and complementary business intelligence offerings provides future upgrade paths that can accommodate any reporting infrastructure needs that arise.

Export support

DataQuant reports can be used with a wide variety of third-party tools.

You can export reports in a variety of formats and transmit them through e-mail or direct file copy to a designated location (such as a Web server or FTP site). File export formats are listed in “File formats and delivery” on page 27. DataQuant also supports the use of style sheets that you might already have in place.

DataQuant content can be directly embedded within custom-developed or third-party Web or desktop applications. You can embed the content by using DataQuant’s API libraries (see “API support”) or by using fully-formed URLs for a given query, report, or dashboard.

API support

DataQuant contains a Java API library that provides the ability to load and execute queries, reports, and procedures through external or custom-developed Java applications.

In addition to its API library, DataQuant offers the ability to call external stored procedures or other external functions with the Execute Shell command.

For more information about application development capabilities, see “Application development” on page 41.
Chapter 10. Flexibility and customization

With IBM DataQuant, you get all the function you need without sacrificing the ability to customize and tailor that function precisely to your operational needs.

Flexibility

DataQuant offers you flexibility in visual design, the capability to work offline, and the ability to schedule database activity on your own terms.

Unlimited design options

Based on the award-winning Eclipse graphical editing framework, DataQuant's visual design environment is extremely flexible, allowing developers to graphically place charts, selectors, controls, and graphical primitives on the report canvas and embed static content such as text, graphics, hyperlinks, and supporting information (data-driven or static).

With over 20 default charts and graphs, over 100 analytical functions, and broad visual palettes of dozens of primitives and other dashboard and report objects, there is much to choose from to accommodate any visual design need.

Designers can nest elements (queries, reports, and dashboards) within each other infinitely, allowing a level of detail that exceeds what would ever be required in an operational environment. DataQuant's dynamic object properties also provide full control of the data elements, down to the individual items in charts. Developers can control the appearance and behavior of these items as a function of the data and user interactions.

Working offline

DataQuant uses offline schemas to allow you to work independent from the system.

Today's users are increasingly mobile. To support this trend, DataQuant allows offline schemas, which allow users to operate from a snapshot of relevant database tables. This snapshot is typically a collection of tables that the user requires in order to perform his or her job. The schema snapshot allows users to make use of the full complement of development aids as they develop queries and reports, all without actually connecting to the database. A database connection is only required when actually executing the reports or queries. In addition, user- and group-level data caching allows data to be saved for offline use on the local workstation. This feature also provides a means of running independent of the database itself. Data in the cache can be configured to expire after a specific, administrator-set time period has passed.

Scheduling queries, reports, and procedures

Using DataQuant's built in job scheduler or an external scheduling facility (such as the built-in Windows scheduling facility or Cron for UNIX or Linux), you can schedule generation and distribution of managed reports.

Reports can be submitted in a variety of ways – on demand, based on a workflow event, according to a predefined schedule, or based on conditional expressions you
specify – and can be distributed – in a variety of file formats and ways (see “File formats and delivery” on page 27 for more information).

In addition, DataQuant offers procedure objects that can perform one or a series of actions at a point in time. For example, a DataQuant procedure can be used to batch operations, allowing you to run a collection of queries, generate a series of reports, export to PDF and HTML, and publish the content through e-mail, over the Web, or to a designated network or local path. These procedures can be scheduled using DataQuant’s built in job scheduler or an external scheduling facility such as the built-in Windows scheduling facility or cron for UNIX or Linux.

Customization

DataQuant offers powerful customization and personalization capabilities.

Customizing workspaces by user or role

Customization is possible based on user and role.

Users can create and tailor their own workspaces according to their needs and preferences, then share these workspaces with other users according to an administrator-defined set of security permissions. Additionally, all views are customizable based on user or role. Users can define workgroups and publish reports and other objects just to a particular workgroup.

Customizing dashboard behavior based on security privileges

Dashboards can be personalized by tailoring visibility and behavior of dashboard elements in accordance with the logged-on user.

For example, graphs and charts can be limited to specific users or groups, or fields can be defined as editable by some users or groups, yet read-only for others. Full behavior can be modified using simple object property expressions.

Customizing views of data using virtual data sources

Administrators can customize how users view data by using virtual data sources.

Virtual data sources provide users with a simpler view of often complex and highly technical underlying data models. Administrators can customize the querying process by providing virtual tables with easier to understand table and column names that only include the fields that the user needs. Also, a single virtual data source may contain tables that draw data from differing databases, allowing users to author federated queries against disparate data.
Chapter 11. Multilingual support

DataQuant supports a wide base of national languages for truly global business interaction.

Global organizations require products that offer multilingual support for all staff to be as productive as possible. The IBM DataQuant product and online help supports all languages shown in the following figure:

Figure 7. National Language Support for DataQuant

The DataQuant Version 1.2 publications “Getting Started” and “Installing and Managing” (which are available in PDF, HTML, and BookManager® formats) support, in whole or in part, all languages shown in Figure 7 except Czech, Danish, Hebrew, Swedish, Italian, and Korean.
Appendix A. Accessibility

Software accessibility features help those with physical disabilities, such as restricted mobility or limited vision, use their computers.

Note: The DataQuant for Workstation application provides the best accessibility functionality. Users with physical disabilities who must make use of the DataQuant accessibility features should install and run DataQuant for Workstation, as opposed to DataQuant for WebSphere.

Standard keyboard equivalents in DataQuant for Workstation

Keyboard equivalents use keyboard keys to perform mouse actions instead of using a mouse.

The keyboard is the most frequently used alternative for performing mouse functions. DataQuant supports the standard keyboard equivalents that include:

- Shortcut, or accelerator keys, to perform the most frequently used functions in pull-down menus without going to the menu. For example:
  - Ctrl+S to save
  - Ctrl+P to print
  - Ctrl+R to run a query

  Shortcuts display in the pull-down menu next to its function.

- Mnemonics, or access keys, are available to perform each function on a menu or window. A mnemonic for a function is the underlined character in the function name. For example:
  - Press F to open the File menu
  - Press O to open the Open window

  Use ALT to activate the mnemonic and move the keyboard focus.

For more information, refer to the documentation for your operating system for a complete list of standard keyboard equivalents.

Standard keyboard navigation

Keyboard navigation refers to using keys instead of a mouse to move from item to item on your screen.

The movement is usually in the order specified by the operating system or your application. DataQuant applications follow the standards with regard to the typical keys used for keyboard navigation such as using TAB and SHIFT+TAB to move between controls, and arrow keys to move up, down, and sideways between items.

Operating system accessibility

Most operating systems have a set of accessibility options that enable individuals with disabilities to customize system-wide settings.

DataQuant applications:

- Inherit their settings from the operating system
Assistive technology products

DataQuant applications support assistive technology products, such as screen readers and voice synthesizers.

Each DataQuant application interface requires special navigation when using a screen reader with query results. See “Navigation in DataQuant for Workstation” or “Navigation in DataQuant for WebSphere” on page 68 for more information.

Navigation in DataQuant for Workstation

The following table shows how to accomplish various tasks using keystrokes.

<table>
<thead>
<tr>
<th>To:</th>
<th>Process:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate mnemonics in menus</td>
<td>Press the Alt key. Mnemonics in each menu are activated.</td>
</tr>
<tr>
<td>Obtain online help</td>
<td>Press F1 to open a specific help task or Alt+H to open the Help menu. Use the arrow keys to move through the menu selections. Press Enter to open a menu command.</td>
</tr>
<tr>
<td>Display keyboard shortcuts</td>
<td>Press Ctrl+Shift+L.</td>
</tr>
<tr>
<td>Left to right switch between editor windows</td>
<td>Press Ctrl+F6</td>
</tr>
<tr>
<td>Right to left switch between editor windows</td>
<td>Press Ctrl+Shift+F6</td>
</tr>
<tr>
<td>Move focus through each element</td>
<td>Press TAB to move forward or SHIFT+TAB to move in the opposite direction.</td>
</tr>
<tr>
<td>Emulate clicks on a link</td>
<td>Use the TAB key to jump from link to link, then press the Enter key to open the link.</td>
</tr>
<tr>
<td>Emulate clicks on a button</td>
<td>Use the TAB key to jump from button to button, then press Space key to activate the button.</td>
</tr>
<tr>
<td>Within a window, activate the default action</td>
<td>Press the Enter key.</td>
</tr>
<tr>
<td>Within a window, cancel the action</td>
<td>Press the ESC key.</td>
</tr>
<tr>
<td>Left to right switch between views</td>
<td>Press Ctrl+F7</td>
</tr>
<tr>
<td>Right to left switch between views</td>
<td>Press Ctrl+Shift+F7</td>
</tr>
<tr>
<td>Left to right switch between perspectives</td>
<td>Press Ctrl+F8</td>
</tr>
<tr>
<td>Right to left switch between perspectives</td>
<td>Press Ctrl+Shift+F8</td>
</tr>
<tr>
<td>Left to right switch between SQL, Diagram, Prompted and Results views in the active editor</td>
<td>Press Ctrl+Page Up</td>
</tr>
<tr>
<td>To:</td>
<td>Process:</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
</tr>
<tr>
<td>Right to left switch between SQL, Diagram, Prompted and Results views in active editor</td>
<td>Press Ctrl+Page Down</td>
</tr>
<tr>
<td>Activate command bar</td>
<td>Press Ctrl+Shift+C</td>
</tr>
</tbody>
</table>
| Use screen reader for query results or Use browser to display query results in high contrast mode for the visually impaired | 1. Run the query.  
2. Select Alt+R to display the Results menu.  
3. Select Export.  
4. The Export Query Results wizard opens.  
5. Use the TAB key to move through controls. Use arrow keys to move through selection items in the control. Select HTML from the Export Type field. Specify where to save the file in the File name field.  
7. Open the *.htm file in a browser.  
| or  
1. Run the query.  
2. Select Alt+R to display the Results menu.  
3. Select Export.  
4. The Export Query Results wizard opens.  
5. Use the TAB key to move through controls. Use arrow keys to move through selection items in the control. Select CSV from the Export Type field. Specify where to save the file in the File name field.  
7. Open the *.csv file in supportive software such as Microsoft Excel or Lotus® 1-2-3®. |
| Add object or condition | 1. Tab to put focus on a pane, such as in the Prompted Query dialog.  
2. Tab to put focus on the Add icon.  
3. Press the spacebar to display the Add dialog. |
| Select multiple objects | 1. Tab to the list box.  
2. Press Shift+Arrow to select additional, consecutive rows.  
3. Press Shift+Enter to add the objects. |
| Edit rows in list boxes (only for rows that can be edited) | 1. TAB to the list box, such as the Global Variables window.  
2. Press the spacebar or use the up and down arrow keys to select the row.  
3. Press F2 to activate edit mode.  
4. Use TAB to move between columns and rows.  
5. Press Enter to accept edits.  
6. Press TAB to exit the list box and move to the next control in the dialog. |
To: Process:

**Edit query results**
1. Press CTRL+F7 or CTRL+SHIFT+F7 until either the Workspaces or Repository Explorer view is highlighted.
2. Use the down or up arrow keys to select the folder that contains a table.
3. Use the right arrow key to expand folders and the left arrow key to collapse folders.
4. Use the arrow keys to expose and select the table that you want to edit.
5. Press the Context Menu key (the key to the left of the right-side of the CTRL key).
6. Press the mnemonic key for Open With (the h key).
7. Use the down arrow key to highlight Table Editor.
8. Press the ENTER key. The table opens in Table Editor.

**Change font for query results**
1. Run the query.
2. Press Alt+A to select all.
3. Press Alt+R to display the Results menu.
4. Press F to display the Font window.

---

**Navigation in DataQuant for WebSphere**

The following table shows how to accomplish various tasks using keystrokes.

<table>
<thead>
<tr>
<th>To:</th>
<th>Process:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move focus through each element</td>
<td>Press TAB to move forward or SHIFT+TAB to move in the opposite direction.</td>
</tr>
<tr>
<td>Emulate clicks on a link</td>
<td>Use the TAB key to jump from link to link, then press the Enter key to open the link.</td>
</tr>
<tr>
<td>Emulate clicks on a button</td>
<td>Use the TAB key to jump from button to button, then press Space key to activate the button.</td>
</tr>
<tr>
<td>Within a dialog, activate the default action</td>
<td>Press the Enter key.</td>
</tr>
<tr>
<td>Within a dialog, cancel the action</td>
<td>Press the ESC key.</td>
</tr>
<tr>
<td>Move focus to the command line</td>
<td>Press ALT+C.</td>
</tr>
<tr>
<td>Move focus to the main menu</td>
<td>Press the left ALT key. Navigate through the menu commands using the arrow keys.</td>
</tr>
<tr>
<td>Open context menus</td>
<td>Press the context menu key if you have an extended keyboard. Press the right Ctrl key. Navigate through the menu commands using the arrow keys.</td>
</tr>
<tr>
<td>Activate top-level menu</td>
<td>Press Shift+Ctrl+1 (Shift+Ctrl+2, Shift+Ctrl+3, and so on)</td>
</tr>
<tr>
<td>Switch between open editors</td>
<td>Press Shift+Ctrl+F6</td>
</tr>
<tr>
<td>Switch between open views</td>
<td>Press Shift+Ctrl+F7</td>
</tr>
</tbody>
</table>
Appendix B. Prerequisites and installation information

This topic covers the software, memory, storage, and other prerequisites you need to have in place before using either DataQuant for Workstation or DataQuant for WebSphere.

DataQuant for Workstation system requirements

Before installing DataQuant for Workstation ensure that your environment meets the following minimum requirements.

**Hardware requirements**

Each workstation on which DataQuant for Workstation runs should accommodate the following minimum requirements:

- Network connectivity
- Approximately 1 GB of disk space (single language installation)
- Minimum of 1 GB of RAM

**Software requirements**

DataQuant for Workstation can run on any of the following operating systems:

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Microsoft Windows Server 2008
- Microsoft Windows Vista
- Microsoft Windows 7
- Red Hat Enterprise Linux WS 5
- SUSE Linux 10 and 11
- Mac OS X V10.5 and V10.6

The following functions require additional support:

- Adobe Flash Player 10.1.51 or later on the client (required if deploying visual dashboards using the optional Flash output format)
- Database connectivity requires an appropriate JDBC driver for each type of RDBMS accessed.
- DataQuant for z/OS requires a QMF Enterprise Edition Version 8 (5625-DB2), Version 9 (5635-DB2) or Version 10 (5605-DB2) license in order to both access QMF objects and enable the QMF interoperability features described in this information.

**Database servers (subject to license restrictions)**

DataQuant for Workstation supports the following database servers:

- DB2 V10 LUW
- DB2 V9 LUW
- DB2 V10 z/OS VUE
- DB2 V9 for z/OS VUE
• DB2 V10 z/OS*
• DB2 V9 for z/OS*
• DB2 UDB for z/OS V8*
• DB2 Server for VSE and VM V7.3 and V7.4*
• DB2 for iSeries V5.4, V6.1, and V7.1
• DB2 for Linux, UNIX, and Windows V9.1, V9.5, and V9.7
• Informix Dynamic Server (IDS) Version 9.x, 10, and V11
• Model 204 V7R1 for z/OS
• Model 204 V7R1 for z/VM®

*z/OS, OS/390®, and VSE & VM support is limited to DataQuant for z/OS.

**Prerequisites**

The following DataQuant for Workstation functions have software dependencies:

- Database connectivity requires an appropriate JDBC driver for each type of RDBMS that is accessed.
- DataQuant for Workstation OLAP support requires InfoSphere® Warehouse V9.5.2 or later or other MDX-based OLAP servers with support for XMLA connectivity.
- DataQuant for z/OS requires a QMF Enterprise Edition Version 8 (5625-DB2), Version 9 (5635-DB2) or Version 10 (5605-DB2) license in order to both access QMF objects and enable the QMF interoperability features described in this information.
- Exporting result sets to Microsoft Excel requires a minimum level of Excel 2003. If you are running a pre-2003 version of Microsoft Excel, the Export to Excel feature is not supported.

**DataQuant for WebSphere system requirements**

Before installing DataQuant for WebSphere, ensure that your environment meets the following minimum requirements.

**Hardware requirements**

DataQuant for WebSphere requires 500 MB hard disk space and 1 GB of RAM.

50 MB RAM per user is the minimum requirement, accessing DataQuant for WebSphere.

**Software requirements**

DataQuant for WebSphere has the following software requirements:

- One of the following application servers:
  - WebSphere Application Server Versions 6.1, WebSphere Application Server V7 or WebSphere Application Server V8 on any platform supported by WebSphere Application Server.
  - Apache Tomcat V5.5 and later
- One of the following Web browsers (with JavaScript support enabled) on each client:
  - Microsoft Internet Explorer V6.0 or later
- Mozilla Firefox V3.0 or later
- Apple Safari browser V5.1 or later
- Google Chrome browser
- Java Runtime Environment (JRE) V1.5 or later
- Adobe Flash Player 10.1.51 or later on the client.

Required if deploying dashboards that use the optional Flash output format.
Required if deploying visual applications. Visual applications use Flash mode only.
- An appropriate JDBC driver for each type of database that you want to access
- DataQuant for z/OS requires a QMF Enterprise Edition Version 8 (5625-DB2) or Version 9 (5635-DB2) license in order to both access QMF objects and enable the QMF interoperability features described in this information.

Database servers (subject to license restrictions)

DataQuant for WebSphere supports any database that provides a JDBC driver.

DataQuant for WebSphere supports the following database servers:
- DB2 V10 LUW
- DB2 V9 LUW
- DB2 V10 z/OS VUE
- DB2 V9 for z/OS VUE
- DB2 V10 z/OS*
- DB2 V9 for z/OS*
- DB2 UDB for z/OS V8*
- DB2 Server for VSE and VM V7.3 and V7.4*
- DB2 for iSeries V5.4, V6.1, and V7.1
- Informix Dynamic Server (IDS) Version 9.x, 10, and V11
- Model 204 V7R1 for z/OS
- Model 204 V7R1 for z/VM

*z/OS, OS/390, and VSE & VM support is limited to DataQuant for z/OS.

Prerequisites

The following DataQuant for WebSphere functions have software dependencies:
- Database connectivity requires an appropriate JDBC driver for each type of RDBMS that is accessed.
- DataQuant for WebSphere OLAP support requires InfoSphere Warehouse V9.5.2 or later, or other MDX-based OLAP servers with support for XMLA connectivity.
- IBM DataQuant for z/OS requires a QMF Enterprise Edition Version 8 license (5625-DB2), or Version 9 license (5635-DB2) in order to both access QMF objects and enable the QMF interoperability features described in this information.
- Exporting result sets to Excel requires a minimum level of Excel 2003.

If you are running a pre-2003 version of Excel, the Export to Excel feature is not supported.
Installation and configuration overview

You can be productive quickly with DataQuant by following a few simple installation and configuration steps.

1. Install JDBC drivers that are appropriate for the data sources that you need to access.

   DataQuant for Workstation and DataQuant for WebSphere use JDBC to connect to all database data sources. A JDBC driver is included with DB2 for Linux, UNIX, and Windows; you need to perform this step only if you need to access a non-DB2 data source or if you want to use a different JDBC driver with DB2 on Linux, UNIX, and Windows.

2. Install and run the self-extracting setup file for the DataQuant platform that you are running.

   For DataQuant for Workstation, simply save the file on the desktop client machine where you want to install DataQuant. For DataQuant for WebSphere, deploy the DataQuant application to your Web application server as you would for any other application.

3. Provide access to your data sources.

   DataQuant for Workstation and DataQuant for WebSphere provide easy-to-use interfaces specifically designed to help you perform administrative tasks. To complete the DataQuant installation and setup, use the DataQuant for Workstation administrator interface or the DataQuant for WebSphere administrator interface to:
   a. Configure access to JDBC drivers.
   b. Define and name storage to hold the DataQuant repositories that your users will be creating. The Create New Repository Storage wizard guides you through this process.
   c. Create repositories within the repository storage. Each repository that you create can have its own independent objects, security models, and users. You can create these repositories or users can set them up for themselves.
   d. Define a connection to each repository, which includes information about the connection. For example, for shared repositories, the repository connection information includes the name of the JDBC driver that is used to connect to that data source.
   e. Configure data source information within each repository by importing data source settings or by using the New Data Source wizard.

4. Define one or more workspaces. A workspace is a tailored view of a user's assets. Workspaces can be role-based, group-based, or user-based. The New User Workspace wizard guides you through this process.

5. Add data sources to the workspace and set workspace security permissions.

   For more information about how to perform any of these tasks, see Installing and Managing DataQuant or Getting Started with DataQuant. After workspace security permissions are set, users can perform any of the query, reporting, and data visualization tasks that are explained in this information.
Appendix C. Purchasing options

The packaging of IBM DataQuant offers simple selection and pricing to match the on-demand information needs of any enterprise, whatever end-user and database platforms are in use. One ordering option is available for z/OS and three options are available for multiplatform environments.

All packages include both DataQuant for Workstation and DataQuant for WebSphere, although the platforms where you can install and run each component might vary from one package to another.

Table 1. Purchasing options for DataQuant

<table>
<thead>
<tr>
<th>Package name</th>
<th>Databases that you can access</th>
<th>Prerequisite products</th>
<th>Pricing structure</th>
<th>Where you can install</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataQuant for z/OS (program number 5697-N64)</td>
<td>Databases on all platforms that are supported by DataQuant (including zSeries)</td>
<td>None</td>
<td>Processor-based pricing</td>
<td>DataQuant for Workstation: Windows, Linux, Solaris</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DataQuant for WebSphere: All platforms that are supported by WebSphere Application Server</td>
</tr>
<tr>
<td>DataQuant for Multiplatforms (program number 5724-R90)</td>
<td>Databases on all platforms that are supported by DataQuant (with the exception of zSeries)</td>
<td>None</td>
<td>Value-unit pricing</td>
<td>DataQuant for Workstation: Windows, Linux, Solaris</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DataQuant for WebSphere: All platforms that are supported by WebSphere Application Server</td>
</tr>
<tr>
<td>DataQuant for DB2 Warehouse Starter Edition (program number 5724-R90)</td>
<td>Databases on Windows and Linux</td>
<td>DB2 Warehouse Starter Edition Version 9.1 or later</td>
<td>Value-unit pricing or per-user pricing</td>
<td>Windows and Linux</td>
</tr>
<tr>
<td>DataQuant for DB2 Warehouse Intermediate Edition (program number 5724-R90)</td>
<td>Databases on Windows and Linux</td>
<td>DB2 Warehouse Intermediate Edition Version 9.1 or later</td>
<td>Value-unit pricing or per-user pricing</td>
<td>Windows and Linux</td>
</tr>
</tbody>
</table>
A 60-day trial version of the product is available from the DataQuant page of the following Web site:

http://www.ibm.com/software/data/db2imstools/db2tools/dataquant/

Ensure that you download the correct trial version for your environment. Separate trial versions are available for z/OS and multiprocessor environments.
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

**Trademarks**

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
Index

A
access 15
accessibility 65
assistance technology 66
keyboard equivalents 65
keyboard navigation 65
navigation in WebSphere 68
navigation in Workstation 66
operating system 65
ACROSS support in reports 24
activity logs 55
ad-hoc capabilities
forecasts 21
queries 15
reports 23
administrator interface 43
AES (Advanced Encryption Standard) support 53
aggregation functions 24
algorithm, encryption 53
alignment properties in reports 23, 26
ALTER statements, restricting 55
analysis functions
- drilling capabilities 19
- OLAP queries (multidimensional) 18, 19
- statistical functions 39
analytical queries 20
APIS (application programming interfaces)
- encryption support 53
- Java libraries 41, 59
append query 20
arrows, embedding in reports 26
audit capabilities 55
authentication methods 49, 51
authorization 45
Average function 24

B
background options in reports 23
bar graphs 25
batch capabilities 41
BIRT report designer 43
bitmap (.bmp) image support 26
BMP file format 26
boldfacing text in reports 23
break points in reports 23
browsers
- data security 53

C
caching
- offline schema support 61
- users’ selecting criteria 49
calculation expressions 23, 24
candlestick charts 25
cascading changes, warnings 46
cells in tables, editing 41
character functions in reports 23
data access
- configuring 72
- methods 15
data cube support 18, 19
data paging 55
data sources 28
database servers
- activity logs 55
dashboards
default types 25
embedding queries 15
metadata 49
client platforms 9
cluster graphs 25
color
- in SQL editor 16
- reports 23
column charts 25
command-line interface 41, 59
comments, sending to IBM vii
conditional formatting 16, 19, 23, 62
configuration overview 72
connectivity
- offline options 61
- security of Web clients 53
conservation of resources 55
constants in reports 23, 24
content-assist function for SQL queries 16
Count function 24
cron interoperability 57, 61
cross-tab functions 24
crosstab query 20
CSV file format 15, 19
cubing support 18, 19
Cumulative percentage functions 24
Cumulative sum function 24

dashboards
calculation expressions 24
charting functions 23
conditional formatting 23
definition 28
ease of use 43
embedding in applications 41
functional overview 28
graphing functions 23
group, summary, and aggregation functions 24
mapping functions 23
metadata 49
nesting layouts 16
outline view 43
output view 43
preview capabilities 46
searching 43
sharing 49
substitution variables 49
visual attributes 23
Visual Designer perspective 43
zero-coding methodology 43
data access
- configuring 72
- methods 15
data cube support 18, 19
data paging 55
data sources 28
database servers
- activity logs 55

© Copyright IBM Corp. 2007
database servers (continued)  51
authentication and security  51
encryption support  53
integration of multiple sources  28
organizing objects by  45
SQL privileges  45
stored procedure support  59
tuning  55
types, levels supported  15
database tables  41
editing  41
schema snapshots for offline work  61
DataQuant for WebSphere system requirements  70
DataQuant for Workstation system requirements  69
DB2  59
DB2 Warehouse
DataQuant interoperability  59
OLAP queries  18
packaging options  73
dBase III file format  15
decryption/encryption support  53
DELETE statements, restricting  55
delivery of reports  27
design attributes  28
dashboard  28
drag-and-drop development  46
graphical interfaces  46, 61
tabular reports  23
Visual Designer perspective  43
visual reports  23
device-independent (.dib) image support  26(diagram queries  16
DIB file format  26
dimensions, analyzing  18
directory lookup  49
drag-and-drop interfaces
dashboards  28
queries  16
visual design environment  23, 46, 61
Draw Query command  16
drilling capabilities  18, 19
DROP statements, restricting  55
e-mailing query results  15
EAR files, support for  15
ease of use  43
audit capabilities  55
change reporting  46
drag-and-drop  46, 61
foreign-language support  63
granular security  51
graphical development  46, 61
help and wizards  47
integration with existing software  59
metadata dictionaries  49
navigation  45
object naming  15
offline mobility  61
personalization  45
role-based user interfaces  43
sample data  46
sharing objects  49
skill levels  57
with firewalled networks  52
ease of use (continued)
zero-coding  43
editing data
cascading-change warnings  46
database tables  41
LOB support  15
query results  41
efficiency of resources  55
efficiency of resources, ellipses, embedding in reports  26
embedded objects  16, 23
encryption/decryption support  53
end-user interfaces  43
environments supported  9
ETL tools, interoperability with  59
event-band reports  25
events, adding to dashboards  43
Excel support  19
Execute Shell command  59
expiration of cached data  61
explorer, project  43
Exporting data
LOB support  15
locations  59
expressions
calculation  23, 24
conditional  23
null values and  16
report scheduling with  61
F
Features
rich client  13
thin client  13
feedback, sending to IBM vii
file formats
import formats for pictures  26
query results  15
reports  27
firewalls, installation with  52
First function  24
font, changing
query results  18
reports  23
footers in reports  23
forecasts  21
foreground options in reports  23
foreign-language support  63
formatting reports
conditional expressions  23
graphics  25
grouping, summary, aggregation  24
OLAP-based  18
saving format as template  27
visual attributes  23
forms, repository storage for  72
formulas in queries, reports, dashboards  24
G
geographic maps  25
GIF file format  26
governing functions
LOB support  15
resource usage  55
graphical interfaces
  dashboard design 28
  query 16
  role-based 43
  table editor 41
  visual design environment 23, 46, 61
graphics interchange format (.gif) image support 26
graphics, embedding in visual reports 26
graphs
  default types 25
  metadata 49
grids
  data analysis 46
  embedding queries 15
group memberships
  authentication 49
  customizing repositories 57
  data caching 61
  metadata 49
  resource control 55
  security assignments 51
  viewing 43
  workspace assignment 45, 62, 72
grouping functions 24
H
headers in reports 23
help functions 43, 47
hierarchy charts 25
horizon charts 25
HTML file format 15, 19, 27, 61
HTTPS support 51, 53
hyperlinks in reports 23
I
images, importing into reports 26
inheritance of security privileges 62
installation
  overview 72
  sample tables 46
  with existing firewalls 52
integrity of data 55
interactive reports 28
interfaces
  role-based design 43
  toolbars, palettes, and views 43
Intermediate Edition packaging 73
international language support 63
DXF file format 15, 19
J
Java API libraries 41, 59
JDBC drivers
  configuring 72
  support for 15
join query 20
joins, creating 16
joint photographic experts group (.jpeg) image support 26
JPEG file format 26
L
labels in reports 26
languages translated 63
large object (LOB) support 15
Last function 24
layouts, nesting 16
LDAP (Lightweight Directory Access Protocol) support 51
limits on database resources 55
line charts 25
linear maps 25
lines, embedding in reports 26
Linux support 9
LOB (large object) support 15, 55
logical expressions in reports 23
logon mappings for authentication 51
logs, database activity 55
M
maps
  default types 25
  metadata 49
  mathematical functions 23, 24, 30
  matrix charts 25
  Maximum function 24
  menus available 43
  metadata
    access to 49
    objects, storage 45, 72
  Microsoft interoperability
    Excel 19
    SQL Server 15
  Minimum function 24
  missing values in query results 16
  multi-pass queries 16
  multidimensional analysis 18
  multilingual support 63
  multivariate charts 25
N
naming of objects 15, 45
navigation
  adding to dashboards 43
  organizing objects 45
  nesting objects 16, 61
  network, firewall support for 52
  notices
    legal 75
  null values in query results 16
  numeric operators and functions in expressions 23
  numeric separators in reports 23
O
object
  inheritance of security privileges 62
  names 15
  organizational methods 45
  outline view 43
  output view 43
  properties, viewing 43
  reuse and sharing 49
  specifying associations 28
  offline productivity options 61
reports (continued)

substitution variables 49

types
  dashboards 28
  tabular 27
  visual reports 23
visual design elements 61
Visual Designer perspective 43
visual formatting options 23
zero-coding methodology 43

repository
  creating 72
  hierarchical organization 45
  viewing 43

requirements for installation 69

resource control
  activity logs 55
  governor functions 55
  LOB support 15
response times 55
retrieval of data 15
reuse of objects
  dashboards 49
  queries 15, 49
  report formats 27
  reports 49

Rich client
  features and functions 13
  role-based design
    changeable interfaces 43
    customizable workspaces 45, 62
row-based resource control 55
rows in tables, editing 41
runaway queries, controlling 55

S

sample data 46
save formats for reports 27
scalability 57
scatter charts 25
schedule-based queries and reports 61
schemas, offline 61
script, changing in reports 18, 23
search functions
  database tables 41
  object repository 43, 49
  performance/activity reports 55
security
  access privileges 51
  authentication 51
  database activity logs 55
  granularity 51
  inheritance of privileges 62
  installation with firewalls 52
  permissions, setting 45, 72
  restricting SQL operations 55
  viewing users and groups 43
  Web clients and HTTPS 53

service information vii
setup overview 72
shapes, embedding in reports 26
sharing of objects 15
side group function 24
sign-on, single 51
simple form charts 25

single sign-on 51
skill levels of users 57
software prerequisites 69
Solaris support 9
spiral charts 25
SQL functions 24
SQL queries 16
SQL Server database support 15
Standard deviation function 24
Starter Edition packaging 73
stock charts 25
storage
  metadata objects 72
  object repository 15
storage-based resource control 55
stored procedure support 59
style sheet support 59
subordinate queries 16
subreports 16, 61
subscriptions to reports 57
substitution variables 49
Sum function 24
summary functions 24
support information vii
system prerequisites 69
system requirements
  DataQuant for WebSphere 70
  DataQuant for Workstation 69

T

table
  embedding queries 15
  joins 16
  sample data 46
  table editor functions 41
  tabular reports 27
taxonomies for objects 45
templates
  report formats 27
  reuse 49
text, embedding in reports 26
thin client
  features and functions 13
  throughput capabilities 58
time-based resource control 55
timeline reports 25
top group function 24
transactional architecture 55
translated languages 63
translation functions 41
tree charts 25
tuning capabilities 55
TXT file format 15, 19

U

underlining text in reports 23
UPDATE statements, restricting 55
URLs in reports 23
user assistance 43, 46, 47
users
  activity logs 55
  authentication 51
data caching by user 61
  skill levels 57
users *(continued)*

- viewing all 43
- workspace assignment 62

V

- variables
  - and reuse of objects 49
  - in calculation expressions 24
  - in conditional formatting 23
- views available 43
- visual dashboards 28
- visual designer interface 43
- visual reports
  - default types 25
  - design attributes 23

W

- WAR files, support for 15
- Web clients
  - security 51, 53
  - supported browsers 9
- Windows support 9
- wizard assistance 46, 47
- workflow events and schedule-based reports 61
- workspace
  - customization 45, 62
  - definition 72
  - viewing 43
- workstation clients
  - supported platforms 9
- WQML file format 15
- wrapping of text in reports 23

X

- XLS file format 19
- XML file format 15, 19
- XY graphs 25

Z

- z/OS
  - scalability 58