Contents

Preface
Maximizing Your Information Resources 13
  Information resources .............................................. 15
  Services .................................................................... 17
  Useful addresses at a glance ..................................... 18
  About this guide ......................................................... 20

Part I Introduction
Chapter 1
Introduction to BusinessObjects 23
  What is BusinessObjects? .......................................... 25
  Demo materials and samples ..................................... 30
  Upgrading from earlier versions of BusinessObjects ...... 31

Part II Accessing Data
Chapter 2
Introduction to Accessing Data with BusinessObjects 35
  Workflows for accessing data ..................................... 41

Chapter 3
Building Queries on Universes 51
  Building a basic query on a universe ......................... 56
  Building a more powerful query ............................... 63
  Running a query on a different universe .................... 72
# Accessing Data and Data Analysis

## Chapter 4 Building Queries with Other Types of Data Provider 73
- Using free-hand SQL .................................................. 75
- Using stored procedures ............................................. 88
- Using personal data files .............................................. 94
- Using Visual Basic for Applications procedures ................. 98
- Using XML files .......................................................... 105
- Using OLAP cubes ...................................................... 116
- Using Web Connect ..................................................... 117

## Chapter 5 The Web Connect Data Provider 119
- Querying the Internet .................................................. 121
- Using Web Connect ..................................................... 126
- Saving Reports ............................................................ 149
- Internet Query Grid Context Menu Options ....................... 150
- Using Prompts with Web Connect Queries ....................... 153
- Managing Queries ...................................................... 164
- Accessing secured web sites ........................................ 175
- Web Connect query examples ...................................... 183

## Chapter 6 Combining Data from Different Sources 201
- Which data sources are available? ................................. 203
- Including data from different data sources in the same report 204
- Basing a data provider on an existing data provider ............ 210
- Linking data providers ................................................. 212

## Chapter 7 Managing Data Providers 219
- Renaming data providers ............................................. 221
- Getting statistics on data providers ............................... 224
- Purging and deleting data providers ............................... 225
- Using data providers efficiently ................................... 227
Part III Analyzing Data

Chapter 8  Introduction to Data Analysis  231
On-report analysis ............................................. 233
BusinessObjects drill mode ................................. 234
OLAP servers ..................................................... 235
Slice and dice mode ............................................ 236

Chapter 9  Analyzing Data in Drill Mode  237
Using drill mode .................................................. 241
Drilling on charts ................................................ 247
Drilling on multiple hierarchies ......................... 248
Getting a different view of your data .................... 250
Analyzing measures in drill mode ....................... 254
Making copies of reports while you work ............... 256
Extending analysis .............................................. 257
Drilling using custom hierarchies ....................... 262
Qualifying data for hierarchies ......................... 266
Printing from drill mode .................................... 269
Setting options for working in drill mode .............. 270

Chapter 10 Using Slice and Dice Mode  271
Working in slice-and-dice mode ......................... 273

Part IV Customizing Report Data

Chapter 11 Filtering and Ranking Data  297
Limiting the data displayed ................................. 299
Ordering data ..................................................... 307
Using ranking to view the top and bottom values .... 315
Hiding columns and rows of data ....................... 321
Highlighting data ............................................... 323
Chapter 12  Customizing Queries on Universes  329
Creating user objects .................................................. 331
Applying complex conditions on queries ......................... 337
Using an existing query in a condition ......................... 356
Applying groups of conditions .................................. 360
Building combined queries ...................................... 365

Chapter 13  Using and Customizing Lists of Values  371
What is a list of values? .............................................. 373
How are lists of values created? .................................. 374
Customizing lists of values in BusinessObjects .............. 375
Editing lists of values .................................................. 376
Assigning personal data to a list of values .................... 380
Displaying, refreshing and purging lists of values ............ 385

Chapter 14  Creating Calculations  387
Calculations .............................................................. 389
Converting to and from Euros ..................................... 398

Chapter 15  Calculation Contexts and Extended Syntax  407
Introduction to contexts and extended syntax ................. 409
Using extended syntax for advanced calculations ............ 416
Quick reference .......................................................... 433
## Table of Contents

### Chapter 16 Calculation Troubleshooting
- #COMPUTATION .......................................................... 439
- #MULTIVALUE ............................................................. 443
- #PARITY ................................................................. 449
- #ALERTER ................................................................. 451
- #DICT.ERROR ............................................................. 452
- #DIV/0 ...................................................................... 454
- #ERROR ................................................................. 455
- #IERR ................................................................. 456
- #OVERFLOW ............................................................. 459
- #SYNTAX ................................................................. 460
- #UNKNOWN .............................................................. 461
- Tips and tricks ......................................................... 462

### Chapter 17 Formulas, Local Variables and Functions
- Formulas ................................................................. 467
- Local variables .......................................................... 472
- Creating local variables by grouping values ......................... 476
- Managing formulas and local variables ................................ 481
- Functions ................................................................. 483
- Function equivalents in Microsoft Excel ................................. 487
- More examples of using formulas ...................................... 490

### Part V Appendix

#### Appendix A
- Launching BusinessObjects with the Run Command ............... 503
  - Using the Run command ........................................... 505

#### Appendix B
- BusinessObjects and Visual Basic for Applications ............... 509
  - Using macros ....................................................... 511
  - Using add-ins ....................................................... 514
  - Converting scripts to macros ...................................... 516
  - Using the Visual Basic editor ..................................... 517
Examples

How do the official figures compare with my personal targets? ............ 45
Adding regional information to an existing document ....................... 47
Building a query in the Query Panel and running the query .................. 61
Limiting query results by using a condition ................................... 66
Create eFashion report that shows sales by store and category in Florida . 82
Create prompted eFashion report on sales by state, store and category . 85
Creating a report showing article sales by state ................................. 90
Accessing an Outlook inbox using VBA ......................................... 100
Creating an Internet query ......................................................... 138
BUSINESSOBJECTS prompts you to link data providers ...................... 212
Reports showing revenue by country and resort, revenue by country ...... 227
Why is revenue better in this resort than in the others? ....................... 238
Move from analyzing Resort to Sales ........................................... 244
Analyzing profit margin ............................................................... 254
Focus analysis on high-profile resort using drill filters as query conditions 259
Filter data to show sales revenue for two regions only ...................... 299
Display only those stores with weekly revenue over $200 000 ............. 304
How can I get months to sort correctly? ....................................... 310
Sort customers by nationality and then by name in alphabetical order ... 312
Display the three top-selling product lines only ............................... 315
Show sales revenue for top three months, compare with overall revenue . 319
Which sales representatives generate revenue over $500 000 ............. 323
Obtaining total ordered revenue by creating a user object .................. 331
Which customers made reservations for 2001 and 2002? ................... 337
When did each customer last pay for a product? ............................ 345
Which customer made the earliest reservation? ............................ 354
Which customer made the earliest reservation? (using calculation) ....... 355
Return list of resorts/revenues where resort country revenue > $1000000 356
Which customers bought a given product in a given time period? ........ 363
Which customers bought products in both 2001 and 2002? .............. 368
Accessing Data and Data Analysis

Examples

- Showing cities and regions in a list of cities .............................................. 376
- Displaying total revenue and subtotals ....................................................... 392
- Displaying average, maximum and minimum revenue ................................. 394
- Calculating the difference in revenue between two quarters ....................... 394
- Displaying total revenue as a table title ..................................................... 395
- Converting to euros: six-digit conversion rates ............................................ 398
- Adding US dollars to the currency list ....................................................... 403
- Triangulation: converting between EMU-compliant currencies .................... 404
- Revenue per region per year, and revenue per region ................................... 410
- Calculate revenue in various default contexts ........................................... 411
- The extended syntax of an aggregate formula ............................................. 413
- Calculating the number of cities per region ................................................ 416
- Calculating the minimum revenue per city for each region ......................... 417
- A formula containing input and output contexts ......................................... 419
- Calculating running total revenue per country ........................................... 420
- Using ForEach and ForAll ............................................................................. 423
- Calculating a grand total by using the Report keyword ................................. 430
- The formula BusinessObjects writes for a simple calculation ....................... 431
- #COMPUTATION resulting from a running sum with a reset context ............ 439
- #COMPUTATION caused by a conditional formula in a break footer ............. 441
- #MULTIVALUE in an aggregation ................................................................. 443
- #MULTIVALUE in a break footer ................................................................. 445
- #MULTIVALUE in a section containing name and address ............................ 447
- Solving #IERR by turning part of a formula into a variable ......................... 457
- Calculating a running total ........................................................................... 470
- Highlighting above-average margin .............................................................. 474
- Group quarters to display revenue per semester .......................................... 476
- Ranking cities according to sales revenue .................................................... 484
- Calculating a 3-week rolling average ........................................................... 490
- Combining first and last names in a single cell ............................................ 491
- Combining text and data in a single cell ....................................................... 492
- Combining text and numbers in a single cell .............................................. 493
- Combining text and dates in a single cell .................................................... 494
- Comparing yearly margin growth using the Where function ....................... 494
- Using function output as input to another function .................................... 496
Determining the first and last days of the previous month . . . . . . . . . . . . . 496
Calculating total revenue for all resorts when some are filtered out . . . . . 499
Examples
Maximizing Your Information Resources
Overview

Information, services, and solutions

The Business Objects business intelligence solution is supported by thousands of pages of documentation, available from the products, on the Internet, on CD, and by extensive online help systems and multimedia.

Packed with in-depth technical information, business examples, and advice on troubleshooting and best practices, this comprehensive documentation set provides concrete solutions to your business problems.

Business Objects also offers a complete range of support and services to help maximize the return on your business intelligence investment. See in the following sections how Business Objects can help you plan for and successfully meet your specific technical support, education, and consulting requirements.
Information resources

Whatever your Business Objects profile, we can help you quickly access the documentation and other information you need.

Where do I start?

Below are a few suggested starting points; there is a summary of useful web addresses on page 18.

► Documentation Roadmap

The Documentation Roadmap references all Business Objects guides and multimedia, and lets you see at a glance what information is available, from where, and in what format.

View or download the Business Objects Documentation Roadmap at www.businessobjects.com/services/documentation.htm

► Documentation from the products

You can access electronic documentation at any time from the product you are using. Online help, multimedia, and guides in Adobe PDF format are available from the product Help menus.

► Documentation on the web

The full electronic documentation set is available to customers with a valid maintenance agreement on the Online Customer Support (OCS) website at www.businessobjects.com/services/support.htm

► Buy printed documentation

You can order printed documentation through your local sales office, or from the online Business Objects Documentation Supply Store at www.businessobjects.com/services/documentation.htm

► Search the Documentation CD

Search across the entire documentation set on the Business Objects Documentation CD shipped with our products. This CD brings together the full set of documentation, plus tips, tricks, multimedia tutorials, and demo materials.

Order the Documentation CD online, from the Business Objects Documentation Supply Store, or from your local sales office.
Multimedia

Are you new to Business Objects? Are you upgrading from a previous release or expanding, for example, from our desktop to our web solution? Try one of our multimedia quick tours or Getting Started tutorials. All are available via the Online Customer Support (OCS) website or on the Documentation CD.

How can I get the most recent documentation?

You can get our most up-to-date documentation via the web. Regularly check the sites listed below for the latest documentation, samples, and tips.

Tips & Tricks

Open to everyone, this is a regularly updated source of creative solutions to any number of business questions. You can even contribute by sending us your own tips.

www.businessobjects.com/forms/tipsandtricks_login.asp

Product documentation

We regularly update and expand our documentation and multimedia offerings. With a valid maintenance agreement, you can get the latest documentation—in seven languages—on the Online Customer Support (OCS) website.

Developer Suite Online

Developer Suite Online provides documentation, samples, and tips to those customers with a valid maintenance agreement and a Developer Suite license via the Online Customer Support (OCS) website.

Send us your feedback

Do you have a suggestion on how we can improve our documentation? Is there something you particularly like or have found useful? Drop us a line, and we will do our best to ensure that your suggestion is included in the next release of our documentation: documentation@businessobjects.com

NOTE

If your issue concerns a Business Objects product and not the documentation, please contact our Customer Support experts. For information about Customer Support visit: www.businessobjects.com/services/support.htm
Services

A global network of Business Objects technology experts provides customer support, education, and consulting to ensure maximum business intelligence benefit to your business.

How we can support you?

Business Objects offers customer support plans to best suit the size and requirements of your deployment. We operate three global customer support centers:

- Americas: San Jose, California and Atlanta, Georgia
- Europe: Maidenhead, United Kingdom
- Asia: Tokyo, Japan and Sydney, Australia

Online Customer Support

Our Customer Support website is open to all direct customers with a current maintenance agreement, and provides the most up-to-date Business Objects product and technical information. You can log, update, and track cases from this site using the Business Objects Knowledge Base.

Having an issue with the product?

Have you exhausted the troubleshooting resources at your disposal and still not found a solution to a specific issue?

For support in deploying Business Objects products, contact Worldwide Customer Support at: www.businessobjects.com/services/support.htm

Looking for the best deployment solution for your company?

Business Objects consultants can accompany you from the initial analysis stage to the delivery of your deployment project. Expertise is available in relational and multidimensional databases, in connectivities, database design tools, customized embedding technology, and more.

For more information, contact your local sales office, or contact us at: www.businessobjects.com/services/consulting.htm

Looking for training options?

From traditional classroom learning to targeted e-learning seminars, we can offer a training package to suit your learning needs and preferred learning style. Find more information on the Business Objects Education website: www.businessobjects.com/services/education.htm
### Useful addresses at a glance

<table>
<thead>
<tr>
<th>Address</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.businessobjects.com/services/documentation.htm">www.businessobjects.com/services/documentation.htm</a></td>
<td>Feedback or questions about documentation.</td>
</tr>
<tr>
<td><strong>Business Objects Documentation mailbox</strong></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:documentation@businessobjects.com">documentation@businessobjects.com</a></td>
<td></td>
</tr>
<tr>
<td><strong>Product documentation</strong></td>
<td>The latest Business Objects product documentation, to download or view online.</td>
</tr>
<tr>
<td><a href="http://www.businessobjects.com/services/support.htm">www.businessobjects.com/services/support.htm</a></td>
<td></td>
</tr>
<tr>
<td><strong>Business Objects product information</strong></td>
<td>Information about the full range of Business Objects products.</td>
</tr>
<tr>
<td><a href="http://www.businessobjects.com">www.businessobjects.com</a></td>
<td></td>
</tr>
<tr>
<td><strong>Developer Suite Online</strong></td>
<td>Available to customers with a valid maintenance agreement and a Developer Suite license via the Online Customer Support (OCS) website. Provides all the documentation, latest samples, kits and tips.</td>
</tr>
<tr>
<td><a href="http://www.techsupport.businessobjects.com">www.techsupport.businessobjects.com</a></td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge Base (KB)</strong></td>
<td>Technical articles, documents, case resolutions. Also, use the Knowledge Exchange to learn what challenges other users – both customers and employees – face and what strategies they find to address complex issues. From the Knowledge Base, click the Knowledge Exchange link.</td>
</tr>
<tr>
<td><a href="http://www.techsupport.businessobjects.com">www.techsupport.businessobjects.com</a></td>
<td></td>
</tr>
<tr>
<td><strong>Tips &amp; Tricks</strong></td>
<td>Practical business-focused examples.</td>
</tr>
<tr>
<td><a href="http://www.businessobjects.com/forms/tipsandtricks_login.asp">www.businessobjects.com/forms/tipsandtricks_login.asp</a></td>
<td></td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td><strong>Content</strong></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Online Customer Support</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.techsupport.businessobjects.com">www.techsupport.businessobjects.com</a></td>
<td>Starting point for answering questions, resolving issues.</td>
</tr>
<tr>
<td><a href="http://www.businessobjects.com/services">www.businessobjects.com/services</a></td>
<td>Information about registering with <strong>Worldwide Customer Support</strong>.</td>
</tr>
<tr>
<td><strong>Business Objects Education Services</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.businessobjects.com/services/education.htm">www.businessobjects.com/services/education.htm</a></td>
<td>The range of Business Objects training options and modules.</td>
</tr>
<tr>
<td><strong>Business Objects Consulting Services</strong></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.businessobjects.com/services/consulting.htm">www.businessobjects.com/services/consulting.htm</a></td>
<td>Information on how Business Objects can help maximize your business intelligence investment.</td>
</tr>
</tbody>
</table>
About this guide

This guide explains how to access and analyze data from different data sources using the BusinessObjects business intelligence software. It does not describe how to format this data. For this information, refer to the BusinessObjects User’s Guide: Reporting Techniques and Formatting guide.

Audience

This guide is intended for non-technical end users who intend to use BusinessObjects to build reports using corporate or personal data.

Conventions used in this guide

The conventions used in this guide are described in the table below.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small capitals</td>
<td>The names of all products such as BusinessObjects, WebIntelligence, Supervisor, and Designer.</td>
</tr>
<tr>
<td>This font</td>
<td>Code, SQL syntax, computer programs. For example: @Select(Country\Country Id). This font is also</td>
</tr>
<tr>
<td></td>
<td>used for all paths, directories, scripts, commands and files for UNIX.</td>
</tr>
<tr>
<td>Some code more code</td>
<td>Placed at the end of a line of code, the symbol (+) indicates that the next line should be entered</td>
</tr>
<tr>
<td></td>
<td>continuously with no carriage return.</td>
</tr>
<tr>
<td>$DIRECTORYPATHNAME</td>
<td>The path to a directory in the Business Objects installation/configuration directory structure.</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>• $INSTALLDIR refers to the Business Objects installation directory.</td>
</tr>
<tr>
<td></td>
<td>• $LOCDATADIR refers to a subdirectory of the BusinessObjects installation directory called</td>
</tr>
<tr>
<td></td>
<td>locData.</td>
</tr>
</tbody>
</table>
Introduction
Introduction to BusinessObjects
Overview

This section gives a description of BusinessObjects, introduces BusinessObjects concepts, and lists the new features in BusinessObjects 6.0 and BusinessObjects 6.1.
What is BusinessObjects?

BusinessObjects is an integrated query, reporting and analysis solution for business professionals that allows you to access the data in your corporate databases directly from your desktop and present and analyze this information in a BusinessObjects document.

BusinessObjects makes it easy to access this data, because you work with it in business terms that are familiar to you, not technical database terms like SQL.

Once you’ve used BusinessObjects to access the data you need, you can present the information in reports as simple as tables or as sophisticated as dynamic documents with drillable charts.

You can then save those documents for your own personal use, send them to other users, or publish them to the corporate repository for potentially even broader circulation.

This section gives an overview of what BusinessObjects does and how it works.

Where does the data come from?

BusinessObjects makes it easy to access data from your corporate database because it has a business-intelligent, semantic layer that isolates you from the technical issues of the database. This semantic layer is called a *universe*. A
universe maps to data in the database, using everyday terms that describe your business environment. This means you can select exactly the data that interests you using your own business terminology.

In your company or organization, universes are created by a universe designer, using BusinessObjects Designer. The designer then makes universes available to you and other users, so that you can access the data you want from the database through an intuitive, user-friendly interface.

### A BusinessObjects Universe

Using this interface, you build a BusinessObjects query using an editor called the Query Panel, by adding and organizing objects from a universe. Objects are elements that map to a set of data from a relational database in terms that pertain to your business situation. When you run the query, BusinessObjects connects to the database and retrieves the data mapped to the objects you selected.

A query is a type of data provider. The data provider contains the data you have chosen to retrieve from the data source. Using this data set, you can build interactive reports.

BusinessObjects lets you access data from a wide range of sources: from relational and multidimensional databases, from packaged applications, from personal data files, and, using Microsoft Visual Basic for Applications procedures, from virtually any source.
Presenting and analyzing data

Once you have the data you need, you can present it in a number of ways. You can present it in a simple table:

<table>
<thead>
<tr>
<th>City</th>
<th>Quantity sold</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>48,359</td>
<td>$7,502,221</td>
</tr>
<tr>
<td>Houston</td>
<td>32,934</td>
<td>$5,447,957</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>26,244</td>
<td>$4,220,929</td>
</tr>
<tr>
<td>San Francisco</td>
<td>19,960</td>
<td>$3,288,641</td>
</tr>
<tr>
<td>Chicago</td>
<td>17,976</td>
<td>$3,012,169</td>
</tr>
<tr>
<td>Washington</td>
<td>16,744</td>
<td>$2,961,960</td>
</tr>
<tr>
<td>Austin</td>
<td>17,078</td>
<td>$2,091,073</td>
</tr>
<tr>
<td>Colorado Springs</td>
<td>12,787</td>
<td>$2,080,275</td>
</tr>
<tr>
<td>Dallas</td>
<td>12,355</td>
<td>$1,970,034</td>
</tr>
<tr>
<td>Miami</td>
<td>11,267</td>
<td>$1,178,139</td>
</tr>
<tr>
<td>Boston</td>
<td>7,676</td>
<td>$1,283,707</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$96,387,203</strong></td>
<td><strong>$36,307,703</strong></td>
</tr>
</tbody>
</table>

Alternatively you can create sophisticated reports containing large amounts of data, organized and formatted to make it easy for people to go directly to the information that interests them:

![Product Sales by City](image)
You can add images and embedded objects and format your documents to high presentation standards for viewing on screen or for printing.

On-report analysis allows you to switch your business perspective by dragging and dropping data, insert on-report calculations or drill into a report for detailed information:

Sharing the information

You can quickly and easily share the documents you have created with other users in your company, either by sending them directly to selected individuals or groups, or by publishing them as corporate documents. When you distribute documents in these different ways, you use the BusinessObjects repository. The repository stores the documents you send so that other users can retrieve and view them. It also stores information about the documents it stores, such as name of sender, date and time, and also which users in the company have the right to retrieve and view a document.

You can retrieve documents that other users have sent, including WebIntelligence and BusinessQuery documents which you can open and view in BusinessObjects. You can also use Broadcast Agent to send documents for scheduled processing.
NOTE
For information on sending, retrieving, printing, and publishing and scheduling documents, see the InfoView User’s Guide. You can open an electronic version of this guide directly from the BusinessObjects Help menu.

Security

The repository is set up and administered by the BusinessObjects supervisor. All of the rights you have as a user are granted by your BusinessObjects supervisor using Supervisor. The supervisor defines:

- The parts of the BusinessObjects interface you can access. Your supervisor can restrict the availability of BusinessObjects functionality, such as access to certain menu commands.
- Your database connections
- The universes you can access for creating and editing queries

The rights accorded to each user define the user’s profile. This profile-based security system allows a single document to be distributed to many users -- with end users having access only to the information that they are authorized to see.

Keeping a document’s data up-to-date

Databases are regularly updated with new data. A document generated at a given point in time reflects the data as it existed at that time, but it may be inaccurate now. BusinessObjects lets you update the data in a document while keeping the same presentation and formatting, either manually, or automatically at specified times. When you update a document, BusinessObjects reconnects to the database, and retrieves the updated data. This is called refreshing a document.
Demo materials and samples

To help you get up and running with BusinessObjects, demonstration databases, universes and sample reports are included in the BusinessObjects demo kit. There are two demonstration universes, Island Resorts Marketing and eFashion. The Getting Started with BusinessObjects tutorial and the multimedia Quick Tour are both based on eFashion. The examples in this user’s guide are based on eFashion and Island Resorts Marketing.

The eFashion demo database contains retail data from a clothing chain. It tracks 211 products (663 product color variations), sold over 13 stores in the US, over three years. The Island Resorts Marketing universe is described in more detail in the section Demonstration materials on page 53.
Upgrading from earlier versions of BusinessObjects

This section concerns those users who are upgrading from an earlier version of BusinessObjects.

Upgrading from BusinessObjects 5.1

Documents created in BusinessObjects 5.1 are fully compatible with BusinessObjects 6.1.

Upgrading from BusinessObjects 4.1

You need to open and save documents created in BusinessObjects 4.1 in BusinessObjects 5.1 before you open them in BusinessObjects 6.0.

The SBL ReportScript macro language used by BusinessObjects 4.1 was replaced in BusinessObjects 5.1 by Visual Basic for Applications (VBA). You can convert these scripts automatically to VBA. See Converting scripts to macros on page 516.

See the “Report Basics and Report Manager” chapter in the BusinessObjects User’s Guide: Report Techniques and Formatting for information on preserving the look of your BusinessObjects 4.0 reports when you upgrade them.

What’s new in BusinessObjects 6.0?

▶ XML data provider

The XML data provider allows you to access data stored in XML files. You decide which XML elements you want to appear in the report, then build the report using the new XML Query panel. BusinessObjects then builds a report using the data in the XML file.

▶ WebConnect data provider

WebConnect was previously sold under a separate license. Now incorporated into BusinessObjects as a new data provider, it allows you to build reports that draw data from pages on the web. You select the parts of the web page that contain the data you are interested in and BusinessObjects builds a report based on this data.

▶ Save as Excel

Save As Excel allows you to save a BusinessObjects report as an Excel spreadsheet, retaining the report’s layout and formatting.
Finding objects in the Query Panel

You can now search for report objects in the Query Panel, making the navigation of large and complex reports simpler.

What's new in BusinessObjects 6.1?

With BusinessObjects 6.1 you can use the output of one query as the input to another. You build your first query, then reference it in the second. This allows for the creation of more complex queries.
Accessing Data
Introduction to Accessing Data with BusinessObjects
Overview

This chapter is about accessing data, or how to get data from your database to the reports that you create with BusinessObjects. It introduces the different data sources that are available, and how, using BusinessObjects, you can access these to get the information you need, when you need it.

What data sources are available?

BusinessObjects let you access data from a wide range of sources. You can access data from
• relational databases (RDBMS), such as Oracle, Microsoft SQL Server, Informix and IBM DB2
• multidimensional (OLAP) databases, such as Microsoft OLAP Services, Hyperion Essbase, and Oracle Express
• text files and spreadsheets
• packaged applications such as SAP
• almost any data source using Microsoft Visual Basic for Applications (VBA) procedures
• XML files
• web pages

How do you access data sources?

BusinessObjects lets you access your data through a graphical user interface. You need no technical knowledge of the underlying data structures to get the information you want. What you do need, however, is knowledge of your business.

To access a data source with BusinessObjects, you build a data provider. The types of data providers supported in BusinessObjects depend on whether you install BusinessObjects from the installation CD or install BusinessObjects via an Internet browser.

The types of data provider that BusinessObjects supports are described in the table below:
### Accessing Data and Data Analysis

<table>
<thead>
<tr>
<th>Data provider</th>
<th>Description</th>
<th>CD Install</th>
<th>Browser Install</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries on universes</td>
<td>A universe consists of classes and objects that represent the parts of a database that contain the data you need, in everyday language that is meaningful to you. In a query on a universe, you select the objects, such as Customer Name, Year, or Region. This is by far the most common type of data provider in BusinessObjects.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Stored procedures</td>
<td>You can only use stored procedures if your supervisor or IS department has provided them, and if the RDBMS at your site supports them. A stored procedure is a SQL (Structured Query Language) script, saved and executable on your database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Free-hand SQL</td>
<td>You can use free-hand SQL if you are familiar with SQL, which is the language used to interact with relational databases. In free-hand SQL, you open or write a SQL script, which you then run against the database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Personal data files</td>
<td>You can retrieve data from Excel, dBASE and text files.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>XML files</td>
<td>You can retrieve data from XML files</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VBA procedures</td>
<td>Procedures written in Microsoft Visual Basic for Applications (VBA) enable you to retrieve data from almost any data source.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
NOTE
OLAP and SAP data providers are documented in dedicated guides that you receive when you purchase your OLAP or SAP Access Pack. The other types of data providers are documented in this guide.

Can all BusinessObjects users build data providers?
You can build data providers only if you have purchased the Reporter module of BusinessObjects.

<table>
<thead>
<tr>
<th>Data provider</th>
<th>Description</th>
<th>CD Install</th>
<th>Browser Install</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLAP servers</td>
<td>You can view the contents of an Online Analytical Processing (OLAP) server, and select the data you want to display in your report. OLAP servers are multidimensional databases that store summarized data, ready for business analysis. To use OLAP servers in BusinessObjects, you must install the BusinessObjects OLAP Access Pack for the multidimensional database at your site. BusinessObjects supports the following OLAP servers: Microsoft OLAP Services Hyperion Essbase IBM DB2 OLAP Server Oracle Express</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SAP</td>
<td>BusinessObjects interfaces with SAP applications, enabling you to use data from SAP BW and SAP BAPI.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>WWW pages</td>
<td>BusinessObjects allows you to retrieve data from pages on the World Wide Web (WWW).</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Furthermore, the BusinessObjects supervisor can restrict access to certain types of data providers, or even certain objects within a universe. As a result, you might be able to build queries on universes but no other type of data provider, and then be able to use only certain objects in the universe.

The way the supervisor sets up access to data providers and other BusinessObjects features depends entirely upon the query and reporting needs of your organization.

By default, all BusinessObjects users can refresh data providers to get the latest information from their database.

**Who sets up database connections?**

In order to access and retrieve data from a database, you need a database connection. For example, if your company or organization stores its corporate data in an Informix database, someone somewhere has to make BusinessObjects “talk” to this data source.

In 99% of cases, you, the BusinessObjects end user, do not have to concern yourself with setting up database connections. Thus, BusinessObjects lets you get the information you need, and you don’t need any technical knowledge of what’s going on behind the scenes.

This does not mean that power BusinessObjects users cannot define their own database connections. For example, in free-hand SQL, you can define a connection, write a SQL script, then run the script against the connection you created.

The following table describes who sets up database connections for the various BusinessObjects data providers.
Restrictive connections

If you are working with a universe that is set up with a restrictive connection, you need to supply the database username and password to run a query. This username/password is not the one that you use to log onto BusinessObjects; it is the username/password of the underlying database (for example an Oracle or SQL Server database) that the universe accesses. This database normally remains hidden to you, but the universe designer can set up a restrictive connection.
connection to add an extra layer of security. Depending on the type of restrictive connection, you need to supply the database username and password in some or all of the following situations:

- When you first run a query (for more information on running a query, see Building a query in the Query Panel and running the query on page 65).
- When you parse a query to test its validity (for more information on parsing a query, see Using SQL from BusinessObjects queries on page 369).

You see the following dialog box, in which you enter your username/password:

If you do not know your database username and password, see your BusinessObjects administrator.

**Can you combine data from different sources in one report?**

Yes. With BusinessObjects, you can build powerful reports with data from corporate databases that you can access using queries on data providers such as universes and free-hand SQL, and data from your own files such as spreadsheets and text files.

For more information, refer to Combining Data from Different Sources on page 201.
Workflows for accessing data

There are two basic workflows for building data providers to access your data in BusinessObjects. You can build a data provider

• when you create a new document

and

• when you are working in an existing document.

Also in an existing document, you can obtain a different set of results by editing a data provider.

The following sections explain these different workflows.

Building a data provider when you create a new document

Building a data provider when you create a new document is a typical way of using BusinessObjects. You create the document in order to see your business data; to do that, you have to build a data provider to access data from a data source.

To help you build a data provider when you create a new document, BusinessObjects launches the New Report Wizard when you start the application for the first time.
To run this wizard once you have launched BusinessObjects, click New Report Wizard on the Standard toolbar.

To build a new data provider using the wizard:

1. Select an option for the report layout, then click **Begin**. The Specify Data Access dialog box appears:

2. The next step depends on what you want to do:

   **If you want to...** | **Then...**
   --- | ---
   Build a query on a universe. | 1. Click **Universe**, then click **Next**.  
   2. Go to the next step.  
   Use a stored procedure, free-hand SQL, personal data file, XML file, VBA procedure, OLAP cube or web page as a data source. | 1. Click **Others**, then select an option from the list box.  
   2. Click **Finish**.  
   The dialog box that appears will allow you to build your data provider and retrieve the data for your report.  
   3. If you chose to build a query on a universe in the previous step, the Select a
4. Select the universe that you want to use, then click Finish.

The Query Panel appears. In the Query Panel, you can view all the classes and objects in the universe you selected, and use these to build your query. For more information, refer to Displaying the query panel on page 56.

**REMINDER**

OLAP and SAP data providers are documented in dedicated guides that you receive when you purchase your OLAP or SAP Access Pack.

**Setting a default type of data provider for new documents**

Do you always use the same type of data provider when you create new documents? If so, you can set an option so that the type of data provider you always use will be preselected in the New Report Wizard. This means that you will not have to select the type of data provider you want every time you create a document.

If you always use queries on universes, you can also select the default universe to use.
To set a default type of data provider:

1. Click **Options** on the Tools menu.
   In the Options dialog box that appears, click the **New Document** tab.

2. Click **Invoke the New Report Wizard with the following settings**:

3. In the Data Access group box, select the type of data provider you want to use.
   - **Use a Default Universe** option lets you select the universe you want.
   - **Use a Different Data Provider** lets you select a data provider type from the drop-down list.

4. Click **OK** to close the dialog box.

**Building a query in an existing document**

You don’t have to create a new document every time you want to see new data in BusinessObjects. You can build data providers inside existing documents. This feature enables you not only to see more data that comes from the same source as the document’s initial query, but also to combine data from different sources in the same report.
EXAMPLE

How do the official figures compare with my personal targets?

Your company’s sales information is stored in your corporate database, which you access by running a query on a universe in BusinessObjects. You already have a BusinessObjects document containing this information.

You keep your quarterly targets in a Microsoft Excel spreadsheet and you want to compare the corporate figures with your personal data. To do this:
1. Open the document containing the corporate data.
2. Click New Data Provider on the Data menu.
3. In the Wizard that appears, click Access new data in a different way.
4. In the next screen, click *Personal data files*:

5. Click *Finish*, and in the dialog box that appears, browse to the Excel file that contains your personal data.
6. Click **Run**.

   BusinessObjects makes the data from the spreadsheet available in your report.

   **To build a data provider inside an existing document**
   1. Click **New Data Provider** on the Data menu.
   2. Follow the wizard to select the type of data provider you want.
   3. Build then run the data provider.
      - BusinessObjects retrieves the data, making it available in the document.

   **TIP**

   If you want to see the new data as soon as BusinessObjects has retrieved it, use the Table, Crosstab or Chart commands on the Insert menu, then follow the wizard to access the data you want.

---

**Editing data providers**

Editing a data provider means changing its definition in order to bring new or different data to the document you are working on. It’s often quicker and easier to edit a data provider than to build a new one.

**EXAMPLE**

**Adding regional information to an existing document**

You’re working in a document with sales figures by year, but you need some regional information to complete the picture. Rather than building a new query, which means creating multiple data providers in the same document, you can simply add result objects to the existing data provider. To do this:

1. Click **Edit Data Provider** on the Data menu.

   In the Query Panel, add the objects you want (for example Region, City) to the Result Objects box. You do this by double-clicking each object’s icon in the Classes and Objects list.

2. Click **Run**.

   BusinessObjects returns the new data to the report, and, provided that your data is displayed in a table, the new columns automatically appear.

   Other reasons for editing a data provider include:
   - You want to restrict the volume of data returned by setting conditions or maximum number of rows.
You want the data to be sorted in a given order at the query level.

To edit a data provider
1. Click Edit Data Provider on the Data menu.
2. The next step depends on whether or not the document contains more than one data provider:

<table>
<thead>
<tr>
<th>If the document contains...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>One data provider</td>
<td>The editor (Query Panel, Access Personal Data dialog box, etc.) for the data provider appears.</td>
</tr>
<tr>
<td>More than one data provider</td>
<td>The List of Data Providers dialog box appears. Select the data provider you want to edit, then click OK.</td>
</tr>
</tbody>
</table>

3. Edit then run the data provider.
   BusinessObjects returns the new data set to your report.

Cancelling data providers
Cancelling a data provider means interrupting the data provider while it is fetching data to create or refresh a report.
1. To cancel a data provider, press the Esc key.
   The Interrupted Execution dialog box appears on your screen.

2. Select which results you want to view in the report.
3. Click **OK**.

<table>
<thead>
<tr>
<th>If you</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Want to view the results that will be created by the data provider you were running,</td>
<td>click <em>Continue the execution</em>.</td>
</tr>
<tr>
<td>Want to view the partial results created by the data provider when you interrupted the execution,</td>
<td>click <em>Stop the execution and keep the partial results</em>. When you have partial results in a report, the following notification appears in the status bar:</td>
</tr>
<tr>
<td>Want to discard the results created by the data provider when you interrupted the execution,</td>
<td>click <em>Discard the results</em>.</td>
</tr>
<tr>
<td>Want to view the results of the previous execution,</td>
<td>click <em>Keep the results of the previous execution</em>.</td>
</tr>
</tbody>
</table>
Building Queries on Universes
Overview

This chapter is about accessing data by using BusinessObjects native technology: building queries on universes.

What is a universe?

BusinessObjects universes make it easy to access data because they contain objects of data in business terms that are familiar to you. What’s more, you need no knowledge of the database structure, or of database technology, to be able to create powerful reports with data that is relevant to your work.

Universes provide the business-intelligent, semantic layer that isolates you from the complexities of the database. A universe maps to data in the database in everyday terms that describe your business situation.

Universes are made up of classes and objects. For example, the objects in a human resources universe would be Names, Addresses, Salaries. Classes are logical groupings of objects. Each class has a meaningful name, such as Vacation (for objects pertaining to employee vacations). Each object maps to data in the database and enables you to retrieve data for your reports.

Who is responsible for creating universes?

In your company or organization, universes are created by a universe designer, who works with a BusinessObjects application called Designer. The designer then makes universes available to you and other users at your site, so that you can access the data you want from the database.

Two demo universes that map to demo databases are delivered with BusinessObjects. A full description of these is provided in “Demonstration materials” below.

What are universe queries?

Universe queries enable you to retrieve data from a database via a universe. You build a query to bring data to a report, either when you create the report or when you want to view new data.

When you build a query, you select objects from a universe, then run the query. BusinessObjects connects to the database, and retrieves the data mapped by the objects you selected.
Demonstration materials

Two demonstration databases, Island Resorts Marketing and eFashion, and their accompanying universes and reports are included in the BusinessObjects package. They are installed with BusinessObjects, and used in the examples in this guide. The databases are compatible with Microsoft Access 2000. The BusinessObjects CD also includes generic SQL scripts and data files to allow a database administrator to build the databases on any RDBMS.

Island Resorts Marketing

The Island Resorts Marketing universe accesses data in the club.mdb database. It is designed for an imaginary tour operator that runs beach clubs in different resorts around the world. You use it to retrieve data on sales and reservations for resorts and customers, over time. The illustration on page 54 shows the universe’s classes and objects as they appear in BusinessObjects.

Because universes provide a business-intelligent semantic layer between you and the database, the names of the classes and objects in the demonstration universe are self-explanatory. For example, the Resort class contains objects that map to data on resorts:

- The Resort object retrieves the names of the company’s resorts.
- The Service object retrieves data for the types of services in each resort: accommodation, food and drinks, recreation.
- The Service Line object retrieves data for the types of service in each resort, for example family suite (for accommodation), restaurant (for food and drinks).

For more information on classes and the different types of objects you find in BusinessObjects, refer to Classes and sub-classes and Dimension objects, measure objects and detail objects on page 54.
Building Queries on Universes

The Island Resorts Marketing demonstration universe,

- **Classes and sub-classes**
  The demonstration universe contains five classes: Resort, Customer, Sales, Reservations and Measures. The purpose of classes is to provide logical groupings of objects. For example, the Customer class contains objects that you map to data on customers in the database.

  The Customer class contains a sub-class, which is entitled Sponsor. A sub-class is to a class what a sub-folder is to a folder.

- **Dimension objects, measure objects and detail objects**
  When creating universes, universe designers define and qualify objects. The qualification of an object reveals how it can be used in analysis in reports.

  An object can be qualified as a dimension, a detail, or a measure. Each type of object serves a different purpose:
• Dimension objects retrieve the data that will provide the basis for analysis in a report. Dimension objects typically retrieve character-type data (customer names, resort names), or dates (years, quarters, reservation dates).

• A detail object is always associated to one dimension object, on which it provides additional information. For example, Address is a detail object that is associated to Customer. Address provides additional information on customers: their addresses.

• Measure objects are semantically dynamic: the values they return depend on the objects they are used with. For example, if you include Resort and Revenue in a query, revenue per resort is calculated. If you include Customer and Revenue, revenue per customer is calculated, and so on.

**eFashion**

The eFashion demo database contains retail data from a clothing chain. It tracks 211 products (663 product color variations), sold over 13 stores in the US, over three years. It contains approximately 90,000 rows of data.
Building a basic query on a universe

You can bring data to a report by building a query on a universe. You complete this task in the Query Panel, a graphical interface that enables you to build a query by dragging and dropping objects from the universe. The Query Panel is illustrated on page 57.

There are three steps in building a basic query on a universe.
1. Display the query panel
2. Build the query in the Query Panel and run the query
3. Save the query definition

Displaying the query panel

How you display the Query Panel depends on whether you’re creating a new document or building a new query inside an existing document. You can use the following commands and toolbar buttons:

<table>
<thead>
<tr>
<th>If you want</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>to create a new document,</td>
<td>click the <strong>New Report Wizard</strong> button (Standard toolbar).</td>
</tr>
<tr>
<td>to edit a query or other type of data</td>
<td>click <strong>Edit Data Provider</strong> on the Data menu.</td>
</tr>
<tr>
<td>provider in the current document,</td>
<td></td>
</tr>
<tr>
<td>to create a new query or other type of</td>
<td>click <strong>New Data Provider</strong> on the Data menu.</td>
</tr>
<tr>
<td>data provider in the current document,</td>
<td></td>
</tr>
</tbody>
</table>

If you need more information, refer to [Workflows for accessing data on page 42](#).
The Query Panel displays the contents of your BusinessObjects universe and lets you select data with simple mouse clicks.

a. Classes appear as folders.
b. Objects appear as cubes (for dimensions), spheres (for measures) or pyramids (for details).
c. This button, selected by default, displays the universe’s classes and objects.
d. This button enables you to set options before running the query, for example to specify a maximum number of rows.
e. This button displays the universe’s predefined conditions.
f. You can type a search string here to search for objects in the universe.
g. The Result Objects box displays the objects that are included in the query.
h. The Conditions box displays the conditions limiting the data returned by the query.
i. Save and Close lets you save the query you have defined without running it. You can run it later on by using the Refresh command.
j. When you click View, the raw data retrieved by the query appears in the Data Manager. From the Data Manager, you can edit, accept or cancel the query.
k. When you click Run, the query connects to the database and the data appears in the report.
Displaying the query panel
You’ve launched BusinessObjects for the first time and the New Report Wizard appears. You use the wizard to display the Query Panel for the Island Resorts Marketing universe. Here are the steps you take:

1. In the New Report Wizard, click **Begin**.
   The Specify Data Access dialog box appears, with the **Universe** option already selected.
2. Click **Next**.
   The Select a Universe dialog box appears.
3. Click **Island Resorts Marketing**.
4. Click **Finish**.

The Query Panel appears with the classes of the Island Resorts Marketing universe displayed:

![Query Panel](image)

**Building a query in the Query Panel and running the query**

Building and running a query includes the following steps:

1. Display all the objects that you can include in a query
2. Include objects in a query
3. Remove objects from a query
4. Change the order of objects in a query
5. Run the query

Steps 2, 3, and 4 are not always sequential. For example, you can include objects in a query, remove some of them, and then include other objects.

**Displaying the objects that you can include in a query**

In the Query Panel, the Classes and Objects box presents the classes, sub-classes and objects of the universe that you are using. Objects represent the data that you can retrieve via the universe. Classes are logical groupings of objects. Classes can also contain sub-classes, as folders can contain sub-folders.

When the Query Panel appears, only the universe’s classes are visible. Click the + plus to the left of a class icon to view the class’s objects and sub-classes.
Searching for objects

You can search for an object by typing its name in the search box. BusinessObjects opens the object folder and selects the object. This is a useful feature if your universe is large with many objects.

Including objects in a query

When you include an object in the query, you instruct BusinessObjects to retrieve the data for that object from the database. For example, to display revenue by resort in your report, you include the Revenue and Resort objects in the query.

You include an object in a query by placing it in the Result Objects box. There are three ways of doing this. You can:

- Click an icon in the Classes and Objects list, then drag it to the Result Objects box.
- Double-click an object in the Classes and Objects list.
- Click a class folder and drag it to the Result Objects box. All the objects in the class appear in the Result Objects box.

Once you have placed objects in the Result Objects box, you have built a basic query.

Removing objects from a query

If you decide you want to remove an object from the query you are building, click its icon in the Result Objects box. You can now remove the object by:

- Dragging the icon to the Classes and Objects list.
- Pressing the Delete key.
- Clicking your right-mouse button, then clicking Delete on the shortcut menu that appears.

Repeat to remove other objects from the query.

Changing the order of the objects in a query

The order in which the objects appear in the Result Objects box determines the order in which the data will appear in the report. To move an object, click its icon. You can now:

- Drag the icon to the left or the right, then release the mouse button.

Swap the icon with another object icon in the Result Objects box, by holding down the Shift key, dragging it until it is above the object you want to swap, then releasing your mouse button.
**Running the query**

Once you have built the query you want, you click Run to have the query retrieve the data from the database.

**EXAMPLE**

Building a query in the Query Panel and running the query

You have displayed the Query Panel for the Island Resorts Marketing universe and want to move objects from the Classes and Objects box to the Result Objects box to build your query.

1. Click the + sign next to the Resort class, the Sales class and the Measures class.
   
   Doing this reveals the objects in each class.

2. Double-click the objects you want. For example, to find out yearly revenue in each resort, double-click **Resort, Year**, and, in the Measures class, **Revenue**.
   
   The Query Panel now looks like this:

3. Click Run.
   
   BusinessObjects retrieves the data for Resort, Year and Revenue and
NOTE

If the universe designer has set the up the universe with a restrictive connection, BusinessObjects prompts you to enter your database username and password before retrieving the data. For more information on restrictive connections, see Restrictive connections on page 40.

Saving the definition of a query

You can build a query without having to run it right away. This feature lets you:

• save a query so that you can continue defining it at a later stage
• save a query that you have finished defining, but that you do not want to run right away, for example because you know network traffic is heavy

To save the definition of a query:

1. Build a query by moving objects into the Result Objects and Conditions boxes in the Query Panel.
2. Click Save and Close.
   The result objects from the query appear as column headings. You then refresh the query in order to view the data.

<table>
<thead>
<tr>
<th>Resort</th>
<th>Year</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>FY1999</td>
<td>$367,320.00</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>FY1999</td>
<td>$407,400.00</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>FY2000</td>
<td>$376,115.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>FY1999</td>
<td>$356,240.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>FY1999</td>
<td>$300,310.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>FY2000</td>
<td>$259,170.00</td>
</tr>
<tr>
<td>Hanoiian Club</td>
<td>FY1999</td>
<td>$879,685.00</td>
</tr>
<tr>
<td>Hanoiian Club</td>
<td>FY1999</td>
<td>$819,830.00</td>
</tr>
<tr>
<td>Hanoiian Club</td>
<td>FY2000</td>
<td>$460,445.00</td>
</tr>
</tbody>
</table>
Building a more powerful query

You build a simple query by adding objects to the Query Panel. The procedures described in the following sections enable you to build a more powerful query by controlling the data that your queries retrieve. You can:

- define scope of analysis, which means that you retrieve data that you will later use for analysis in the report
- limit the query results to data that satisfies conditions
- sort data, for example alphabetically
- retrieve a specified number of rows of data
- eliminate duplicate rows of data from the query result

NOTE

All the above tasks are easy to perform for non-technical end users. In "Customizing Queries on Universes" on page 329, you can find information on more powerful query building procedures that are designed for advanced users.

Defining scope of analysis

Analysis means looking at data from different viewpoints and on different levels of detail. In reports, you can use scope of analysis to ensure that the data included in your report can be displayed at the appropriate level of detail for your analysis. Setting a scope of analysis allows you to work in drill mode, which enables you to display data in progressively greater detail.

"Scope of analysis" means a subset of data, returned by a query, that you will use for analysis in your report. The data for your scope of analysis does not appear in the report until you decide that you want to use it in analysis.

The scope of analysis you can define depends on hierarchies in the universe. A hierarchy, which the designer sets up when creating the universe, consists of dimension objects ranked from "less detailed" to "more detailed". The objects that belong to hierarchies are the ones you can use to define scope of analysis.

To view the hierarchies in the universe you are working with, click the Scope of Analysis button on the Query Panel toolbar. The Scope of Analysis dialog box appears:
NOTE

If a universe contains no hierarchies, BusinessObjects uses its classes as hierarchies by default.

> To define default scope of analysis

Once you include one object that belongs to a hierarchy in a query, you can define a default scope of analysis that includes other objects at other levels from the same hierarchy. Including more levels in your scope of analysis allows you to view lower levels of detail in your analysis. For example, the Resort object belongs to the Resort hierarchy. Once you include Resort in a query, you can automatically include the Service Line and Service objects in your scope of analysis because these objects also belong to the Resort hierarchy.
To use this feature, first insert an object from a hierarchy in the Result Objects box. Then, click the arrow on the Scope of Analysis list box on the Query Panel toolbar:

This list enables you to include one, two or three objects from the hierarchy in your scope of analysis. For example, if you insert Resort in the Result Objects box, then click One Level Down, your scope of analysis contains the object below Resort (Service Line) in the Resort hierarchy. Click the option that corresponds to the number of objects you want to include in your scope of analysis. This option is now active in the Scope of Analysis list box.

When you run the query, the report displays the data for the objects that you included in the Result Objects box of the Query Panel. The data for the objects in your scope of analysis is not displayed, but it is available for use in analysis.

To define scope of analysis manually

Instead of using the default method described in the previous section, you can manually select the dimension objects you want. To do this:

1. Click the **Scope of Analysis** button in the Query Panel toolbar.
   The Scope of Analysis dialog box appears.
2. Click inside the checkbox of each object you want to include in your scope of analysis.
3. Click **OK** to return to the Query Panel.
   The Scope of Analysis list box on the Query Panel toolbar displays “Custom Level”, which indicates that you manually defined your scope of analysis.

**TIP**

You can select all the objects in a hierarchy by clicking the hierarchy check box in the Scope of Analysis dialog box.

Applying conditions

A condition is a way of limiting the data that a query returns. Here’s a simple example.
**EXAMPLE**

Limiting query results by using a condition

The Resort object retrieves five values: Australian Reef, Bahamas Beach, French Riviera, Hawaiian Club and Royal Caribbean.

You can apply a condition on the Resort object to stipulate that you want to retrieve the data for only the Bahamas Beach and Royal Caribbean resorts only.

In BusinessObjects, you can set three types of conditions on a query:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Predefined conditions</strong></td>
<td>When universe designers build universes, they can create predefined conditions for you to use. For example, the Island Resort Marketing universe contains predefined conditions such as Year 2002, which lets you obtain reservations for 2002 only. You can apply one or more predefined conditions when you build a query. However, you can neither delete predefined conditions from a universe, nor can you edit their definition.</td>
</tr>
<tr>
<td><strong>Simple conditions</strong></td>
<td>Enable you to limit data returned by a result object. For example, you can find out about certain customers by applying a simple condition on the Customer object, then selecting the customer names that appear in a dialog box.</td>
</tr>
<tr>
<td><strong>Complex conditions</strong></td>
<td>Enable you to limit the query results by any object in the universe. For more information on complex conditions, refer to Applying complex conditions on queries on page 337.</td>
</tr>
</tbody>
</table>

To apply a predefined condition

1. Click Predefined Conditions below the Classes and Objects box in the Query Panel.

The Predefined Conditions box replaces the Classes and Objects box. The predefined conditions in the Island Resorts Marketing universe are
illustrated here:

2. Double-click the predefined condition you want to apply. The condition appears in the Conditions box. When you run the query, only the data corresponding to the predefined condition appears in the report.

**TIP**

For information on using two or more conditions in the same query, refer to Using an existing query in a condition on page 356.

- **To remove a predefined condition**
  Click the condition’s icon in the Conditions box, then press the Delete key.

- **To apply your own simple condition**
  Before you can apply a simple condition on an object, you must include the object in the query. Then:
  1. Click the object icon in the Result Objects box.
  2. Click the **Simple Condition** button on the toolbar.
     The list of values for the object is retrieved from the database, and appears in
3. Hold down the Ctrl key on your keyboard, click the values you want the object to retrieve, then click OK.

The condition appears in the Conditions box.

When you run the query, only the data corresponding to the value(s) you selected will appear in the report.

To select different values for a simple condition

Once you have applied a simple condition on an object in a query, you can modify it by selecting different values for the object to return. To do this:

1. In the Conditions box of the Query Panel, click the value(s) that appear(s) on the right-hand side of the condition.

The Classes and Objects box becomes the Operands box.

2. Double-click the Show list of values operand.

The object's list of values appears in the List of Values dialog box.

3. If you want to select more values for the condition, hold down the Ctrl key and then, in the List of Values dialog box, click each value that you want the object to retrieve.

4. Click any selected values that you do not want the object to retrieve, and click OK.
To delete a simple condition
Click the condition in the Conditions box, then press the Delete key.

Applying sorts

Sorts control the order in which data appears: ascending or descending. For example, you can apply a sort on a measure object so that its data appears in ascending order, from lowest to highest values.

The following table summarizes the order in which data appears:

<table>
<thead>
<tr>
<th></th>
<th>Text</th>
<th>Numbers</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ascending</td>
<td>A-Z</td>
<td>lowest to highest</td>
<td>past to present</td>
</tr>
<tr>
<td>Descending</td>
<td>A-Z</td>
<td>highest to lowest</td>
<td>present to past</td>
</tr>
</tbody>
</table>

To apply a sort on an object
1. Click an object in the Result Objects box.
2. Click the Sort button on the toolbar.
   A sort icon appears below the object icon in the Result Objects box.

To remove a sort
There are two ways of doing this:
   • Click the sort icon then press the Delete key.
   • Drag the sort icon from the object in the Result Objects box to the Classes and Objects list, where you release your mouse button.
     In both cases, the sort icon disappears from the object in the Result Objects box.

To invert a sort
Double-click the sort icon below the object. The arrow in the sort icon appears the other way up, to indicate that you have inverted the sort.

Sorts and free-hand SQL
If you apply a sort on a query and then use the SQL statement generated by the query to create a new report, the SQL statement will ignore the sort. You need to either adjust the order of the columns in the SQL statement to create the report you want or modify the order of the columns in the report itself.
To define sort priority and apply transparent sorts

When you apply more than one sort on a query, you may want to define sort priority. You can also apply transparent sorts (sorts on objects that are not result objects) provided that the database at your site supports this feature.

To define sort priority or apply transparent sorts, click Manage Sorts on the Query Panel toolbar. The Sorts dialog box appears.

Manage Sorts  To find out more about these tasks, click Help in the Sorts dialog box.

Setting options and running a query

Before running a query, you can set options that enable you to:

• Specify the number of rows of data that you want the query to return. The Default Value option corresponds to the maximum number of rows that the universe designer specified for queries on the current universe, in the Designer module.
• Eliminate duplicate rows of data. This feature is useful if you think that the query will return many rows containing the same data.
• Retrieve no data when you run the query. In this case, BusinessObjects generates the query SQL but does not connect to the database. The names of the objects included in the query appear as column headings in the report. This option is useful if you want to save the query you have built, but refresh it at an off-peak time.

To set options, then run a query:

1. Click Options in the Query Panel.

The Query Options dialog box appears.

2. Click No Duplicate Rows if you want to eliminate duplicate rows of data from
the query result.

3. To obtain a partial result, you can:
   • Click 10 rows or 20 rows.
   • Enter a number of rows in the Other field. You can use the arrows to raise or lower the value.

4. Click Do Not Retrieve Data if you do not want the query to connect to the database when you run it.
   When you refresh the query, this option will be automatically switched off, meaning that the query will connect to the database and the data will appear in the report.
   • Click OK to return to the Query Panel.
   Once you are satisfied with the query you have built, click Run.
   The query connects to the database and retrieves the data you specified. The report that appears displays the data for the objects that you placed in the Result Objects box in the Query Panel.
Running a query on a different universe

BusinessObjects allows you to run a query on one universe and then run the same query on a different universe. By doing this, you can test your query on a pilot universe before applying it to your real data. The following procedure describes how to run a query on a different universe after having run it on an initial universe.

1. Open the report containing the query.
2. Click View Data on the Data menu.
3. The Data Manager dialog box opens.
4. Choose the query you want to use in the Data Providers list, then click the Definition tab.
5. Click the button to the right of the current universe name.
6. In the dialog box that appears, select the universe you want to use, then click OK.
7. Click the Results tab, then Refresh.
8. Click OK to close the Data Manager.
Building Queries with Other Types of Data Provider
Overview

This chapter explains how to create reports using data providers other than BusinessObjects universes. In addition to universes, you can build reports using free-hand SQL, stored procedures, personal data files, Visual Basic for Applications (VBA) procedures, XML files, OLAP cubes and Web Connect.
Using free-hand SQL

SQL is the native query and reporting language understood by relational database management systems (RDBMSs). When you create a report based on a BusinessObjects universe, the universe generates the SQL that is passed to the server, thus shielding you from the complexities of SQL queries.

Alternatively, using free-hand SQL, you can interact directly with the database by creating the SQL yourself.

Creating a report using free-hand SQL

When you create a report using free-hand SQL, you can:
- write a new script or open an existing script
- define lists of values and prompts
- create a new connection to the database or use an existing one
- view raw data before it appears in the report
- parse the script for SQL errors
- save any changes you make to a file
To create a report using free-hand SQL:

1. Click the **New Report Wizard** button on the Standard toolbar.
2. Select an option for the report layout, then click **Next**.
   The Specify Data Access dialog box appears.
3. Under **Others**, select **Free-hand SQL** from the list box, then click **Finish**.
The Free-hand SQL editor appears.

4. The next step depends on what you want to do.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>First...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write a new SQL script</td>
<td>Type the script.</td>
<td>Go to the next step.</td>
</tr>
<tr>
<td>Open an existing script</td>
<td>In the Free-Hand SQL editor, click <strong>Open</strong>:</td>
<td>Use the dialog box that appears to locate the SQL script file.</td>
</tr>
</tbody>
</table>

5. Click the **Parse** button to check the script for SQL errors. BusinessObjects runs the SQL against the database and displays any error message that the database returns.
NOTE
BusinessObjects does not execute COMPUTE and ORDER BY clauses in free-hand SQL statements.

6. To make a connection to the database:
   • Select a connection in the Connection list box, or
   • Create a new connection. (See Creating or editing a connection for free-hand SQL on page 79).

7. Click **Build Hierarchies and Start in Drill Mode** if you want to perform drill-down analysis as soon as the data appears in the report.

8. Click **View** if you want to see the raw data that the script retrieves.
   The Data Manager dialog appears with the raw data in the Results tab. Click **OK** to close the Data Manager dialog box.

9. Click **Run**.
   The data retrieved by the SQL query appears in the report.
Editing a free-hand SQL script

To get different results from a free-hand SQL script that you have already run, all you have to do is edit the script then re-run it. To do this:

1. Open the report containing data from the free-hand SQL script, then click **Edit Data Provider** on the Data menu.
   The Free-Hand SQL dialog box appears.
2. Make the changes to the script. As you work, you can:
   - Click **Parse** to check for SQL errors.
   - Click **View** to see the raw data that the script retrieves.
   - Click **Save** to save the changes you make.
3. Click **Run**.
   BusinessObjects retrieves the new data and displays it in the report.

Creating or editing a connection for free-hand SQL

To retrieve data using free-hand SQL, you need to define a connection to your database in BusinessObjects. This is not the case when you run queries on universes because the required connection is stored in the universe.

▶ Using the free-hand SQL editor to create and edit connections

You create and edit connections for free-hand SQL in the free-hand SQL editor.
To create a connection
1. Click Create a New Connection.
The Add a Connection dialog box appears.
2. Choose the driver that you will use to connect to the database, then click **OK**. The Connection Properties dialog box appears. This box varies according to the database driver you selected.

![Connection Properties dialog box](image.png)

3. Type a name for the connection in the Name box and select the RDBMS from the Database Engine list box.

4. Type the username, password and database/datasource name in the Login Parameters box.

5. In the Type list box, select Personal or Shared.
   - **Personal** means that only you can use the connection.
   - **Shared** means that other users can use the connection.

6. Click **Test** to check that the connection is correctly defined. If you receive an error message, check the parameters you have entered and try again. If you still cannot successfully create a database connection, see your database administrator.

**To edit a connection**

You can edit any connection after you have created it. To do this:

1. In the Free-Hand SQL dialog box, select the connection from the Connections
Accessing Data and Data Analysis

Building Queries with Other Types of Data Provider

1. Click the list box, then click **Edit Connection**.
   The Connection Properties dialog box appears.

2. Make your modifications to the connection.

3. Click **Test** to ensure that the modified connection is still valid.
   You can now:
   • Click **Run** to run a script against the connection.
   • Click **View** to see the raw data that the query retrieves.
   • Click **Cancel** to save the connection for future use.

Creating a report showing sales by store and category

This section gives an example of a simple report created using free-hand SQL.

**EXAMPLE**

Create *eFashion* report that shows sales by store and category in Florida

1. Start BusinessObjects

2. Click **New Report Wizard**.

3. Select an option for the report layout, then click **Next**.

4. Select Free-hand SQL from the Others list box, then click **Next**.
   The Free-Hand SQL dialog box opens.
   You now need to create a connection to the Microsoft Access *eFashion* database. To do this:

5. Click **Create New Connection**.
   The Add a Connection dialog box opens.

6. Select ODBC drivers from the list of drivers and click OK.
   The Connection Properties dialog box opens.

7. Type ‘*eFashion*’ in the Name box, select ‘*eFashion*’ from the Data Source Name list box, select ‘MS Access 2000’ from the Database Engine list box, then click **OK**.
   The *eFashion* connection you have just created now appears as the current connection in the Connection box.

8. Type the following SQL in the Free-Hand SQL dialog box:
Using free-hand SQL

Creating interactive reports using free-hand SQL

This section gives an example of a free-hand SQL script that includes a BusinessObjects prompt. When you run a report containing a prompt BusinessObjects displays a dialog box in which you specify one or more parameters to be passed to the report query. The report then returns data based on your input. Prompts are a BusinessObjects rather than an SQL feature, but the BusinessObjects free-hand SQL data provider allows you to incorporate them into an SQL query.

Creating a prompt with a list of values for a free-hand SQL script

A prompt is a question that requires users to select values when they run queries. In this way, users filter the query to get the data that is pertinent to them.

In addition, a prompt can display a list of values; the user can select from this list rather than typing directly into the prompt.
Syntax for prompts and lists of values in free-hand SQL.

You define a prompt and its list of values by including the @prompt function in the SQL WHERE clause. The syntax of the function is as follows:

@prompt (‘prompt’, ‘data type’, {‘value1’, ‘value2’, etc.}, mono/multi, free/constrained)

The following table describes each function component:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@prompt</td>
<td>The @prompt function, which can take up to five arguments. The only mandatory argument is ‘prompt’. If you omit an argument, BusinessObjects supplies its default value. Even if you omit an argument, you must still include the commas that precede and follow it. Thus, the syntax for a prompt in which only the first argument is specified is as follows: @prompt(‘Which year?’,,,)</td>
</tr>
<tr>
<td>prompt</td>
<td>The text that appears in the prompt box when you run the report. This argument takes a character string enclosed in quotes, for example ‘Select a customer or customers’</td>
</tr>
<tr>
<td>data type</td>
<td>The type of data that the prompt returns (character, number or date). This argument can be one of the following three values enclosed in quotes: • ‘A’ for character data • ‘N’ for numeric data • ‘D’ for date data</td>
</tr>
</tbody>
</table>
Accessing Data and Data Analysis

Using free-hand SQL

EXAMPLE

Create prompted eFashion report on sales by state, store and category

2. Select an option for the report layout, then click Next.
3. Choose ‘free-hand SQL’ from the Others list box, then click Next.
   The Free-Hand SQL dialog box appears.
4. If necessary, create a connection to the eFashion database (see Creating a report showing sales by store and category on page 82 for an explanation of how to do this).
5. Type the following SQL into the Free-Hand SQL dialog box:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>value1, value2...</td>
<td>The list of values displayed when you run the report. The list can consist of up to 256 character strings enclosed in single quotes, for example: ‘London’, ‘New York’, ‘Paris’ If you do not include this argument you will have to type values directly into the prompt.</td>
<td>N/A</td>
</tr>
<tr>
<td>mono/multi</td>
<td>Specifies whether the user can select one or multiple entries from the list of values. This argument takes one of two values: • mono, which prevents multiple selection • multi, which allows multiple selection</td>
<td>mono</td>
</tr>
<tr>
<td>free/constrained</td>
<td>Determines whether users can enter values directly. This argument takes one of the following parameters: • free - user can enter values directly • constrained - user must select values from the list of values</td>
<td>constrained</td>
</tr>
</tbody>
</table>
SELECT ol.shop_name as shop_name,
al.category as category,
SUM(sf.quantity_sold) as quantity_sold
FROM outlet_lookup ol INNER JOIN (shop_facts sf INNER JOIN article_lookup al
ON sf.article_id = al.article_id)
ON ol.shop_id = sf.shop_id
WHERE state = @prompt ('Choose a state', 'A', {'California', 'Illinois', 'Florida'}, multi, constrained)
GROUP BY ol.shop_name, al.category

6. Click Run.
The Enter or Select Values dialog box appears.

7. Click Values.
The List of Values dialog box appears.

8. Select a state or states from the list and click OK.
BusinessObjects generates the report based on the states you selected.
Restrictions on free-hand SQL scripts

The types of SQL script that you are allowed to run as free-hand SQL are determined by your BusinessObjects administrator. If you attempt to run a script for which you do not have permission, you will receive an error message. Typically, you are able to run scripts that contain only one SELECT statement. See your BusinessObjects administrator if you need to run scripts that are more complex or that make changes to database data.
Using stored procedures

This section describes stored procedures and explains how to use them to bring data to your BusinessObjects reports.

What are stored procedures?

Stored procedures are SQL scripts—ranging from simple to extremely complex—that are stored as executable code in an RDBMS. They can receive arguments and return data.

How do you use stored procedures in BusinessObjects?

In BusinessObjects, stored procedures are data providers like universes or free-hand SQL. In the New Report Wizard, you select the stored procedure that you want to use. When you run the report you enter data for any input parameters that the procedure has and the procedure returns data to BusinessObjects which BusinessObjects presents as a report.

Restrictions on stored procedures

- The BusinessObjects supervisor grants access to the database or account where stored procedures are located.
- Not all RDBMSs support stored procedures. Consult your database guide to see if yours does.
- COMPUTE, PRINT, OUTPUT or STATUS statements contained in stored procedures are not executed.

Using a stored procedure to retrieve data

This section demonstrates how to retrieve data into a BusinessObjects report using a stored procedure. The following example uses a stored procedure that returns data from the eFashion database running on Microsoft SQL Server. The procedure takes the state and article name as input parameters and returns a list of shops within the state and their total sales of articles with names similar to the one specified. The query in the stored procedure is as follows (@state and @article are parameters passed to the procedure):
SELECT ol.shop_name, al.article_label, 
    SUM(sf.quantity_sold) as total_sold
FROM outlet_lookup ol
    INNER JOIN ( 
        shop_facts sf INNER JOIN article_lookup al 
        ON sf.article_id = al.article_id 
    ) sf
    ON ol.shop_id = sf.shop_id
WHERE ol.state = @state
    AND al.article_label LIKE '%%' + @article + '%%'
GROUP BY ol.shop_name, al.article_label
Creating a report showing article sales by state

2. Select an option for the report layout, then click **Next**. The Specify Data Access dialog box appears.
3. Click **Others**, then select **Stored procedures** from the list.
4. Click **Next**, then select a connection.

5. Click **Next**, then choose the stored procedure.

6. Click **Finish**.
The Stored Procedure Editor appears.
If the stored procedure has input parameters, supply values for each parameter by typing its value in the Values box.

For each parameter

<table>
<thead>
<tr>
<th>If you want...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>To reuse the value you typed the next time you run the report</td>
<td>Select <em>Use this value</em> in the Next Execution dropdown</td>
</tr>
<tr>
<td>BusinessObjects to prompt you for a value the next time you run the report</td>
<td>Select <em>Prompt me for a value</em> in the Next Execution list</td>
</tr>
</tbody>
</table>

7. Click **Run**.

BusinessObjects runs the stored procedure and places its data in a report.

<table>
<thead>
<tr>
<th>shop_name</th>
<th>article_label</th>
<th>total_sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-Fashion Chicago 32nd</td>
<td>Ladies Eton Belt</td>
<td>25</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Belt Buckle</td>
<td>197</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Belted Pocket Shirt</td>
<td>21</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Leather Belt</td>
<td>1</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Panache Belted Dress</td>
<td>13</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Patchwork Leather Belt</td>
<td>9</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Straight Belted Shirt</td>
<td>41</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Suede Belt with Heart Shaped Buckle</td>
<td>16</td>
</tr>
<tr>
<td>e-Fashion Chicago 33rd</td>
<td>Suede Cloth Belt</td>
<td>1</td>
</tr>
</tbody>
</table>
Using personal data files

The Personal Data File data provider allows you to access data in Microsoft Excel spreadsheets, dBASE files, and text files.

What are the benefits of using personal data files?

The main benefits of using personal data files are as follows:

• You can display corporate data next to personal data in the same report. For example, you can compare your company budget (corporate data) with your own running costs (personal data). You can obtain such a report by building a query to retrieve the corporate data, then by inserting a new table that displays data from a personal data file.

• If you have no connection to a remote database or if there is no RDBMS at your site, you can use personal data files as your only data source.

• You can use BusinessObjects reporting and analysis features to work on data that comes from other applications.

Creating a report using a personal data file

Creating a report from a personal data file is a two-stage procedure:

• Specify the personal data file that you want to use for the report. This is described under Selecting the personal data file for the report on page 95.

• Set options that depend on the type of file you selected in the first stage. For example, the options to set for a spreadsheet are different from those for a text file.
Selecting the personal data file for the report

To use the New Report Wizard to get to the personal data file containing the data you need:

2. Select an option for the report layout, then click **Next**. The Specify Data Access dialog box appears.
3. Under Others, click **Personal data files**, then click **Finish**.
4. Click **Browse** to locate the file that contains the data you want.
The Open a File to Access Personal Data dialog box appears.
When you have located the file and closed the dialog box, the path to the file appears in the Name field of the Access Personal Data dialog box.
The Format field displays the format of the file you selected.
The options in the dialog box are now specific to the file type you are working with.
If you have selected a dBASE file, no further options are available so click Run.

5. Set the options you want:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then...</th>
<th>Applies To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show the first line of the file as</td>
<td>Select <em>First row contains column names</em></td>
<td>All files</td>
</tr>
<tr>
<td>column headers in the report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create drill hierarchies and</td>
<td>Select <em>Build hierarchies and start in drill mode</em></td>
<td>All files</td>
</tr>
<tr>
<td>open the report in drill mode.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(BusinessObjects can do this only if</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the dimensions in your report have a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hierarchical structure, for example</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year, Quarter, Month</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specify the delimiter in a text file</td>
<td>Select <em>Tabulation, Space or Character.</em> (If you select Character you</td>
<td>Text Files (.asc; .prn;</td>
</tr>
<tr>
<td></td>
<td>need to enter the character that delimits the data.)</td>
<td>.txt; .csv)</td>
</tr>
<tr>
<td>Select the worksheet containing the</td>
<td>Select the worksheet from the Sheet Name list</td>
<td>Microsoft Excel (.xls)</td>
</tr>
<tr>
<td>data you want</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select data from all fields in a</td>
<td>Select <em>All Fields</em></td>
<td>Microsoft Excel (.xls)</td>
</tr>
<tr>
<td>worksheet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select data from a range of cells in</td>
<td>Type the range (for example A3:R25) in the Range Definition box</td>
<td>Microsoft Excel (.xls)</td>
</tr>
<tr>
<td>a worksheet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select data from a named range in a</td>
<td>Select the range in the Range Name list</td>
<td>Microsoft Excel (.xls)</td>
</tr>
<tr>
<td>worksheet</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Click **Run**.

The data from the personal data file appears in the report.
Using Visual Basic for Applications procedures

A VBA data provider is a powerful and flexible tool for accessing external data. Very often you will want to access automation servers through VBA to retrieve their proprietary data. VBA allows you to retrieve data from various sources: ADO, DAO, RDO, Application Object Models, EDK and low-level APIs.

To create a VBA data provider, you write a VBA procedure that takes the interface DpVBA Interface as a parameter. You can write this procedure from within the VBA environment of BusinessObjects. The DpVBAInterface is the interface to the VBA data provider Automation object which is described in detail in the BusinessObjects SDK Reference Guide.

The procedure for writing a VBA data provider is:
1. Create a connection to the data source.
2. Create a data cube.
3. Set the data cube dimensions.
4. Populate the cube with data from the data source.

Once the data cube is populated, you can generate a report based on this data in BusinessObjects.
Creating a report using a VBA data provider

To create a report using a VBA data provider:


2. Select an option for the report layout, then click **Next**.
   The Specify Data Access dialog box appears.
3. Under Others, click *Visual Basic for Applications procedures*, then click *Finish*.

   The Access Data From VBA dialog box appears.

   ![Access Data From VBA dialog box](image)

4. Select the subroutine and click **Run**.

   BusinessObjects generates the report.

**EXAMPLE**

**Accessing an Outlook inbox using VBA**

This example shows how to generate a report based on the contents of an Outlook inbox.
Referencing the Outlook object library

To do this:
1. Click **Macros** on the Tools menu, then click **Visual Basic Editor**
   The Visual Basic editor appears.
2. Click **References** on the Tools menu.
   The References dialog box appears.
3. Select the Microsoft Outlook Object Library and click **OK**.

Now you need to enter the code of the VBA data provider. To do this:
2. Select an option for the report layout, then click **Next**.
   The Specify Data Access dialog box appears.
3. Under Others, click **Visual Basic for Applications procedures**, then click **Finish**.
   The Access Data from VBA dialog box appears.
4. Type Outlook for the subroutine name, then click **Create**. The Visual Basic editor opens with a skeleton subroutine.

```
Public Sub Outlook(dpInterface As DpVBAInterface)
    Dim olkApp As Outlook.Application
    Dim nspsNameSpace As NameSpace
    Dim objInboxFolder As Object
    Dim objMail As Object
    Dim oCube As DpVBACube
    Dim sName(10) As String
    Dim oColumns As DpVBAColumns

    Set olkApp = CreateObject("Outlook.Application")
    Set nspsNameSpace = olkApp.GetNamespace("MAPI")
    Set objInboxFolder = nspsNameSpace.GetDefaultFolder(olFolderInbox)

    dpInterface.UserString(1) = "User String for Outlook Data Provider"

    Set oCube = dpInterface.DpVBACubes.Item(1)
    Set oColumns = oCube.DpVBAColumns
    oColumns.SetNbColumns (7)

    Dim oCol As DpVBAColumn
    Dim row As Integer
    Dim col As Integer
    Dim sColName(?) As String
```

5. Type the following code:

```
Public Sub Outlook(dpInterface As DpVBAInterface)
    Dim olkApp As Outlook.Application
    Dim nspsNameSpace As NameSpace
    Dim objInboxFolder As Object
    Dim objMail As Object
    Dim oCube As DpVBACube
    Dim sName(10) As String
    Dim oColumns As DpVBAColumns

    Set olkApp = CreateObject("Outlook.Application")
    Set nspsNameSpace = olkApp.GetNamespace("MAPI")
    Set objInboxFolder = nspsNameSpace.GetDefaultFolder(olFolderInbox)

    dpInterface.UserString(1) = "User String for Outlook Data Provider"

    Set oCube = dpInterface.DpVBACubes.Item(1)
    Set oColumns = oCube.DpVBAColumns
    oColumns.SetNbColumns (7)

    Dim oCol As DpVBAColumn
    Dim row As Integer
    Dim col As Integer
    Dim sColName(?) As String
```
sColName(1) = "From"
sColName(2) = "To"
sColName(3) = "Cc"
sColName(4) = "Subject"
sColName(5) = "Size"
sColName(6) = "Created"
sColName(7) = "Received"

Dim oColData(7) As Variant

'Loop through 10 rows in the inbox and assign values to the 7 columns in each row.
For row = 1 To 10
    'Get the row's data.
    Set objMail = objInboxFolder.Items.Item(row)
    oColData(1) = objMail.SenderName
    oColData(2) = objMail.To
    oColData(3) = objMail.CC
    oColData(4) = objMail.Subject
    oColData(5) = objMail.Size
    oColData(6) = objMail.CreationTime
    oColData(7) = objMail.ReceivedTime
    'Loop through the columns.
    For col = 1 To 7
        Set oCol = oColumns.Item(col)
        'Set the column name and data type on the first iteration.
        If row = 1 Then
            oCol.Name = sColName(col)
            'First 5 columns are strings, last 2 are dates.
            If col < 6 Then
                oCol.Type = boCharacterObject
            Else
                oCol.Type = boDateObject
            End If
        End If
    Next col
    oCol.Qualification = boDimension
    oCol.Item(row) = oColData(col)
Next row
Next row
dpInterface.CheckDataIntegrity (boCheckAll)

6. Click **Compile** on the Debug menu to compile the project.
7. Click **Close and Return to BusinessObjects** on the File menu to return to
8. Click **Run**.
BusinessObjects generates the report.

<table>
<thead>
<tr>
<th>Form</th>
<th>To</th>
<th>Cc</th>
<th>Subject</th>
<th>Store</th>
<th>Created</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike Marcus</td>
<td>Brian Muskat, PA</td>
<td>RE: Anyone for a chocolate</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Mike Marcus</td>
<td>Brian Muskat, PA</td>
<td>RE: Anyone for a chocolate</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Brian Muskat</td>
<td>Richard Napoli</td>
<td>RE: Anyone for a chocolate</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Elizabeth VanEnder</td>
<td>Phil Naughton</td>
<td>Fix: h:template tips</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Elizabeth VanEnder</td>
<td>Phil Naughton</td>
<td>Fix: h:template tips</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Jason Shultewort</td>
<td>BPS - DGC</td>
<td>bpe documentation</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Jason Shultewort</td>
<td>Dan Kowarz</td>
<td>bpe: Documentations R1.2902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>lunch Meeting</td>
<td>Stu Miskett, Phil</td>
<td>Blind 22 available on TOSAC</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Web Thomson</td>
<td>Terry Austin, St</td>
<td>Looking for examples of VSO</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
<tr>
<td>Stuart Miskett</td>
<td>Mike Marcus</td>
<td>RE: Anyone for a chocolate</td>
<td>12902</td>
<td>7/06/02</td>
<td>7/06/02</td>
<td></td>
</tr>
</tbody>
</table>
Using XML files

This section describes how to use XML as a BusinessObjects data provider.

What is XML?

XML is a text-based data format that structures data in elements or tags. XML files are similar to the HTML files used to build pages on the World Wide Web. The principal difference is that, whereas the set of HTML elements is limited to those used to describe the structure of a Web page, an XML file can contain any elements, depending on its application.

Here is an example of an XML file containing data from the Island Resorts Marketing database:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Resorts>
  <Resort>
    <Country>France</Country>
    <ResortName>French Riviera</ResortName>
    <ServiceLine>Accomodation</ServiceLine>
    <Revenue>563250</Revenue>
  </Resort>
  <Resort>
    <Country>France</Country>
    <ResortName>French Riviera</ResortName>
    <ServiceLine>Food and Drinks</ServiceLine>
    <Revenue>107400</Revenue>
  </Resort>
  <Resort>
    <Country>France</Country>
    <ResortName>French Riviera</ResortName>
    <ServiceLine>Recreation</ServiceLine>
    <Revenue>164770</Revenue>
  </Resort>
  <Resort>
    <Country>US</Country>
    <ResortName>Bahamas Beach</ResortName>
    <ServiceLine>Accomodation</ServiceLine>
    <Revenue>67364</Revenue>
  </Resort>
  <Resort>
    <Country>US</Country>
    <ResortName>Bahamas Beach</ResortName>
    <ServiceLine>Food and Drinks</ServiceLine>
    <Revenue>169680</Revenue>
  </Resort>
</Resorts>
```
XML files can store many different types of data. This manual could be stored as XML, as could the data in a relational database. Database-like XML, such as the bookstore data above, is the only XML that is meaningful as a datasource for BusinessObjects.

Creating a report using an XML file

Creating an XML-based report involves two steps:

- building an XML filter
- building the report

**Building the XML filter**

When you build an XML filter you choose the elements in the XML file that you want to be available for inclusion in your report. To build a filter:

1. Click **New Report Wizard**.
2. Select a layout option, then click **Begin**.
3. Click **Others** and select *XML Data Provider* from the Others box, then click **Next**.
4. In the Select an XML Filter step, click **New**.

![Create XML Filter dialog box](image1)

The Create XML Filter dialog box appears.

5. Click **Load XML**, then use the Open dialog to navigate to and select the XML data.

![Load XML and Open dialog](image2)
file.

The structure of the XML file appears in the Structure box.

To reload the XML, click **Refresh**. To display the values of an element, select the element in the structure box and click **Display Sample Values**.
6. Select the elements to be included in the Structure box.
   Some XML elements do not contain data; they act as a container for other elements. "Resorts" is an example of such an element in this file. It is not meaningful to include such elements in a report. If you do, their values appear as `<element_name> + '_' + number`. If you include the Resorts element in a BusinessObjects report, it appears as Resorts_00001, Resorts_000002.

7. Edit the object names, qualifications and data types in the Variables box.

8. Click Save.
   The Save XML Filter As dialog box appears.
9. Type a filter name in the New Filter Name text box and click OK. The filter appears in the list of XML filters.

Building the report
You build reports based on XML using the XML filters that you have defined. To build a report:
2. Select a layout option, then click Begin.
3. Click Others and select \textit{XML data provider} from the Others box, then click \textit{Begin}.
4. Select the filter in the list of filters and click **Finish**. The XML Query Panel appears, showing the elements you selected when you built the filter available for inclusion in the report.
5. Double-click the elements that you want to include in the report.

![XML Query Panel](image)

6. Click **Run**. BusinessObjects generates the report.

<table>
<thead>
<tr>
<th>Country</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>$265,428.00</td>
</tr>
<tr>
<td>US</td>
<td>$265,144.00</td>
</tr>
</tbody>
</table>
Setting the location of XML files

When you create an XML filter, BusinessObjects creates a file with the filter definition. You can tell BusinessObjects where to store XML filter files and XML files. To do this:

1. Click **Options** on the Tools menu.
   The Options dialog box appears.
2. Select the **File Locations** tab.

3. Select **XML Sources** to change the location of XML source files or **XML Filters** to change the location of XML filter files.
4. Click **Change**.
   The Browse for Folder dialog box appears.
5. Use the Browse for Folder dialog box to select the folder where you want the files to be stored.
6. Click **OK** to close the Browse for Folder dialog box.
7. Click **OK** to close the Options dialog box.
Using OLAP cubes

OLAP servers store data in cubes that contain dimensions and measures. You can access cubes through BusinessObjects by installing the OLAP access packs. There are several packs providing access to several OLAP servers. For further details on OLAP access using BusinessObjects, see the BusinessObjects OLAP documentation.
Using Web Connect

Web Connect allows you to draw data from pages on the World Wide Web to create BusinessObjects reports. For instructions on its use, see The Web Connect Data Provider on page 119.
The Web Connect Data Provider
This chapter discusses Web Connect, a data provider that allows you to retrieve data from pages on the World Wide Web.
Querying the Internet

Much of the success of Business Objects has been based on the capability of its products to retrieve data from relational and OLAP databases using its own patented semantic layer. BusinessObjects users can run query, reporting and analysis on the information stored in their data warehouses, and format the extracted data into sophisticated reports.

The next logical step in this process is the ability to run queries on data which is not already stored in databases, but which is available globally in HTML format on the World Wide Web. WEB CONNECT provides this capability.

Internet queries and query catalogs

The Internet queries that you create using Web Connect are listed in query catalogs. A query catalog points to a list of predefined queries. For example, you might have a query catalog called Competitor that points to a number of predefined queries that pull data from competitor sites. Both catalogs and queries are saved in XML (eXtensible Markup Language) format.

**NOTE**

The catalogs themselves are simply pointers to the query files. It is the queries themselves that contain all the necessary information for retrieving specific data from the referenced web page(s), as well as the display options and any prompts that may have been included.

HTML and XML file formats

WEB CONNECT uses XML technology. XML (eXtensible Markup Language) is a simplified subset of the Standard Generalized Markup Language (SGML). XML provides:

- a file format for representing data
- a schema for describing data structure
- a mechanism for extending and annotating HTML with semantic information

Web Connect retrieves HTML data and serializes the definition of the queries in XML format. These XML files contain the web metadata and Internet query definitions.
Accessing Data and Data Analysis

Although the long term distribution media of data over the Internet is likely to be in XML format, it seems that more than 99% of the data available on the web today is in HTML format. Web Connect allows you to query this HTML data, providing access to the vast majority of the Internet.

Assembling your query data

You build your Internet queries by dragging and dropping selected web data from your web browser to the grid area at the bottom of the Internet Query Panel.
Selecting corporate and Business Objects Queries

You can retrieve existing Internet queries from your own file systems, or from the web. Business Objects also provides access to a web site, from which you can access predefined Internet queries to get you started.

Once you have selected a query in the Internet Query Wizard and clicked on Finish, the query is run and displayed in a BusinessObjects report.

You can also create your own predefined queries.

Creating Internet queries

By selecting the New Query option, you open the Internet Query Panel. This panel has two sections:

- The upper section contains a web browser (by default, your home page is displayed).
- The lower section is where you create your Internet query.

You can drag and drop any tagged HTML cell into the lower window. This data may include text, table data, and hyperlinks.

By dragging the cell onto the gray bar it will become a column header. Dragging to the white page, the HTML cell will become a table cell. When you highlight a section of a web page, Web Connect will automatically select the smallest complete tagged cell that encompasses your selection.
You can also select cells from any number of web pages and combine them into the same table in the lower section of the window. For example, you might want to pull your competitors’ stock quotes (from a Stock Market financial web site) and their quarterly revenue (from their corporate web site) into the same table.

Furthermore, you can build reports from both Internet data and other data sources. For example, you could combine the Euro to Dollar exchange rate from the Internet with sales figures from a universe to get European sales in Dollar equivalent.

**Adding prompts to queries**

An Internet query prompt allows you to pass parameters to web pages. For example, on a historical quotes financial page you could use a prompt to define the ticker symbol or the historical date range. So when you run or refresh that Internet query you will get a prompt that asks you to enter the ticker symbol (or select a value from a list) or the date range.

There are two ways to create prompts within a web query:

- One or several monovalued prompts: A maximum of nine monovalued prompts can be entered within each URL.
- One multivalued prompt and, optionally, up to eight monovalued prompts.

**Generating reports**

Once you have selected the data you want in your Internet query, you click Run to generate a BusinessObjects report. You can format this report as with regular BusinessObjects reports. If you choose to edit the data provider, you are returned to the Internet Query Panel.

**NOTE**

You can slice and dice in an Internet query report, and you can save your Internet query for personal or for corporate use. However, you cannot directly drill down into an Internet query report within Web Connect. In order to drill into data in an Internet query, you must first manually define the object hierarchy in BusinessObjects. For information on this procedure, refer to Drilling using custom hierarchies on page 262.

**Refreshing reports**

When you refresh a report based on an Internet query, BusinessObjects will reconnect to the web page(s) and bring back the most current data. If the structure of the web page has changed, you will be notified by an error message.
displayed in the corresponding cell. The message will appear as #ERROR or #EMPTY, depending on the option selected in the Options dialog box in the Internet Query Panel (for more information on configuring the Internet Query Panel, refer to Setting individual Internet query options on page 172). When this error appears, you must modify the Internet query data provider to correspond to the new web page structure.
Using Web Connect

For most users, the recommended procedure when using the Web Connect data provider is to use existing corporate Internet queries (maintained by your IT staff). To build reports, you then simply select the query you require, choose options from any of the displayed prompts, and run the query.

Selecting the Web Connect data provider

To select the Web Connect data provider

1. Start BusinessObjects and log in.

The New Report wizard appears
2. Click *Generate a standard report*, then click **Begin**. The Specify Data Access window appears.

3. Select Others, click on the list of data providers, choose Web Connect from the list, then click **Finish**.
The New Internet Query wizard opens.

This wizard helps you select the way you want to access Internet data:

- Click *New Query* to create your own query.
- Click *Corporate Query* to retrieve and use an existing query from a predefined Corporate catalog.
- Click *Business Objects Query* to retrieve and use an existing qualified query from the Business Objects web site.
Running a Predefined Business Objects Query

To make a quick start, run a predefined Business Objects query.

1. Click *Business Objects Query*, then click **Next**.

The list of standard BusinessObjects queries appears.

These queries may be stored anywhere on the external Web or on the intranet file server. The queries themselves are listed in catalog subdirectories. The catalogs are arranged hierarchically in a tree structure that can be expanded and collapsed.

Both catalogs and queries are saved in XML format. When you double-click on an XML file, it is automatically opened in your web browser.
The catalogs are simply pointers to the query files. It is the queries that contain all the necessary information for retrieving specific data from the referenced web page(s), as well as the display options and any prompts that may have been included.

2. Select a query.
   A description of the query appears under Help on the selected query.

3. Click Finish.
   A progress bar appears while Web Connect makes a connection to the associated web site(s) and retrieves the query information.

   The retrieved query information is displayed in the report window like any BusinessObjects report.
You can now work with this report as you would with any other BusinessObjects report.

**Running a predefined corporate query**

In order to run a predefined query, you need to know the location on the network where predefined queries are saved.

To run a predefined query:

1. Start BusinessObjects and create a new report.
   The Specify Data Access window is displayed.

2. Click *Others*, click on the list of data providers, choose Web Connect from the list, then click *Finish*.
   The New Internet Query wizard appears.
3. Click *Corporate Query*, type the path to the location of the queries, then click *Next*.

A list appears from which you can select a query.

4. Select a query and run it.

You can now work with the resulting report as you would with any other BusinessObjects report.
Creating a new Web Connect query

This section shows you how to create your own Internet queries, and how to add content to them. You will also learn how to add prompts. Prompts allow you to specify particular report options when you eventually generate your reports.

To create a new Web Connect query:

1. Follow the steps in Selecting the Web Connect data provider on page 126 up to and including step 4.

The New Internet Query wizard appears.
2. Click New Query to create your own query, then click Next.
The Internet Query Panel is displayed.

The Internet Query Panel comprises:

- **A scrollable browser window** - When you create or edit a query, Internet Explorer is automatically started, and your currently configured home page is displayed by default. However, you can select pages from any web address. Once the page is displayed, you can select data from the page that you want to include in your query.
- **The query grid** - Where you will build your query. This comprises a shaded area for header information, and a blank area for the body data.
- **A toolbar** - From where you can quickly access the more commonly used features of the Internet Query Panel (see below).
### The Internet Query Panel Toolbar

The toolbar gives you access to the following features:

- **Back**: Displays the previous web page in the browser window.
- **Forward**: Displays the next web page in the browser window.
- **Stop**: Stops all ongoing updates in the browser window.
- **Refresh**: Updates the data displayed in the browser window.
- **Home**: Displays your Home page in the browser window.
- **Import**: Imports an existing query.
- **Export**: Allows you to save the current query.
- **Prompts**: Allows you to add or edit prompts in the current query.
- **Authentication**: Allows you to set up authentication data for access to secured web sites.
- **Options**: Allows you to specify Decimal Separator and Error Management options.
- **Undo**: Undoes the last action in the Query grid. You can undo a succession of actions by repeatedly clicking the Undo button.
- **Redo**: Redoes the last action in the Query grid. You can redo a succession of actions by repeatedly clicking the Redo button.
- **Autoqualify**: Automatically qualifies all data in the Query grid. This action resets any manual qualifications you may have made.
- **Preferences**: Allows you to set a page download timeout and a default size for the Internet Query Panel.

**NOTE**

If you want to change the page that is automatically displayed as your home page in Internet Explorer, click Internet Options in the Tools menu in Internet Explorer, and enter an alternate URL in the home page address field.
Adding content to a query

When you create a new query, you are presented the Internet Query Panel. This is a split-screen window. In the upper section of the screen, you will see your standard Internet browser. In the lower section of the screen, you will see the Query Grid.

Initially, the browser displays your default home page, and the Query Grid is empty. You add content to the Query Grid by:
• browsing to the information you want in the browser
• selecting specific cells of web data, and dragging it down to the Internet Query Panel

Selected web data can be an individual cell or a collection of cells from an HTML table. If dropped at the top of the grid area, within the first grayed row, the top row of the selected web data will correspond to header information.

If dropped below the first row of the grid area, the selected web data will correspond to actual data to be retrieved. There are therefore 2 types of cells to be found within the grid area:
• Header cells
• Data cells

When you drag data from a web page and drop it into the query grid, Web Connect interprets the data by reading the HTML tags, then handles the enclosed data according to its structure. This data structure classification is based on the Microsoft Internet Explorer HTML Document Object Model.

When you select an area within a web page, Web Connect searches the selected area for row and header HTML tags (<TR>, <TH>, <TD>). Web Connect then selects the closest parent HTML tag for this collection of table tags. Depending on your initial selection it will select either an individual cell, a row or a table of data. Web Connect will therefore assign one cell of the query grid to each selected table cell. For example, if you select two adjacent cells on an HTML table, Web Connect interprets them on the lowest level as elements of a row, and will therefore assign a row of the query grid to the row of data.

Web Connect also allows you to add several cells of HTML data or even an entire web page to a single cell in the query grid. This method overrides the classification according to the HTML Document Object Model and enables you to place multiple data cells or multiple tables into a single cell on the query grid. In order to place data into a single cell, select the appropriate cells in the Browser window on the Internet Query panel. While holding down the Shift key, drag and drop the selected data into the corresponding cell on the query grid. The selected web data will be dropped into the selected cell. The resulting report will display
the web data in a single cell. This method is primarily useful if you ultimately display this data as HTML, by right clicking the appropriate cell in the query grid and then selecting HTML from the drop down menu (for information on creating a BusinessObjects report in HTML, refer to, “Creating a BusinessObjects report in HTML” on page 144).

**NOTE**

URLs used to create your queries are saved in the registry so you can retrieve them within the Internet Query Panel combo box.
Creating an Internet query

In the following example you create a query by extracting currency conversion data from a web site. Once the query has been successfully generated, you will be able to create a simple BusinessObjects report by clicking **Run**.

To create an Internet query:

1. Enter the URL of the site you wish to query in the Address box. (In this example the site is a site that provides currency conversion data.)

2. Select the two currencies for which you want the conversion rate: US Dollars
and Euros.
3. Click **Convert** in the browser window to display the current exchange rate.
4. Highlight the text “US Dollars”, and drag it to the header area of the Internet Query Panel. This creates a small table of cells, 3 columns wide by two rows deep.

![Internet Query Panel](image)

Note that the top row of the new table in the Internet Query Panel is shaded. This is the header row. The non-shaded rows are the body rows. For each query that you generate, you will need to differentiate between header data and body data.
5. In the browser window, highlight the “1” below US Dollars and drag it to the first body cell in the Internet Query Panel.

Note that the first column of the table in the Internet Query Panel is also shaded. This is used by Web Connect to number the rows.
6. Repeat the above procedure to drag the Euro data into the Internet Query Panel.

The result appears as follows:

As you add values, Web Connect always adds an extra column to the right of your data in the grid, and an extra row below. Only the drag and drop and paste options are enabled on these added cells. All other options on the right-click menu are grayed out.

If you make a mistake while copying data to the cells in the Internet Query Panel, right-click on the cell and click **Clear Selection** on the shortcut menu.
7. Click **Run** to run the query. BusinessObjects processes the query and displays the results as a standard BusinessObjects report.

Creating a BusinessObjects report in HTML

Web Connect allows you to drag any component of a web page and drop it into a cell on the query grid. This enables you to display the retrieved data as HTML, either in a static BusinessObjects report viewed through a web browser, or as a refreshable BusinessObjects report viewed in InfoView. This feature allows you to pull images and graphs from web sites and view them as HTML in a BusinessObjects environment.

In order to view a document as HTML, you must tell Web Connect to interpret the assigned data as HTML. To do this:
1. Within a new report in the Internet Query Panel, create a new query and enter the URL of the web site from which you want to extract your data.

2. Construct your query in the query grid (for information on creating Internet queries, see Creating a new Web Connect query on page 133).
3. Select your data and drag it into a single cell in the query grid by dropping the selected data while holding down the shift key. The selected data appears in a single cell in the query grid.

4. Optionally, right click on the active column and click Rename Header on the shortcut menu to enter a name in the header row.

5. Right click on the selected cell and click HTML on the shortcut menu.

6. Click Run in the Internet Query panel to run the query. The resulting BusinessObjects report appears to be empty. You must now
format the individual cells of the report in order to view the HTML data.

7. To format a cell, right-click it then click **Format Cell** on the shortcut menu. The Format Cell window appears.

![Format Cell window]

8. Click **Read as HTML** then Click **OK**.

9. Repeat this step for all cells containing data that you want to appear as HTML.

10. Save your BusinessObjects report.

11. Click **Publish to Corporate Documents** on the **File** menu.

   This exports your document so that you can view and refresh your data in InfoView. Alternatively, you can click **Save as HTML** on the **File** menu and then view your report in a Web browser.

**NOTE**

Publishing your document to Corporate Documents allows you to view and refresh your data in InfoView. Saving your report as HTML will allow you to view your document in a Web browser, but you will not be able to refresh your data. Your document will be static.

12. Open your report in InfoView or in your Web browser. The resulting report will display the data you extracted from web sites using Web Connect.
NOTE

Creating reports in HTML using Web Connect allows you to display web components such as graphics in your reports, as demonstrated with the charts in the above report. However, only components referenced by absolute links may be viewed. Graphics linked by relative reference cannot be viewed.
Saving Reports

You can save any of the reports you generate as follows.

1. Click **Save** on the **File** menu.

2. Select a location, enter a name for the report, then click **Save**.
   The report is saved with a .rep extension.
   You can click on the .rep file to open the report in BusinessObjects and refresh it.

**NOTE**

When you save a report, you are saving the query results and the definition of the query itself. For information on saving queries individually, refer to Saving Web Connect queries on page 167.
Internet Query Grid Context Menu Options

When you right-click anywhere in the Internet Query Grid, you will see a context menu with several options. These options vary depending on the type of cell that you select.

**Context menu options in header cells**

In addition to the standard Cut, Copy, Clear and Paste options, the context menu also displays the following options when you right-click on one of the header cells.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename header</td>
<td>Allows you to enter another text string in the selected cell.</td>
</tr>
<tr>
<td>Delete column</td>
<td>Deletes the selected column.</td>
</tr>
<tr>
<td>Dimension</td>
<td>Allows you to qualify the selected data as a dimension</td>
</tr>
<tr>
<td>Detail</td>
<td>Allows you to qualify the selected data as a detail</td>
</tr>
<tr>
<td>Measure</td>
<td>Allows you to qualify the selected data as a measure</td>
</tr>
<tr>
<td>Character</td>
<td>Allows you to set the selected data type as a character</td>
</tr>
<tr>
<td>Date</td>
<td>Allows you to set the selected data type as a date</td>
</tr>
<tr>
<td>Numeric</td>
<td>Allows you to set the selected data type as numeric</td>
</tr>
</tbody>
</table>

**NOTE**

You can also delete the entire grid by right-clicking in the top lefthand cell of the table.
Context menu options in body cells

The context menu displays the following options when you right-click on one of the body cells:

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Text</td>
<td>Retrieves the dynamic text corresponding to the selected cell. The data is directly linked to the corresponding data in the browser window, and will change accordingly. This is the default property of a cell.</td>
</tr>
<tr>
<td>Hyperlink</td>
<td>Retrieves the hyperlink corresponding to the selected cell.</td>
</tr>
<tr>
<td>HTML</td>
<td>Retrieves the HTML source corresponding to the selected cell.</td>
</tr>
<tr>
<td>Static Text</td>
<td>Converts the data in the selected cell(s) to static text. Static text will no longer have any link to the corresponding data in the browser window. It is therefore processed as a constant.</td>
</tr>
<tr>
<td>Edit Cell</td>
<td>Allows you to edit the text that is displayed in the cell. This automatically converts it to static text.</td>
</tr>
</tbody>
</table>

Applying commands to entire rows

The context menu displays the following options when you right-click on any of the cells in the first column of the Query grid:

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Cuts the data from the selected row, and places it in the clipboard.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the data from the selected row, and places it in the clipboard.</td>
</tr>
<tr>
<td>Paste</td>
<td>Pastes the data from the clipboard into the currently selected row. This will overwrite any data in the selected row.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears the data from the selected row.</td>
</tr>
<tr>
<td>Delete Row</td>
<td>Completely deletes the currently selected row from the Query grid.</td>
</tr>
</tbody>
</table>
### Applying commands to the entire Query grid

The context menu displays the following options when you right-click in the top left-hand cell of the Query grid:

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Cuts the data from all rows and places it in the clipboard</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the data from all rows, and places it in the clipboard.</td>
</tr>
<tr>
<td>Paste</td>
<td>Pastes the data from the clipboard into the Query grid. This will overwrite any data in the Query grid.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears the data from all rows.</td>
</tr>
<tr>
<td>Reset</td>
<td>Completely clears the Query grid.</td>
</tr>
</tbody>
</table>
Using Prompts with Web Connect Queries

You can add prompts Web Connect queries. Prompts allows you to restrict your report to specific data.

What is a prompt?

A prompt is an extra feature that you can add to your queries in order to refine the type and amount of information that is displayed in the report each time the query is run.

When you run a query or refresh a report that contains prompts, a small dialog box is displayed. This dialog box prompts you to make a selection from a series of items, and will determine the content of the report.

You can create a prompt by opening a query and clicking the Prompts button. This opens a dialog which allows you specify exactly how the prompt will work. Here, you specify the type of prompt(s) you will use:

• Up to nine monovalued prompts.
• One multivalued prompt plus up to eight monovalued prompts.

Monovalued prompts

A monovalued prompt is a prompt which allows you to select one -- and only one -- item from a list of items. For example, the prompt illustrated above shows that 3 items have been selected, AUD;GBP;ITL. However, if this is a monovalued prompt, then only the first item selected (AUD) will be used when the query is run.

Multivalued prompts

A multivalued prompt is a prompt which allows you to select several items from a list of items. Using the same example, if all 3 items are selected (AUD;GBP;ITL), then the data relating to these three prompts is retrieved when the query is run.

Using Prompts with Web Connect Queries
Creating a prompt

This section explains how to create a single multivalued prompt using the previous example.

1. Open the report in BusinessObjects.
2. Click **Edit Data Provider** on the **Data** menu.

   The Internet Query Panel appears.
3. Click Prompts.
   The Internet Query Prompts dialog box appears.
   The URL used within the current Internet query is automatically retrieved and
displayed in the URL list box.

   ![Prompts Dialog Box]

4. Click on the URL.
   The URL appears in the list of URLs beneath the New button and the New
button becomes active.
5. Click **New**.

The Prompt Properties dialog box appears.

[Image: Prompt Properties dialog box]

The Web Connect Data Provider
6. Click **Monovalued**
   - Monovalued prompts allow the user to select one value from the list of values related to the prompt.
   - Multivalued prompts can reference multiple data cells. This option is grayed out if you have already created a multivalued prompt on the selected query: only one multivalued prompt is allowed per Internet query.

7. Type “Currency” in the Prompt Name box.
8. Click **Insert** to enter additional items in the List of Values field. A “new item” is added in the box.

![Prompt Properties dialog box]

9. Rename the item “AUD” (Australian Dollars), and press Enter.
10. Click **Insert** again, and repeat this procedure to add two more items: GBP (Pounds Sterling) and ITL (Italian Lire).
11. Click **Create an associated column** in order to display the prompt values in a separate column when you run or refresh the query.
   In this case, a column name is automatically entered, corresponding to the
prompt name you entered.

The Web Connect Data Provider
12. Click OK.

The Internet Query Prompts dialog box is re-displayed with the prompt added.
13. To associate the newly created prompt with the URL currency field, select the section of the URL you want to replace with the prompt (in this case, “EUR” at the end of the URL). The selected text area is displayed at the bottom right-hand side of the dialog box.
14. Click **Activate**.

You will see the prompt change. You will also notice that the prompt is added to the end of the URL, in this case `@Prompt("Currency")`.

![Prompt dialog box](image)

**NOTE**

The format for a prompt in a URL is:

```
http://...@prompt('message').
```

You can enter several prompts within the same URL, but only one multivalued prompt can be entered per Internet Query.

15. Click **OK** to finish.

The Internet Query Panel is displayed again. Because you have replaced the EUR in the URL with the prompt, when you next run the query, the associated column of the report will display the exchange rates of one or more of the 3 currencies you specified in the prompt. As a result you need to change the heading of this column in the Internet Query Grid. Should you decide to deactivate the new prompt by clicking **Deactivate**, the query runs using the original initial value (in this case, EUR).
and the URL returns to its initial value (that is, with EUR at the end.).

16. Right-click on the Euro header click **Rename Header** on the shortcut menu. Enter “Exchange Rate” in the Rename Cell dialog box then click **OK**.

![Rename Cell dialog box](image)

- **Running a query with a prompt**
  1. Run the query by clicking **Run**. The Enter or Select Values dialog box appears.

![Enter or Select Values dialog box](image)

  2. Click **Values**, select one or more values, then click **OK**.

![List of Values dialog box](image)

The Enter or Select Values dialog box shows a list of the values you
selected.

3. Click **OK** again.
The Internet Query in Progress dialog box appears while the query executes, and then the report is displayed.

<table>
<thead>
<tr>
<th>US Dollar</th>
<th>Currency</th>
<th>Exchange Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>AUS</td>
<td>1.05</td>
</tr>
<tr>
<td>1.00</td>
<td>GBP</td>
<td>0.63</td>
</tr>
<tr>
<td>1.00</td>
<td>ITL</td>
<td>1899.10</td>
</tr>
</tbody>
</table>
Managing Queries

This section explains how to open, edit, and save existing Internet queries, and how to set your Internet query preferences.

Opening existing queries

To open an existing query.

1. Click **Import** on the Internet Query Panel

   The Open dialog box appears.

This dialog box allows you to open and edit an existing query. This is an important function in dealing with web data, given that the structure of web pages may change frequently. If the structure of a web page changes, the cells of the corresponding report which contain this data will display an error message (for more information on data-source error messages, refer to Conversion errors on page 174). If you select the Save query location option, Web Connect points to the location of the selected query by default the next time you click Open Internet Query.
NOTE

The Open dialog box also allows you to reload a predefined query if its definition has changed and the new definition has been uploaded on the web by IT. This is an effective way to resolve errors resulting from changes in the structure of source web pages.

2. To browse for an existing query, click **Browse**.
   The Browse dialog box appears.
3. Locate the query you want to import, then click **Open**.

![Image of Internet Query Panel]

4. Add, delete, or change any of the cells that are referenced in the query, as required.

5. Edit the prompt fields, as necessary.

6. If you wish, save the query with a new filename.

7. Click **Run**.

**Editing a Web Connect data provider**

You can edit the source query information for a report as follows.

1. In BusinessObjects, open your created document.
   
   You can also simply double-click the BusinessObjects document (.rep file) to open it.

2. Click **Edit Data Provider** on the **Data** menu.

   The Internet Query Panel appears, showing both the referenced web page and the selected data cells in tabular format.

3. To view the source of each data cell, right-click the cell in the table, then click **View Source**.
**Select In Browser** on the shortcut menu, or double-click the selected cell.

The associated information is highlighted in the browser.

---

**Saving Web Connect queries**

By saving your queries to a shared network location, you can make them available for other Web Connect users. This location can then be used as a central storage point for all such predefined queries.

1. Click **Export** in the Internet Query Panel.
   - The Save As dialog box appears.
2. Choose a location, enter a name for the file, and click **Save**.
   - The query is automatically saved as an XML file.
Setting preferences

You can modify your Internet Query Panel preferences to set a page download timeout and a default size for the Internet Query Panel.

1. Click **Preferences** in the Internet Query Panel.
   The Preferences dialog box appears.

![Preferences dialog box](image)

2. Choose one of the available timeout options. This is the maximum amount of time, in seconds, that Web Connect will use when attempting to display a web page in the browser window.
   You can select an incremental value between 5 and 180 seconds, or enter the value of your choice.

3. Choose a default size for the Internet Query Panel. Your options are:
   - X-Small
   - Small
   - Medium (default)
   - Large
   - X-Large (full screen)

   **NOTE**
   These sizes will always be proportional to your screen resolution.

4. Click **OK**.

Refreshing documents

When you refresh a report that contains prompts, a dialog box appears with the different prompt names.
When refreshing a report without prompts, the latest web data will be automatically retrieved based on the information entered in your Internet query.

The difference between the monovalued and multivalued prompt options appears when you try to refresh the Internet Data Provider. For monovalued prompts, you enter only one value per prompt, while for multivalued prompts, you can enter several values.

► Refreshing in Broadcast Agent

You can also use Web Connect with Broadcast Agent. When refreshing a document in Broadcast Agent, the latest prompt values will be used. Broadcast Agent does not need access to the XML files, as the query definition is stored within the data provider in the BusinessObjects document.

► Refreshing in InfoView

When refreshing a document in INFOVIEW, the regular INFOVIEW prompt dialog is displayed.

Extracting data from multiple-frame pages

Web Connect can extract data from single-frame web pages only. If the web page you want to uses multiple frames, you need to:

1. Select the multiple-frame page in the web browser area of the Internet Query Panel.
2. Right-click in the area of the page which contains the data you want to extract, then select **Properties** on the shortcut menu.

   The Properties dialog box appears.

   ![Properties dialog box]

   Note the URL address. This is the specific address of the frame that contains the data you want to access.

3. Highlight the address and press CTRL-C to copy it to the clipboard.

4. Click **Cancel** to quit the Properties dialog box.

   The multiple-frame web page is displayed again in the Internet Query Panel.

5. Highlight the URL address in the browser area, press CTRL-V to insert the specific frame address you copied, and press Enter.

   The browser now displays only the single-frame data.

6. Extract the cells you require from the frame for your report.

**NOTE**

You cannot apply this workaround to some multiple-frame pages, such as those containing frames that reference other frames.
Autoqualifying cells

When you create and run your Internet queries, the data is automatically qualified by Web Connect. This qualification is actually performed when you click Run.

However, as you have seen in Internet Query Grid Context Menu Options on page 150, you can manually qualify any of the data cells in the Query grid. If you do this, then whenever you run the query, Web Connect will always use your manual data qualifications.

To override any manual qualifications you may have made, you must autoqualify the data in the Query grid. This resets all manual qualifications, forcing autoqualification on previously qualified columns.

1. Open or create your Internet query.
2. Click Autoqualify on the Internet Query Panel

Parsing cells involves retrieving their content and checking it for possible conversion to numeric data. A numeric cell can only contain numbers and two additional character types ([ ] [ ] and [ ] for thousands separator and [ ] or[,] for decimal separator. [+ ] and [- ] are also accepted in front of the number).

For example, the following strings can be converted to numeric:

- 1 000 432
- 1.234.345,23
- 1.43
- 1’324.23

NOTE

A column can only be autoqualified to a Dimension/Character or Measure/ Numeric.

When there is an ambiguity (1.123 can be either 1123 or 1[decimal separator]123) the decimal separator entered in the Options dialog is used. If one cell fails to be converted to Measure/Numeric, the whole column is converted to Dimension/Character.

NOTE

The autoqualify button forces autoqualification for all columns, whether previously qualified or not.
Setting individual Internet query options

You can set the following options specific to each Internet query you create/edit.

- **Separator options**: Allow you to specify how Web Connect interprets the different types of separators used in numeric strings. The option you select here will only set a rule for interpreting ambiguous data when importing cells into the Query grid, as described in the previous section.

- **Error management options**: Allow you to specify the values that Web Connect will enter in the microcube on which the report is generated if errors occur.

#### Separator options

When you drag numeric data to the Query grid, Web Connect interprets any separators it finds according to a predefined algorithm.

For example:

<table>
<thead>
<tr>
<th>Numeric format</th>
<th>...is by default interpreted as</th>
</tr>
</thead>
<tbody>
<tr>
<td>93.15</td>
<td>93.15 (decimal separator)</td>
</tr>
<tr>
<td>94,15</td>
<td>94.15 (decimal separator)</td>
</tr>
<tr>
<td>940,151.13</td>
<td>940,151.13 (thousands separator, followed by decimal separator)</td>
</tr>
<tr>
<td>941.152,14</td>
<td>941,152.14 (thousands separator, followed by decimal separator)</td>
</tr>
</tbody>
</table>

However, a value such as 94,153 could be interpreted with either a dot separator or a comma separator. In this case, Web Connect will then adopt the rule you have specified in your options.
To specify the separator:

1. Click **Options** in the Internet Query Panel
   
   The Options dialog box appears

   ![Options dialog box]

   If you select a dot (.) decimal separator, then whenever Web Connect encounters an ambiguous numerical value, it will always use the dot separator instead of the comma separator.

   If you select a comma (,) decimal separator, then whenever Web Connect encounters an ambiguous numerical value, it will always use the comma separator instead of the dot separator.

   **NOTE**

   If you do not set a Separator option, then by default in an ambiguous situation Web Connect will use a dot separator.

---

**Error management options**

The Error Management options allow you to specify the values that Web Connect will enter in the microcube on which the report is generated if errors occur.

There are two levels of error management:

- conversion errors
- cells not found in page
Conversion errors
When refreshing an Internet query, if a cell content cannot be converted to the previously set Qualification/Type, then you can configure Web Connect to act according to any of the three following rules:
- Set cell to #EMPTY: The report cell is displayed empty.
- Set cell to #ERROR: The text #ERROR will be displayed in the report cell.
- Convert column to Character: All of the cells in the column containing the error are converted to type Character and are displayed in the report.

Cells not found in page
When refreshing the Internet Query, if a cell content cannot be located in the page, the two options below are available:
- Set cell to #ERROR: The text #ERROR will be displayed in the report cell.
- Set cell to #EMPTY: The report cell is displayed empty.
Accessing secured web sites

You might find that some of the web sites you are trying to access are secured and require authentication to access them.

Web Connect allows you to set up an Internet query so that it can automatically access most secured web sites, and allows you to refresh query reports by simply entering a password.

The way you set this up depends on the design of the secured web page you are trying to access. If the web site uses forms to acquire security data from customers, then Web Connect needs to be able to parse the form data before it can process it.

To allow you to access both secured and non-secured web sites, Web Connect provides the following authentication options:

- No Authentication: This is the default selection, for web pages that do not require any authentication.
- Basic Authentication: This is for standard web pages that require authentication data, but that do not use forms.
- Form Authentication: This is for web pages that require authentication, but that are presented as forms.

The following sections explain how to set up your Internet queries to work with secured web sites, using both basic and form authentication.

Querying a page using Basic Authentication

This section explains how to set up Web Connect to access a web page that requires authentication data (but that does not use forms). Once you have set up this query authentication, you will be asked to enter your username and password when you run the query for the first time. However, after entering this authentication data once, you will not be required to enter it again during the current session.

Whenever you start a new session, you will need to enter your username and password again.
1. Start your Internet browser, and try to access a web page that requires basic authentication.
A login dialog box such as the one below might appear:

![Login Dialog Box](image)

In order to have Web Connect automatically specify a username and password when accessing this web page, you need to set up the Authentication properties of your query.
1. Click **Authentication** in the Internet Query Panel. The Authentication dialog box appears.
By default, No Authentication is selected. However, you will notice that the Basic Authentication URL box contains the template string:

http://USER:PASS@.

2. Click Basic Authentication in the Authentication Properties dialog box.

The template string is activated.
1. Extend the `http://USER:PASS@` string with the URL of the web page you are trying to access.
   In this example:
   `http://USER:PASS@localhost/wi/bin/iswi.dll/WIGenerator/wigrandator/generator/ExecuteWIS?swis=Welcome`

2. Click OK.
   The Internet Query User Identification dialog box appears.

3. Enter the required user name and password, then click OK.
   Your Internet query is now set up to automatically provide the necessary authentication data each time you run the query, or refresh reports that are based on it.

   **NOTE**
   Web sites that require Windows NT authentication are also supported, provided the domain the user is trying to log into is the default domain. Only sites requesting client certificates are supported. Sites requesting server certificates are not supported.

**Form authentication**

This section explains how to configure Web Connect to access a web page that requires authentication data which is embedded in a form. Once you have set up this query authentication, you will be asked to enter your username and password when you run the query for the first time. However, after entering this authentication data once, you will not be required to enter it again during the current session.

Whenever you start a new session, you will need to enter your password again. Your username is stored with the document.

1. Start your Internet browser and access a web page that requires form
authentication.
A web page such as the following appears.

To have Web Connect automatically specify a username and password when accessing this type of web page, you need to set up the Authentication properties of your query.
1. Click **Authentication** in the Internet Query Panel. The Authentication dialog box appears.

2. Click **Form Authentication**

3. Enter the URL of the login web page, then click **Parse**. The web page is displayed in the mini-browser at the bottom of the dialog box. You will also see that a list of parsed items is displayed in each of the four list boxes: Forms, Users, Password and Submit. If these form items do not have names assigned to them in the source web page, then they are automatically allocated standard names by Internet Explorer, as in the example above (ms_id1, ms_id2, ms_id3, ms_id4). In this example, there is only one item available per box.

4. Select an item from each of the list boxes. As each item is selected, you will see the corresponding fields highlighted in the mini-browser window.

5. Click **OK**. Web Connect will now automatically specify a User Name and Password when accessing this web page. The next time you access this page, the following dialog box will pop-up after clicking on **OK** in the Authentication dialog box.
6. Enter a valid name and password, and click **OK**.
Web Connect query examples

This section provides a series of Web Connect query and report examples which demonstrate how you can generate different report types.

• Example 1 - This example creates a chart of historical quotes using an entire table of data, and a single multivalued prompt.
• Example 2 - This example uses a combination of monovalued and multivalued prompts to build a query that allows you to retrieve the latest conversion rates between one of several currencies (monovalued) to one or to several other currencies (multivalued).

Example 1 - Creating a chart of historical quotes

You want a report in the form of a chart that displays the latest company valuations from the Nasdaq. You can generate this report by building a query that selects an entire table of data of historical quotes.

Additionally, using a single multivalued prompt, you can customize the query so that each time you refresh the report, the query asks you to choose which company valuations you want to see.
Creating a tabular report

1. Open BusinessObjects and choose the Web Connect data provider, then click New Query.
   The Internet Query Panel appears

2. Locate the web site that contains the data you require.
3. Select the first cell in the header and extend the selection to the second row. This selects the entire table.

<table>
<thead>
<tr>
<th>Date</th>
<th>Open</th>
<th>High</th>
<th>Low</th>
<th>Close</th>
<th>Volume</th>
<th>Adj Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-Jul-05</td>
<td>16.80</td>
<td>20.44</td>
<td>16.40</td>
<td>20.00</td>
<td>3,434,400</td>
<td>28.00</td>
</tr>
<tr>
<td>23-Jul-05</td>
<td>21.01</td>
<td>21.48</td>
<td>19.76</td>
<td>20.65</td>
<td>2,026,900</td>
<td>28.65</td>
</tr>
</tbody>
</table>

4. Drag the selection to the header area in the Internet Query Grid.

The entire table is replicated in the Internet Query Grid. If you double-click on any individual cell, the corresponding data is located.
and highlighted in the browser window.

5. Right-click on the Date header and click **Date** on the shortcut menu.

6. **Click Run** to generate the report.

The tabular data is presented as a standard BusinessObjects report.

<table>
<thead>
<tr>
<th>Date</th>
<th>High</th>
<th>Low</th>
<th>Open</th>
<th>Close</th>
<th>Volume</th>
<th>Adj. Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/25/2002</td>
<td>25</td>
<td>18.75</td>
<td>18.75</td>
<td>18.44</td>
<td>990,562.00</td>
<td>18.44</td>
</tr>
<tr>
<td>11/26/2002</td>
<td>16</td>
<td>18.95</td>
<td>18.95</td>
<td>18.66</td>
<td>1,793,272.00</td>
<td>18.66</td>
</tr>
<tr>
<td>11/27/2002</td>
<td>20</td>
<td>19.02</td>
<td>19.02</td>
<td>19.76</td>
<td>603,500.00</td>
<td>19.76</td>
</tr>
<tr>
<td>12/02/2002</td>
<td>21</td>
<td>19.48</td>
<td>20.00</td>
<td>19.59</td>
<td>1,025,300.00</td>
<td>19.59</td>
</tr>
<tr>
<td>12/05/2002</td>
<td>19</td>
<td>18.04</td>
<td>18.39</td>
<td>18.33</td>
<td>1,382,300.00</td>
<td>18.33</td>
</tr>
<tr>
<td>12/06/2002</td>
<td>15</td>
<td>18.81</td>
<td>19.39</td>
<td>18.20</td>
<td>1,570,900.00</td>
<td>18.20</td>
</tr>
<tr>
<td>12/09/2002</td>
<td>14</td>
<td>19.01</td>
<td>19.34</td>
<td>17.79</td>
<td>499,100.00</td>
<td>17.79</td>
</tr>
<tr>
<td>12/12/2002</td>
<td>17</td>
<td>16.00</td>
<td>15.90</td>
<td>16.17</td>
<td>614,400.00</td>
<td>16.17</td>
</tr>
<tr>
<td>12/19/2002</td>
<td>16</td>
<td>15.52</td>
<td>16.03</td>
<td>17.25</td>
<td>1,272,000.00</td>
<td>17.25</td>
</tr>
<tr>
<td>12/21/2002</td>
<td>17</td>
<td>16.00</td>
<td>17.00</td>
<td>16.97</td>
<td>1,317,000.00</td>
<td>16.97</td>
</tr>
<tr>
<td>12/23/2002</td>
<td>17</td>
<td>16.00</td>
<td>16.00</td>
<td>16.95</td>
<td>1,257,000.00</td>
<td>16.95</td>
</tr>
<tr>
<td>12/25/2002</td>
<td>16</td>
<td>15.49</td>
<td>16.30</td>
<td>16.07</td>
<td>603,500.00</td>
<td>16.07</td>
</tr>
</tbody>
</table>

► **Creating a multivalued prompt**

So far, whenever you run this query or refresh this report, you will see all the data in the table. However, by adding a single multivalued prompt, you can customize the query so that each time you refresh the report, the query asks you to choose which company valuations you want to see in the report. You will then see only these values, and not the entire table. To do this, you need to add and activate your prompt data.

**NOTE**

In this example, before adding a prompt, make sure you have a list of company ticker symbols. You will need to enter these when building the prompt. Later on, when you generate your reports, you will then be able to choose the companies that you want to include (by selecting the appropriate ticker symbol).
To create a multivalued prompt:

1. Open the report and click **Prompts** in the Internet Query Panel.
   
The Prompts dialog box appears.

2. Select the displayed URL.
   
The URL is copied to the lower URL panel.
3. Click **New** to display the Prompt Properties dialog box, and then click Insert. A new item appears in the List of Values panel.

![Prompt Properties dialog box]

4. Enter one of the ticker symbols, and press the Return key. Alternatively, select an existing list of values by clicking Import list of values (for more information on importing lists of values, refer to *Using Prompts with Web Connect Queries on page 153*).
5. Click **Insert** again to enter another ticker symbol, and repeat this process until you have entered all the symbols you require.

6. In the Prompt Name field, enter the name “Quote”.

7. Click **Create an associated column**.
   The Prompt name you entered is automatically entered into this field.

8. Click **OK** to return to the previous dialog box.
   You will see the new “Quote” prompt in the middle panel.

9. Click on the “Quote” prompt to select it.
   The prompt text is then displayed below in the “Selected prompt” box.

10. In the third panel, select the part of the displayed URL that you want to replace with the prompt.
    In this example, it is BOBJ, the ticker symbol for Business Objects. The URL
text that you select is then displayed below in the “Selected text area” box.

11. Click **Activate**.
The selected text, BOBJ, is replaced with the string @Prompt('Quote').

This string will now be used by Web Connect to retrieve any of the specific data that you are prompted to request the next time you run the query.

12. Click OK to return to the Internet Query Panel.
13. Click **Run** to run the query again.

   This time, the prompt is displayed, and you must select one or more of the available items. The report that is produced will include only the items you select here.

   ![Enter or Select Values]

14. Select the company ticker symbols from the values displayed, and click **OK**. BusinessObjects generates the report.

**Example 2 - monovalued and multivalued prompts**

This example uses a combination of monovalued and multivalued prompts to build a query which allows you to retrieve the latest conversion rates from one of several currencies (monovalued) to one or to several other currencies (multivalued).

- **Creating the query**
  1. Start Web Connect, and create a new query.
2. Locate the web site that contains the data you require. In this example, use the browser window to select two currencies for which we want the conversion rate – US Dollars and Japanese Yen.

3. Drag the required data cells into the Internet Query Panel.

---

**Adding a monovalued prompt**

Currently this report always contains US dollars and Japanese Yen but you want to make it flexible enough to handle different currencies. To do this, you need to create a couple of prompts which allow you to choose from a predefined set of currencies when you run the query.
To do this:

1. Click **Prompts** in the Internet Query Panel. The Prompts dialog box appears.

2. Select the displayed URL. The URL is copied to the lower panel.

3. Click **New** to define the first prompt.

4. For the Prompt Name, enter “Convert from”.

5. For Type, select **Monovalued**.
   
   With a monovalued prompt you will only be able to select one value in response to this prompt.

6. In the List of Values box, insert the names of the currencies that you would like to be able to convert from.
   
   Each time you generate a report, you will be able to choose one of these currencies (monovalued) to act as your base currency.

7. Click **Create an associated column** to associate the extra values you have...
added with a new column. In this case, a column name is automatically entered, corresponding to the Prompt name you entered.

8. Click **OK**. The Internet Query Prompts dialog box is re-displayed with the prompt added.
9. Select the section of the URL you want to replace with the prompt (in this case, "USD").

The selected text area is displayed at the bottom right hand side of the dialog box.

![Prompts](https://example.com/prompts.png)
10. Click **Activate**.
The prompt is added to the URL.

11. Click **OK** to finish.
The Internet Query Panel is displayed once more.
When you next run the query you will be prompted to select one of the currencies in the list of values.

**Adding a multivalued prompt**

The first prompt allows you to select a currency to convert from. Now you need to add a second prompt, this time a multivalued prompt, that allows you to select one or more currencies to convert to.

1. Click **Prompts** in the Internet Query Panel
   The Prompts dialog box appears.
2. Click **New** to define the second prompt.
3. Type “Convert to” in the Prompt Name box.
4. Select Multivalued.
A multivalued prompt allows you to select more than one value from a list of
values.
5. Type the names of the currencies that you would like to be able to convert to in the list of values.
   Each time you generate a report, you will be able to choose any or all of these currencies to display their current conversion rate against the base currency you select at the first prompt.
6. Click *Create an associated column* to associate the extra values you have added with a new column.
   A column name is automatically entered, corresponding to the Prompt name you entered.

- **Prompt Properties**

7. Click **OK**.
   The Internet Query Prompts dialog box is re-displayed, with the prompt added.
8. Select the “Convert to” prompt in the middle window.
9. To associate the newly created prompt with the URL currency field, select the section of the URL you want to replace with the prompt (in this case, “JPY” in the URL address).
   The selected text area is displayed at the bottom right-hand side of the dialog.
10. Click **Activate** to link this text to the selected prompt. The prompt is added to the URL.
11. Click **OK** to return to the Internet Query Panel.
12. Click **Run** to run the query again. Both prompts are displayed together and you must select a currency to convert from, and one or more currencies to convert to.

13. Click **OK** after selecting your prompts. BusinessObjects displays the report.
Combining Data from Different Sources
Overview

The data you need might not all come from the same source. For example, you might have business objectives in a corporate database and personal data that you store in a spreadsheet. BusinessObjects enables you to combine data from different sources in the same report.

This chapter explains
- the different data sources you can use
- how to include data from different sources in the same report
- when BusinessObjects automatically links data from different sources, and when you have to make the link yourself
Which data sources are available?

BusinessObjects lets you access data from a wide range of sources. You can access data from
- relational databases (RDBMS), such as ORACLE and Microsoft SQL Server
- multidimensional (OLAP) servers such as Microsoft OLAP Services, ORACLE Express, and IBM DB2
- text files and spreadsheets
- packaged applications such as SAP.
- almost any data source using Microsoft Visual Basic for Applications (VBA) procedures
- XML files
- pages on the web
Including data from different data sources in the same report

You access data sources in BusinessObjects by building data providers for the data sources. To include data from different sources in the same report, you display data from different data providers. For example, if you want to display data from a Sybase database and a Microsoft Excel file in the same report, you could retrieve the data from the Sybase database by building a query or by using a stored procedure and retrieve the data from Excel by accessing a personal data file. BusinessObjects supports the following types of data providers:

- queries on universes
- stored procedures
- free-hand SQL
- personal data files
- VBA procedures
- OLAP servers
- SAP
- XML
- World Wide Web pages

Which data providers can you combine in one report?

You can combine data from any BusinessObjects-supported data provider with data from any other BusinessObjects-supported data provider in a single report. For example, in a report that displays data from a query on a universe, you can build a new query on a different universe. You can also use a different type of data provider: a stored procedure, a free-hand SQL script, a personal data file, or an OLAP server.
Using separate data providers for separate blocks in one report

You can display data from separate data providers in one block or separate blocks in a BusinessObjects report. To display data from separate data providers in one block, you first create a separate block with the separate data provider and then combine data from the blocks. To create a separate block in a report using a separate data provider, follow this procedure:

1. Open a report.
2. Click **Table** (or **Crosstab** or **Chart**) on the **Insert** menu. Your choice depends on the type of block you want to insert.
3. With your mouse, draw a rectangle where you want the new block to appear.
4. When you release the mouse button, a wizard appears. Which wizard (New Table, New Crosstab, or New Chart) appears depends on the command you clicked on the Insert menu.
5. To use a separate data provider, click **Access new data in a different way**, then click **Next**. The New Table wizard with **Access new data in a different way** selected appears:
6. Click **Begin**. The Specify Data Access screen appears.

7. Select the type of data provider you want to use, then click **Finish**. The editor for the data provider appears.

8. Define and run the data provider.
   BusinessObjects can automatically link data providers. It will prompt you to link the new data provider with the existing data provider if:
   • No common dimension exists between them. Common dimensions are dimensions with the same name occurring in the same universe. Dimensions called Year that occur in a universe and an Excel spreadsheet are not common. BusinessObjects will prompt you to link them.
   • The new block is in a section.
     The new data appears in the new block.

   **Further information**
   For further information on linking data providers, refer to [Linking data providers](#) on page 212.
Displaying data from separate data providers in the same block

Once you have created a separate block in a report from a separate data provider and you manually or BusinessObjects automatically has linked the data providers, you can display data from the separate data providers in one block. You can do any of the following:

<table>
<thead>
<tr>
<th>If you</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Want to display data in an existing table or crosstab,</td>
<td>use the Pivot tab in the Table Format dialog box.</td>
</tr>
<tr>
<td>Want to display data in an existing chart,</td>
<td>use the Pivot tab in the Chart Format dialog box.</td>
</tr>
<tr>
<td>Want to display data in any type of existing block,</td>
<td>use the Slice and Dice Panel.</td>
</tr>
</tbody>
</table>

Which variables from linked data providers can you display?

Compatibility rules determine which variables from separate data providers can be combined in the same block. When you cannot include a variable in a block, it appears dimmed and italicized as shown in the following figure.
You can use the common dimension from either data provider. Most often, you can use measures from both data providers in the same block.

**Displaying data in an existing table or crosstab**

1. Click inside the table or crosstab that you want to modify with data from another data provider.
2. Click **Table** or **Crosstab** on the **Format** menu.
3. In the **Table Format** dialog box, click the **Pivot** tab.
4. In the **Used Variables** box, click the folder that represents where you want to display the data: **Columns**, **Rows**, **Body**.
5. In the **Available Variables** box, click the variable you want to add, then click **Add**.
6. Click **OK**.

**Displaying data in an existing chart**

1. Click inside the chart that you want to modify with data from another data provider.
2. Click **Chart** on the **Format** menu.
3. In the **Chart Format** dialog box, click the **Pivot** tab.
4. In the **Used Variables** box, click the folder that represents the axis where you...
Accessing Data and Data Analysis

want to display the data: **Columns, Rows, Body**.

5. In the Available Variables box, click the variable you want to add, then click **Add**.

6. Click **OK**.

   In 2-D charts, all the variables are located in the X-Axis and Y-Axis folders. In 3-D matrix charts, the variables are located in all three folders: X-Axis, Y-Axis, and Z-Axis.

   ▶ Adding data in slice and dice mode

1. With a report open, click **Slice and Dice** to display the Slice and Dice Panel.

2. Variables for all the blocks in the report are displayed in the Available Variables box.

3. Drag the icon of the variable you want to add to the report from the Available Variables box and drop it either in the Section box or in the Block Structure box.

4. Repeat the previous step for other variables you want to add.

5. Click **Apply**.

Including data from different data sources in the same report
Basing a data provider on an existing data provider

You can base new data providers on data existing data providers that use universes, personal data providers or OLAP data sources. To do this:

1. Click **Table** (or **Crosstab** or **Chart**) on the **Insert** menu. Your choice depends on the type of block you want to insert.
2. With your mouse, draw a rectangle where you want the new block to appear.
3. When you release the mouse button, a wizard appears. Which wizard (New Table, New Crosstab, or New Chart) appears depends on the command you clicked on the Insert menu.

![New Table Wizard](image-url)

**Insert a New Table**

This wizard helps you insert a new table in the active report. You must specify the data that you want to display in the new table:

- To display the new table, you can:
  - Use existing data from the document
  - Build a new query on the universe currently in use
  - Access new data in a different way
  - Use an existing query to build a new one

![Wizard Screenshot](image-url)
4. Click **Use an existing query to build a new one.** (This option is not available if your report does not already contain at least one data provider based on a universe, personal data provider or OLAP data provider.)

A list of data providers currently in the document appears.

5. Select a data provider and click **Finish.**

The Query Panel appears with the definition of the data provider you selected loaded.

6. Modify the query in the query panel, then click **Run.**

7. The table, chart or crosstab based on the new query appears in the report.

**Prompts and linking**

Because prompt name are unique throughout a report, a data provider based on an existing data provider contains prompts with names in the form `<prompt_name>_<prompt_number>` if the original data provider had prompts. For example, if the original data provider contained a prompt called `Which Country?`, the copied prompt in the new data provider is called `Which Country?_1`.

If the original data provider was linked to another data provider (see **Linking data providers on page 212** for details on linking data providers), the link is not preserved in the new data provider.
Linking data providers

Linking data providers enables data from different sources to be computed in the same table, crosstab, or chart in a report.

What situations require you to link data providers?

BusinessObjects automatically links data providers with a common dimension. Two dimensions in separate data providers are common when they belong to the same universe and have the same name. BusinessObjects prompts you to link data providers if there is no common dimension between the data providers.

If you simply want to add columns of data to a report, use the Edit Data Provider command on the Data menu instead of building a new query. This method lets you add result objects to the initial query; BusinessObjects automatically inserts the new columns of data in the report or creates a new report.

EXAMPLE

BusinessObjects prompts you to link data providers

Here’s an example scenario where BusinessObjects prompts you to link data providers:

- You create a new document by running a query on a universe.
- You format the report as a master/detail, using for example the Year dimension.
- You want to compare yearly revenue with your sales targets, so you insert a new table in the Year section.
- Rather than inserting data from the document, or using the universe you ran the first query on, you pull in data from the spreadsheet that contains your personal targets.
- Even though the spreadsheet contains the Year column, BusinessObjects prompts you to link the personal data file with the query already in the report, because you’re inserting the new data in a section that’s generated by the query data.

Which dimension should act as the link?

It is necessary that the dimension you use to link data providers be the same type (numeric or alphanumeric) in both data providers. If not, two rows of data will appear for the linked object when you create a table that uses the object.
Additionally you should use only dimensions that return the same type of values. It doesn’t make sense to create a link between dimensions with totally different lists of values (Year and Region, for example).

**To link data providers when you’re inserting a new block**

1. Click **Table, Crosstab or Chart** on the **Insert** menu.
2. In the Wizard that appears, click **Access new data in a different way**, then click **Begin**.
3. The **Specify Data Access** dialog box appears.
4. Select the type of data provider you want to run, then click **Finish**.
5. Build and run the data provider.
6. BusinessObjects displays a dialog box which prompts you to link the new data provider with the data in the report.
7. To make the link:
   - Select the linking dimension from the new data provider by clicking an icon in the Dimensions box.
   - Select the report’s section master by clicking an icon in the Master(s) in the Report box.
   - Click **Link**.
The dimension you clicked in the Dimensions box appears below the dimension in the Master(s) in the Report box:

8. Click **OK**.  
The new data appears in the report. Measures are automatically calculated.

**NOTE**  
If the Links Between Data Provider and Report dialog box appears, and you click **OK** or **Cancel** without creating a link, you will obtain a Cartesian product. A Cartesian product is a report result that returns each row from the first data provider joined to every row from the second. If the first data provider has 100 rows and the second 50, the Cartesian product contains 5000 rows.

**Linking existing data providers**  
The procedure above describes how to link data providers when you’re bringing new data to a report section. But what if you find yourself in the following situation?  
• Your report contains two tables - one from a universe, the other from a personal data file.  
• There’s no link between the tables because when you inserted the second table, you simply placed it alongside the existing table without having
previously linked their data providers.

• You now want to create the master/detail format, which is possible because the tables share a dimension with the same name and same type.

The procedure for linking existing data providers is as follows:
1. Open the document containing the data providers you want to link.
2. Click View Data on the Data menu. The Data Manager appears.
3. In the Data Providers box, click the dimension you are going to use as the link between the data providers.

In the illustration below, the Resort dimension has been selected:
4. Click the **Definition** tab, then click **Link To:**

The Define Link Between Dimensions dialog box appears. It lists the dimensions you can use to link the two data providers.

5. Click the dimension you want to use as the link.

The symbol next to the dimension name now changes to indicate that the
dimension is the link between two data providers. This symbol will also now appear when you click the dimension in the Data Manager.

6. Click **OK** to close the dialog box, then click **OK** in the Data Manager.

You can now use the linking dimension to apply a master/detail format in the report.

**Deleting the link between data providers**

To delete the link between data providers, you delete the link of the common dimension that links the data providers. You would delete the link between data providers to use all the variables in a single data provider to build an independent query. The procedure for deleting the link of the common dimension is as follows:

1. Click **View Data** on the Data menu.
   The Data Manager appears.

2. In the Data Providers box, click the dimension that acts as the link.
   Click the **Definition** tab, then click **Unlink**:

3. Click **OK** to close the Data Manager.
Overview

In what ways can you manage queries and other data providers in BusinessObjects documents? This chapter answers that question by describing how to:

• rename data providers
• get statistics such as the date and time a data provider was last refreshed, and how many rows of data were returned
• empty data providers of their data (called purging), and delete data providers you no longer need

REMINDER

Data provider is the BusinessObjects generic term for all types of queries: queries on universes, free-hand SQL, stored procedures, VBA procedures, XML files, OLAP cubes and Web Connect.
Renaming data providers

BusinessObjects assigns a name to all data providers. Here are some examples:

<table>
<thead>
<tr>
<th>Data provider name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1 on eFashion</td>
<td>The first query built on the eFashion universe in the current document.</td>
</tr>
<tr>
<td>PS1 on Stock</td>
<td>Stored procedure run on a database account called Stock.</td>
</tr>
<tr>
<td>SQL 1 with Sales</td>
<td>A free-hand SQL script run on a database connection named Sales.</td>
</tr>
<tr>
<td>PD1 in C:\My Documents\Forecast.xls</td>
<td>Personal data coming from a spreadsheet stored in My Documents.</td>
</tr>
<tr>
<td>VBA 1 with ThisDocumentApp</td>
<td>The first VBA data provider built in the current document, where ThisDocumentApp is the name of the VBA macro itself.</td>
</tr>
</tbody>
</table>

Why rename data providers?

Renaming data providers is by no means required in BusinessObjects. You can work with the software without ever thinking about these names.

However, data provider names appear in several places in BusinessObjects, for example in the Report Manager:

Here are a few cases where you might want to rename data providers.
Documents with data from different sources

In documents containing two or more data providers, the names of variables may also contain the name of the data provider that they belong to. For example, Year (Query 2 on eFashion) is the Year dimension from the second query built on the eFashion universe.

This additional information only appears if the data providers in the document contain variables with the same name. In such cases, to help you distinguish between variables with the same name, which in fact come from different data sources, BusinessObjects automatically inserts the full variable name for example Year (Query 2 on eFashion) in column headings in the report.

Selecting the data provider you want to edit

Editing a data provider in a document with multiple data providers also causes each data provider’s name to appear. When you select the Edit Data Provider command on the Data menu, you are prompted to select one data provider in the following dialog box:

Giving data providers more meaningful names can help you know right away which one you want.

To rename data providers

You rename data providers in the Data Manager. To do this:
1. Open the document containing the query you want to rename.
2. Click View Data on the Data menu.
   The Data Manager appears.
3. Click the icon of the data provider you want to rename, then click the
**Definition** tab:

4. Type the new name in the Name box, then click **OK**.
Getting statistics on data providers

BusinessObjects allows you to find out useful information on data providers, such as when a query was last refreshed, how many rows of data a personal data file returned, and how long it took to refresh a free-hand SQL script. To get these statistics:

1. Open the document containing the data provider.
2. Click View Data on the Data menu.
   The Data Manager appears.
3. Click the icon of the query you want to find out about, then click the Definition tab.
   The information appears in the bottom right corner of the dialog box:
Purging and deleting data providers

What’s the difference between purging and deleting a data provider? *Purging* means emptying a data provider of its results, *deleting* means getting rid of the data provider for good—an action that cannot be undone.

Why purge or delete a data provider? Purging reduces the size of a document, so is useful when you want to send the document to other users, or save it on a diskette, for example. You should only delete a data provider, however, if you are certain that you and other users no longer need it.

To purge or delete a data provider:
1. Open the document containing the data provider.
2. Click **View Data** on the **Data** menu.
   The Data Manager appears.
3. Click the icon of the query you want to purge or delete:

   ![Data Manager screenshot]

4. Click **Purge** or **Delete**.
5. Click **Yes** in the confirmation box that appears, then click **OK** to close the Data Manager.
You can populate a purged data provider by clicking Refresh Data on the Data menu. Note that this command refreshes all the data providers in the active document.
Using data providers efficiently

If you have multiple reports on separate report tabs and these reports draw their data from the same source, you do not need to create a separate data provider for each report. Instead you create a “base” data provider that contains the data used by all the reports. This approach is better because BusinessObjects performs one data retrieval for each data provider; it is more efficient to retrieve data once and share it among reports than to retrieve the same data several times.

**EXAMPLE**

Reports showing revenue by country and resort, revenue by country

In this example the Revenue and Country objects are common to both reports. Instead of creating a data provider for each report you create a data provider containing the Revenue, Country, and Resort objects and use these objects in both reports. To do this:

1. Create a report showing revenue by country and resort by dragging the Country, Resort and Revenue objects into the report. (See Building a basic query on a universe on page 56 for an explanation of how to build a report on a universe.)

2. Add an additional report tab by clicking Report on the Insert Menu. Drag the Country and Revenue objects to the new report tab to create the revenue by country report.

When you run this report, BusinessObjects retrieves the data once and shares it between the two reports, rather than retrieving the data twice.

**NOTE**

You can use the Query Panel to modify a data provider by clicking Edit on the Data menu. Do this rather than create a new data provider if you wish to add a new report to a new tab and the new report uses the same or similar data to your existing report(s).
Analyzing Data
Introduction to Data Analysis
Overview

With BusinessObjects, you analyze data by looking at it on different levels of detail and from different viewpoints. Through your analysis, you gain new information and thereby answer questions.

You need go no further than the BusinessObjects interface to address all your multidimensional analysis needs:

- BusinessObjects on-report analysis allows you to work directly on your data in your report using drag and drop and or with simple mouse clicks.
- Explorer, an optional component in BusinessObjects, enables you to carry out multidimensional analysis in Drill mode.
- OLAP servers are databases that store summarized data, ready for business analysis
- BusinessObjects Slice and Dice mode allows you to organize data for analysis in the slice and dice panel.
On-report analysis

This part of the User's Guide describes how BusinessObjects on-report analysis allows you to analyze your data directly on your report using easy mouse clicks and drag and drop or with a simple mouse click.

You can drag and drop data on your report to get a different viewpoint for your analysis. You can add data from the Report Manager to create tables and sections. You can replace, swap and re-organize data on the report.

BusinessObjects redoes the calculations in your report immediately so that you can see at once how different combinations of factors affect your performance.

Dynamic on-report grouping allows you to create groups for comparative analysis and you can quickly insert common business calculations or easily create your own formulas and variables for analysis.

You can sort, filter and rank your data using a simple mouse click to focus your analysis on a slice of data.
BusinessObjects drill mode

BusinessObjects drill mode allows you to analyze data from different angles and on different levels of detail. Typically, you start off by looking at the high level data and when you spot an unusually low or high value, or an unexpected value, you can analyze it by displaying related data on a more detailed level. This allows you to see how different factors of your business, seasonal, geographical, product line affect your revenue.

See Analyzing Data in Drill Mode on page 237 for more information on drill mode.
OLAP servers

If you work with an OLAP server, you can view and select the data you want when creating a report. OLAP servers are databases that store summarized data, ready for business analysis.

For more information on OLAP servers, refer to the OLAP Access Pack documentation for the server at your site.
Slice and dice mode

"Using Slice and Dice Mode" on page 271 describes how you can use the slice and dice panel to add, delete and swap data round, to analyze your data from a different viewpoint.
Analyzing Data in Drill Mode
Accessing Data and Data Analysis

Overview

This chapter describes how to use BusinessObjects drill mode.

What is drill mode?

Drill mode is a BusinessObjects analysis mode that allows you to break down data and view it from different angles and on different levels of detail to discover what the driving factor is behind a good or bad result.

EXAMPLE

Why is revenue better in this resort than in the others?

You see that revenue is much higher in the Hawaiian Club resort than in the others. To find the reason for this, you need to look at all the factors that affect revenue.

<table>
<thead>
<tr>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>571,444</td>
</tr>
<tr>
<td>French Rivier</td>
<td>835,003</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>1,475,000</td>
</tr>
</tbody>
</table>

If you displayed all the factors that could be useful for analysis in a table or chart, there would be too much data, making the report difficult to read. Drill mode allows you to include data for analysis behind the scenes of your report and display the top level data only. If necessary, you can drill down to the more detailed data to understand the higher-level data. This allows you to see how different aspects of your business affect your revenue step by step.

This underlying data is set up by the person who creates the report.
How does drill mode work?

When you make a query on a BusinessObjects universe, the objects you can include are grouped in folders and organized in a specific order.

The person who creates the BusinessObjects universe organizes objects in hierarchies, with the most general object in the class at the top and the most detailed at the bottom. Objects are grouped in this way to make it easy for you to find what you are looking for. They are classified inside the groups so that if you want to make a high level report you know you need to include objects at the top of the list in your query and if you want a more detailed report then you choose objects from further down the list.

Hierarchies

Objects are also organized in this way for drilling. When you analyze data in drill mode, you use hierarchies. The universe classes are the default hierarchies you use for drilling but the universe designer can also set up custom hierarchies. You can also create and edit hierarchies in your reports.

Drill hierarchies contain dimension objects only. In drill mode, you drill down on dimensions, for example from Year to Quarter to Month. At each level BusinessObjects recalculates measures such as Revenue or Profit Margin.
The classic dimensions on which a designer or advanced user creates hierarchies are geography, time and product. In the demo universe, Island Resorts Marketing, there are four hierarchies:

- Resort (Country, Resort, Service Line, Service)
- Sales (Year, Quarter, Month, Week, Invoice Date)
- Customer (Country of Origin, Region, City, Customer)
- Reservations (Reservation Year, Reservation Quarter, Reservation Month, Reservation Week, Reservation Date)

When you set up a report for drilling, you include high level objects to display in your table or chart but include more detailed objects in your scope of analysis. BusinessObjects retrieves these objects from the database and stores them behind the scenes in your report so that they are there when you need them.

Before you can analyze data in drill mode, you have to set up this behind-the-scenes data.

For information on how to set up data for analysis in drill mode see Defining scope of analysis on page 63.
Using drill mode

Before you can analyze data in drill mode, your report must contain data that has been set up for analysis.

Switching to drill mode

You open drill mode from a BusinessObjects report. To do this:

1. Select the table, crosstab or chart that you want to analyze in drill mode.
   You can analyze only one block at a time.
2. Click Drill on the Standard toolbar or click Drill on the Analysis menu.
   If no part of any table, chart or crosstab was selected before you clicked the drill button, the cursor becomes a magnifying glass with a question mark next to it.
   ![Drill Cursor]
   This cursor appears if you do not select a block before switching to drill mode.

If this happens, click inside the table, chart or crosstab you want to analyze.

When you go into drill mode, by default:

- BusinessObjects creates a new report that contains a copy of the selected table, crosstab or chart. The report tab displays the drill icon to show you are in drill mode. The original report remains intact.
- BusinessObjects adds a sum on measure objects.
- If you are working on a master/detail report, BusinessObjects displays the Drill toolbar.

**NOTE**
You can change these default behaviors. See Setting options for working in drill mode on page 270 for more details.

You are now ready to drill on the data in the report.

Drilling down

When you drill down, you display the next level of detail in a hierarchy.

- **To drill down**
  1. Rest the cursor over the data.
     The cursor changes to a magnifying glass with a plus sign in it. The plus sign...
indicates that you can drill down on this value. A tooltip shows you the next dimension in the hierarchy:

<table>
<thead>
<tr>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beverly Beach</td>
<td>751,444</td>
</tr>
<tr>
<td>French Resort</td>
<td>483,363</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>1,478,893</td>
</tr>
</tbody>
</table>

The tooltip over Resort shows Service Line, which is the next dimension in the hierarchy. This means that by drilling on a Resort value, you will display values for Service Line.

2. Double-click the value.
The data for the next dimension appears in the table and the selected value appears in the Drill toolbar.

<table>
<thead>
<tr>
<th>Service Line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accommodated</td>
<td>581,280</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>277,790</td>
</tr>
<tr>
<td>Recreation</td>
<td>220,700</td>
</tr>
</tbody>
</table>

The selected value, Hawaiian Club, now appears in the Drill toolbar. The table now displays the revenue for Service Line services for Hawaiian Club only.

Continuing to drill down

Each time you drill down BusinessObjects moves the value you drilled on to the drill toolbar and filters the data in the report according to the values in the Drill toolbar.

<table>
<thead>
<tr>
<th>Service</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Food</td>
<td>56,885</td>
</tr>
<tr>
<td>Poolside Bar</td>
<td>35,200</td>
</tr>
<tr>
<td>Restaurant</td>
<td>203,760</td>
</tr>
</tbody>
</table>

The table now displays the revenue for all services in the Food & Drinks category for Hawaiian Club only.

You can drill down as long as there are objects in the hierarchy. When you reach the last level in a hierarchy, the normal cursor is displayed. This indicates you are at the bottom of a hierarchy. If tooltips are turned on, a tooltip displays the message *Right-click to explore.*
Displaying different values in the Drill toolbar

The block is filtered to only show data for the values currently displayed in the Drill toolbar. You can change the values in the toolbar to look at data for a different value.

1. Click the down arrow.

2. Choose a value from the drop-down list.
   The data for the chosen value is displayed in the table.

Drilling up

Drilling up is the opposite of drilling down. When you drill up, you go back up through the hierarchy to display data on less detailed levels.

- **To drill up from one dimension to the next**
  1. Place the cursor over a value.
  2. Right-click on the value you want to drill up on and click **Drill Up** on the shortcut menu.
     BusinessObjects displays the data for the next level up.

BusinessObjects updates the Revenue column to show revenue for French Riviera.
Undoing drill actions

You can undo up to ten drill actions which can be useful if you lose track of your analysis. To do this:

• Click Undo on the Edit menu.

Drilling across

When you drill down and up you move through the levels of the same hierarchy. However, if you cannot find the answer to a question by analyzing data in its current hierarchy, you can move to another hierarchy to analyze other data.

EXAMPLE

Move from analyzing Resort to Sales

Your report displays data for Resort, which belongs to the Resort hierarchy. The report also contains Year, which belongs to the Sales hierarchy, but the data for Year is not currently displayed. If you drill across from Resort to Year, you can then drill down on the next dimension in the Sales hierarchy. You can also drill across to other hierarchies, or back to the hierarchy you were originally working in. Drilling across opens up new paths that you can follow when analyzing data.

Before you can drill across, your report must contain dimensions from more than one hierarchy. If this is not the case, you can:

• Retrieve data for dimensions from more than one hierarchy by expanding your scope of analysis, or
• Create new hierarchies inside the report.
To drill down to another hierarchy

1. Right-click a value and click **Drill By** from shortcut menu.
   The list of dimensions that you can drill to appears on a sub-menu:

   In this example, you selected Food & Drink from the Service Line column of the table. The first dimension in the list, Service, is the next dimension down in the current hierarchy, the Resort hierarchy.

   The next three dimensions belong to the Sales hierarchy.

   The next dimension belongs to the Customer hierarchy.

   More opens up a list of all the available dimensions.

2. Choose a dimension from the list.
Drill By - More

The list in the Drill By sub-menu displays five dimensions only by default. To display the full list of dimensions:

1. Right-click a value and click **Drill By More** on the shortcut menu.
   
The Drill By - All Available Dimensions dialog box appears.

2. Choose a dimension from the list and click **OK**.
   
The dimensions displayed in gray are already used in the current analysis.

**NOTE**

You can change the setting in the Drill By menu to display more items. See **Setting options for working in drill mode on page 270** for more information.
Drilling on charts

You can also drill on chart blocks in the same way you drill on tables and crosstabs.

1. Rest the cursor over a part of the chart.
   A tooltip appears to indicate the next dimension down in the hierarchy and the cursor turns to a magnifying glass.

2. Double-click.
   The chart is updated and the value you drilled on is displayed in the Drill toolbar.

**NOTE**
You cannot drill on the data series of 2D and 3D Area charts.
Drilling on multiple hierarchies

If your block contains more than one hierarchy, you can simultaneously drill down from one dimension to the next in all the hierarchies in the block. To do this, you have to drill down on a measure object.

In the table illustrated below, Resort belongs to the Resort hierarchy and Year to the Sales hierarchy. If you rest the cursor over the Resort column, you see that the next level down is Service. If you rest the cursor over the Year column, you see that the next level down is Quarter. You could drill down on one hierarchy and then the other or you could drill down on both at the same time by drilling on the Revenue column.

<table>
<thead>
<tr>
<th>Resort</th>
<th>Year</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>FY1998</td>
<td>$107,928.00</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>FY1999</td>
<td>$307,400.00</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>FY2000</td>
<td>$376,115.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>FY1998</td>
<td>$296,940.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>FY1999</td>
<td>$200,310.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>FY1998</td>
<td>$479,685.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>FY1999</td>
<td>$69,195,00.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>FY2000</td>
<td>$460,445.00</td>
</tr>
</tbody>
</table>

To do this:

1. Rest the cursor over the Revenue column. A tooltip shows you can drill down on both Service Line and Quarter.

   ![Tooltip showing drill down on two hierarchies](image)

2. Double-click on the Revenue column in the cell that displays revenue for Hawaiian Club in FY1998. This is the highest value in the column. BusinessObjects moves Resort and Year to the Drill toolbar. The values in the
Drill toolbar are Hawaiian Club and FY1998. Service Line and Quarter appear in the table and BusinessObjects updates the Revenue column.

<table>
<thead>
<tr>
<th>Service Line</th>
<th>Quarter</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accommodations</td>
<td>Q1</td>
<td>$108,456.00</td>
</tr>
<tr>
<td>Accommodations</td>
<td>Q2</td>
<td>$73,740.00</td>
</tr>
<tr>
<td>Accommodations</td>
<td>Q3</td>
<td>$64,960.00</td>
</tr>
<tr>
<td>Accommodations</td>
<td>Q4</td>
<td>$803,170.00</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>Q1</td>
<td>$17,170.00</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>Q2</td>
<td>$21,590.00</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>Q3</td>
<td>$24,760.00</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>Q4</td>
<td>$24,360.00</td>
</tr>
<tr>
<td>Recreation</td>
<td>Q1</td>
<td>$19,400.00</td>
</tr>
<tr>
<td>Recreation</td>
<td>Q2</td>
<td>$18,800.00</td>
</tr>
<tr>
<td>Recreation</td>
<td>Q3</td>
<td>$19,400.00</td>
</tr>
<tr>
<td>Recreation</td>
<td>Q4</td>
<td>$18,800.00</td>
</tr>
</tbody>
</table>

**To drill up on multiple hierarchies**

To drill up on all dimensions:

- Right-click on a measure object value and click **Drill Up** on the shortcut menu.
Getting a different view of your data

As you work, you can change the data to analyze by inserting, removing or replacing dimensions and measures in the block.

You can only replace data with data that is of the same type. You can replace measures with measures and dimensions with dimensions.

<table>
<thead>
<tr>
<th>To...</th>
<th>Right click...</th>
<th>From the menu click...</th>
<th>Then choose...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert a variable</td>
<td>• in a table, on the cell below or to the right of where you want to insert the variable</td>
<td>Insert</td>
<td>the variable you want to insert</td>
</tr>
<tr>
<td></td>
<td>• in a chart, on the data series (or its data label, or its legend key) next to which you are going to insert the variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace a variable</td>
<td>on the data you want to replace</td>
<td>Replace</td>
<td>the variable you want to use from the list</td>
</tr>
<tr>
<td>Delete a variable</td>
<td>the data you want to delete</td>
<td>Delete</td>
<td></td>
</tr>
</tbody>
</table>

NOTE
If the Insert and Replace commands are not available when you click on data, this means there is no compatible variable with which to replace the selection.

Changing the data in tables as you drill

As you work on tables or crosstabs in drill mode, you can move data around from the Report Manager to the Drill toolbar or to and from your table or crosstab. You can add, replace and remove data as required to get a different viewpoint for your analysis.

• You can move variables from the Report Manager window to the Drill toolbar or to a table or crosstab in the report.
• You can move variables from the Drill toolbar to a table or crosstab in the
NOTE
You cannot drag and drop variables onto charts.

The following tables summarize how you can move data around between the Report Manager, Drill toolbar and tables and crosstabs.

**Report Manager to drill toolbar**
You can drag a variable from the list in the Report Manager and drop it in the Drill toolbar.

<table>
<thead>
<tr>
<th>To...</th>
<th>Click...</th>
<th>Drag...</th>
<th>Drop when...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert a variable</strong></td>
<td>the icon of the variable you want to move</td>
<td>the variable to where you want to insert it</td>
<td>the Drill toolbar shows the Insert highlighting the status bar displays <em>Drop to insert</em></td>
</tr>
<tr>
<td><strong>Replace a variable</strong></td>
<td>the icon of one of the variables</td>
<td>the variable over the variable you want to replace</td>
<td>the Drill toolbar shows the Replace highlighting the status bar displays <em>Drop to replace contents</em></td>
</tr>
</tbody>
</table>

**Data tab or Drill toolbar to table or crosstab**
You can also drag a variable from the Drill toolbar or Report Manager to the table or crosstab you are working on.
Accessing Data and Data Analysis

Analyzing Data in Drill Mode

Using the Drill toolbar

You use the Drill toolbar to filter the data displayed in the block you are analyzing. If you hold your cursor over one of the boxes a tooltip appears showing you:

- which hierarchy the chosen value belongs to
- the name of the dimension
- a list of the top three values available for that dimension

Three dots at the end of the list of values indicates that more values are available than those shown in the tooltip list. You can display a value from the list by typing in the first letter on the keyboard.

<table>
<thead>
<tr>
<th>To...</th>
<th>Click...</th>
<th>Drag...</th>
<th>Drop when...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert a variable</td>
<td>the icon of the variable you want to insert</td>
<td>the variable to where you want to insert it</td>
<td>the table or Drill toolbar is shows the Insert highlighting the status bar displays Drop to insert</td>
</tr>
<tr>
<td>Replace a variable</td>
<td>the icon of one of the variables</td>
<td>the variable over the variable you want to replace</td>
<td>the table or Drill toolbar is shows the Replace highlighting the status bar displays Drop to replace contents</td>
</tr>
</tbody>
</table>

Removing an object from the Drill toolbar

1. Click the arrow next to the object name.
2. Choose (Remove) from the list.

The cell disappears from the drill toolbar and the data for the variable is no longer displayed.
Moving an object from the Drill toolbar to the block

1. Click the arrow next to the object name.
2. Choose (Move to block) from the list.
   The cell disappears from the Drill toolbar and a column of data is added to the table.

Re-organizing the Drill toolbar

You can re-organize the order in which variables are displayed in the Drill toolbar.

<table>
<thead>
<tr>
<th>To...</th>
<th>Click...</th>
<th>Drag...</th>
<th>Drop when...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move a variable</td>
<td>the icon of the variable you want to move</td>
<td>the variable to where you want to insert it</td>
<td>the Drill toolbar shows the Insert highlighting the status bar displays <em>Drop to insert</em></td>
</tr>
<tr>
<td>Swap two variables</td>
<td>the icon of one of the variables</td>
<td>the variable over the variable you want to swap it with</td>
<td>the mouse cursor becomes a swap icon the status bar displays <em>Drop to swap</em></td>
</tr>
</tbody>
</table>

TIP

You can drag the drill toolbar from its docked position under the other toolbars and place it where it is convenient for you in your workspace.
Analyzing measures in drill mode

Measures display numeric data that is the result of calculations. For example, Profit Margin is a measure that is the result of Revenue - Cost.

A report can contain two different types of measure objects:
- those created by the universe designer and which are retrieved by a query (or measures retrieved by other data providers)
- measures created locally in the report

You can analyze measures in drill mode only if you have created them locally, based on data in the document that you are working on.

**NOTE**

For information on creating measure objects locally, see "Formulas, Local Variables and Functions" on page 465.

In drill mode, you can expand a measure in order to view its component parts in the report and analyze numeric data.

Collapsing a measure is the opposite of expanding it. Instead of viewing the measure's component parts, you display its aggregated values once more.

**EXAMPLE**

**Analyzing profit margin**

The data for Customer, Service and Margin are displayed in a report. Margin is a local variable with the following syntax: Product Price - Product Cost.

When you expand Margin, the data for Product Price and Product Cost appear in the report. When you collapse the expanded measure, the report shows the data for Margin only.

**To expand a measure**

To expand a measure while working in drill mode:

1. Right-click the cell or chart element that displays the measure you want to expand.
2. Click **Expand** on the shortcut menu.
   
   The measure's component parts are displayed in the block.
To collapse a measure while working in drill mode:

1. Right-click the cell or chart element that displays the measure you want to collapse.
2. Click **Collapse** on the shortcut menu.
   The data for the measure's component parts disappears.

**NOTE**

If the Expand and Collapse commands are not available when you click on a measure, it means the variable was not created locally in the report and therefore cannot be analyzed.
Making copies of reports while you work

To keep track of the different stages of your analysis you can make copies of your work as you go along. Each copy you make of the report appears in a new tab inside the document.

To make a copy of a report:

- Click **Take Snapshot** on the Report toolbar.

  A copy of the report appears in a new tab inside your document. The name that appears in the tab is *Report Name*(n+1). For example, if the report you copied is named Sales, the new report is named Sales (1).

**NOTE**

The filters displayed in the Drill toolbar when the snapshot was taken are turned into global report filters.
Extending analysis

You can bring new data into your report if it does not contain all the data you need for your analysis. You do this by extending the scope of analysis in the Scope of Analysis window or editing your query to retrieve more data from the database.

Expanding scope of analysis

To expand the scope of analysis:

1. Right-click on the block you are analyzing.
2. Click **Scope of Analysis** on the shortcut menu.
   The Scope of Analysis dialog box appears.

3. Choose the dimensions to include in your scope of analysis and click **OK**. The dimensions are now displayed in the list in the Report Manager window.

**NOTE**

BusinessObjects shows you which values are currently displayed in the Drill toolbar by placing a filter icon next to them.
Drilling through to the database to bring in new data

If the lowest level of detail you need is not currently available in the report, you can drill through to the database directly from drill mode and get the data you need. You do not have to edit the query in the Query Panel.

Here’s how to do it:

1. Right-click on the column or row where you want the new data to be displayed.
2. Choose Drill Through from the menu.
   The Drill Through dialog box is displayed.

3. Choose a dimension and click OK.
   The dimension is retrieved from the database and displayed in the block you are analyzing.

**NOTE**

You can choose one dimension only at a time.
Bringing in new data using filters

You can use the filters applied in drill mode as query conditions when you bring in new data from the database. The example below illustrates how this works.

**EXAMPLE**

**Focus analysis on high-profile resort using drill filters as query conditions**

You are interested in analyzing the excellent revenue results in the US. You have drilled down on Country and see that the resort Hawaiian Club has generated the best revenue. You now want to concentrate on this one resort and find out some information on the country of origin of the customers for that resort, data that is not currently available in your report.

To do this you first need to turn the drill filters option on:

1. Click Options on the Tools menu.
   The Options dialog box appears.
2. Click the Drill tab.
3. Click Apply Drill Filters box in the Drill Through section.

You can now retrieve the new data from the database. Your drill mode report currently looks like this:

<table>
<thead>
<tr>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>$971,444.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>$1,259,662.00</td>
</tr>
<tr>
<td><strong>Sum:</strong></td>
<td><strong>$2,251,106.00</strong></td>
</tr>
</tbody>
</table>

You drilled down on Country and the table displays revenue for US resorts only.
1. Right-click on Hawaiian Club in the Resort column and click **Drill Through** on the shortcut menu.

The Drill Through dialog box appears.

The dimensions already available in the report are grayed out. The Country dimension has a filter to display data only for the US and the Resort dimension a filter to display data only for Hawaiian Club. These filters will be used as query conditions and only data concerning the US resort Hawaiian Club will be retrieved from the database.

Notice that France is no longer available in the Country list and Hawaiian Club is the only name in the resort list. The table now displays data for Hawaiian Club only.
Drilling using custom hierarchies

You may find that the default order in which dimension objects are arranged in hierarchies is not optimal for your analysis needs, or that you need to drill on a hierarchy that has objects from different classes.

You can edit a hierarchy by changing the order of the dimensions it contains, by adding dimensions to it, and removing dimensions from it. You can also rename a hierarchy, and even delete it.

You can also create your own hierarchies using dimensions available in the report or by using user objects. There are two types of custom hierarchies:

- Pre-defined custom hierarchies that are set up by the universe designer and which can be re-used in other documents.
- Hierarchies you create yourself and which are saved only in the document you created them in.

Editing hierarchies

The changes you make to a hierarchy that was created by a universe designer affect your work in drill mode only. For example, if you delete a hierarchy, you can no longer use it in drill mode. However, the hierarchy remains unchanged in the universe so you can still use the hierarchy to define scope of analysis when building a query on the universe. The universe designer is the only person who can edit or delete hierarchies at the universe level.
To edit a hierarchy for drill mode

1. Click **Hierarchies** on the Analysis menu.
   The Hierarchy Editor opens.

2. In the Available Hierarchies box, click the + sign to view the contents of the hierarchy folder.

3. In the Available Dimensions box, click the + sign to the left of the folder to view the list of dimensions.

4. Make the required changes and click **OK** to close the Hierarchy Editor.

To add a dimension to an existing hierarchy

- Click the dimension you want to add to the hierarchy, then click Add.

The dimension you clicked appears in the hierarchy's folder in the Available Hierarchies box.

If the Hierarchy Editor does not list the dimension that you want to add to your hierarchy, you may have to expand your scope of analysis. See **Expanding scope of analysis on page 257.**

**NOTE**

You cannot combine dimensions from different data providers in a single hierarchy.
To change the order of the dimensions in a hierarchy
1. Click the dimension you want to move up or down in its hierarchy.
2. Click Move Up or Move Down to change the dimension's position in the hierarchy.

To rename a hierarchy
1. Click the hierarchy you want to rename.
2. Click a second time on the hierarchy name, then type the new name.

To remove a dimension from a hierarchy
• Click the dimension you want to delete, then click Remove.

To delete a hierarchy
• Click the hierarchy you want to delete, then click Remove.

Creating hierarchies
You can create your own custom hierarchies from any dimensions available in the report. The dimensions you include in a hierarchy can be local variables, derived variables, or dimensions returned by data providers.

NOTE
You can also use a date-type user object as the basis for a time hierarchy. For information on user objects see Creating user objects on page 331.

To create a custom hierarchy
1. Click Hierarchies on the Analysis menu.
2. The Hierarchy Editor opens.
3. Click New in the Hierarchy Editor.
4. Type the name of the new hierarchy, then click outside the name box.
5. In the Available Dimensions box, click the first dimension for the new hierarchy, then click Add.
6. The dimension you clicked appears in the new hierarchy's folder in the Available Hierarchies box.
7. Add the other dimensions you want to include and then click OK.
If the Hierarchy Editor does not list the dimensions that you want to include in the hierarchy, you may have to expand your scope of analysis. See Expanding scope of analysis on page 257.
NOTE
You cannot combine dimensions from different data providers in a single hierarchy.
Qualifying data for hierarchies

BusinessObjects qualifies data as dimensions, measures or details.
Hierarchies contain dimensions only, so if you want to include an object in a hierarchy for analysis in drill mode, you must qualify it as a dimension.

In preparation for drill mode, you might need to change the qualification of user objects, variables or formulas that you have created in your report. For example, you want to include a variable in a hierarchy, but cannot because the variable is qualified as a detail. In this case, you must qualify the variable as a dimension before you can include it in the hierarchy.

You can change the qualification of data returned by stored procedures, free-hand SQL scripts, and personal data files. You can also change the qualification of user objects, local variables and formulas.

NOTE
You cannot change the qualification of data returned by a query on a universe.

Re-qualifying local variables and formulas
1. Select one of the following report elements that displays the formula or local variable you want to qualify:
   • A cell in a table or a crosstab.
   • In a chart, the data series, a data label or the legend.
   • A free-form cell.
2. Click Variables on the Data menu.
   The Variables dialog box opens.
3. Select the local variable or formula from the list. Local variables are stored in the Variables folder and formulas are stored in the Formulas folder.

4. Click **Edit**.

5. In the Definition tab of the Variable Editor, click an option button to change the qualification, then click **OK**.

6. Click **Close** in the Variables dialog box.

**Re-qualifying variables**

1. Click **View Data** on the **Data** menu.
2. The Data Manager dialog box opens.
3. In the Data Providers box of the Data Manager, click the icon that represents the column of data whose qualification you want to change.
4. Click the **Definition** tab.
5. The name, type, qualification and values for the column are displayed.
6. In the Qualification box, click an option button to change the column's qualification.
7. Click **OK**.
Re-qualifying user objects

1. Click **Universes** on the **Tools** menu.
2. The Universes dialog box opens.
3. Select the universe that contains the user object, then click **User Objects**.
4. Click **Edit**.
5. In the Qualification box of the Definition tab, click an option button to change the qualification, then click **OK**.
6. Click **OK** in the User Objects dialog box.
Printing from drill mode

You can print out a table, crosstab or chart from drill mode. Before printing you can insert the contents of the Drill toolbar into to your report to keep track of the filters applied.

Inserting Drill toolbar contents as a title

1. Click Special Field on the Insert menu then click Drill Filters.
2. The cursor becomes an insert cell cursor.
3. Holding down the left mouse button, draw a box on your drill mode report page in the position where you would like to display the title.
4. Release the mouse button.
5. A cell is inserted that contains a list of the values currently displayed in the Drill toolbar.

You can edit the size and formatting of the cell just as you would edit any cell in a report.

Printing a report from drill mode

1. Make sure the report you want to print is active.
2. Click Print on the File menu.
Setting options for working in drill mode

There are a number of options you can set to manage how you work in drill mode. You can:
- control the number of items that appear on the popup sub-menus
- view the number of values for each dimension that appears on the popup menu
- systematically create a new report when you switch to drill mode, or be prompted to choose whether or not to create a new report
- control the cursor and the tooltip that appear in drill mode
- automatically display the totals or percentages of numeric data (measures)
- choose to display a message before closing drill mode
- set drill filters as query conditions when you drill through to retrieve more data from the database

To set options for drill mode:
1. Click Options on the Tools menu. The Options dialog box opens.
2. Click the Drill tab:

3. Set the required options and click OK.
Using Slice and Dice Mode
Overview

Slice-and-dice mode enables you to switch the position of data in a report, for example by moving columns to rows to create a crosstab.

You can also use slice-and-dice mode to:
• work with master/detail reports
• display and remove data
• rename, reset and delete blocks
• turn tables and crosstabs into charts, and vice versa
• apply, edit and delete breaks, filters, sorts, rankings and calculations

You access slice-and-dice mode through the Slice and Dice Panel, a pop-up window that provides a graphical representation of the report you are working on. You carry out slice-and-dice tasks by dragging and dropping icons that represent your data.
Working in slice-and-dice mode

To display the Slice and Dice Panel, click **Slice and Dice** on the Analysis menu, or click **Slice and Dice** on the Standard toolbar.

The Slice and Dice Panel appears.

You can rename, transform, reset or delete a block by clicking its tab with the right mouse button, then clicking a command on the shortcut menu that appears.

**Working with master/detail reports in slice and dice mode**

Master/detail reports display data in sections. Each section contains a “master” or parent piece of data, for example a resort, or a year. The rest of the data in the section relates to the master.
The following illustration shows a master/detail report, and its corresponding representation in the Slice and Dice Panel:

**A master/detail report and its representation in the Slice and Dice Panel**

The Slice and Dice Panel makes it easy to work with master/detail reports. You can:
- build a master/detail report
- reorganize a master/detail report by replacing the master or by building a master/master detail report
- undo a master/detail report by removing the master
- deactivate sections of master/detail reports

The following sections describe how to perform these tasks.
To structure an existing report as a master/detail report in slice and dice mode, you drop the icon of the master in the Section box.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then...</th>
</tr>
</thead>
</table>
| Use data that is already displayed in the report, | 1. Drag an icon from the Block Structure box  
2. Drop it in the Section box  
3. Click Apply. |
| Use data that is not yet displayed in the report, | 1. Drag an icon from the Available Variables box  
2. Drop it in the Section box  
3. Click Apply |

You can perform this task in drill mode. To do so, switch to drill mode and select the data you want to use as master. Click your right-mouse button and on the shortcut menu that appears, click **Set as Master**.
Reorganizing a master/detail report

You can re-organize a master/detail report by:

- Using a different master. For example, if the master is Year, you can replace it with Resort.
- Building a master/master/detail report. This structure enables you to view data on two levels of detail. The illustration below shows one section of a master/master/detail report. For FY1998, the report displays revenue for each resort per quarter:

<table>
<thead>
<tr>
<th>FY1998</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td></td>
</tr>
<tr>
<td>Resort</td>
<td>Revenue</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>$76,395.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>$76,200.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>$111,005.00</td>
</tr>
<tr>
<td>Q2</td>
<td></td>
</tr>
<tr>
<td>Resort</td>
<td>Revenue</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>$76,390.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>$85,800.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>$111,330.00</td>
</tr>
<tr>
<td>Q3</td>
<td></td>
</tr>
<tr>
<td>Resort</td>
<td>Revenue</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>$76,730.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>$81,035.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>$129,120.00</td>
</tr>
<tr>
<td>Q4</td>
<td></td>
</tr>
<tr>
<td>Resort</td>
<td>Revenue</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>$56,426.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>$123,020.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>$126,720.00</td>
</tr>
</tbody>
</table>

One section of a master/master/detail report
To use a different master

You can replace the master with data that is already displayed in the report, or with data that is not yet displayed. Here is the procedure:

1. Open a master/detail report.
2. Select the icon of the master in the Section box, then press the Delete key.
3. The icon disappears from the Section box.
4. To use data that is already displayed in the report, drag an icon from the Block Structure box, and drop it in the Section box.
5. To use data that is not yet displayed, drag an icon from the Available Variables box, and drop it in the Section box.
6. Click Apply to display the report with its new master.

You can swap the master with data that is displayed in the report. To do so, hold down your Shift key, then drag the master until it is located above the icon with which you want to swap it, in the Block Structure box. Release your mouse button, then click Apply.

To build a master/master detail report

A master/master detail report contains two masters, as its name suggests. This means that in the Slice and Dice Panel, there must be two icons in the Section box.

You are most likely to build a master/master/detail report from an existing master/detail report. Here is the procedure:

1. Open a master/detail report, then open the Slice and Dice Panel.
2. Drag an icon to the Section box.
   - You can drag an icon from the Available Variables box, or from the Block Structure box.
3. Drop the icon until it is located just below the existing master, then release
4. Click **Apply** to display the master/master detail report.

   **To undo a master/detail report**

   Undoing a master/detail report means removing the master. In the Slice and Dice Panel, this means that you remove the master from the Section box.

   **If you want to...**
   **Then...**
   
   **Move the master to a block,**
   1. Drag the master from the Section box to the Block Structure box
   2. Click **Apply**.

   **Remove the master from the report,**
   1. Select the master in the Section box.
   2. Press the Delete key.
   3. Click **Apply**
Deactivating sections of master/detail reports

Deactivating sections of a master/detail report enables you to recalculate the report without removing data from it. This feature is particularly useful in reports that contain multiple blocks, as the following example describes.

The section of the report illustrated below displays revenue and number of guests per quarter for FY1998. The chart on the left shows revenue per quarter, while the table on the right shows number of guests per quarter:

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Number of guests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>45</td>
</tr>
<tr>
<td>Q2</td>
<td>48</td>
</tr>
<tr>
<td>Q3</td>
<td>50</td>
</tr>
<tr>
<td>Q4</td>
<td>46</td>
</tr>
</tbody>
</table>

**FY1998**

**Bahamas Beach**

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Number of guests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>39</td>
</tr>
<tr>
<td>Q2</td>
<td>45</td>
</tr>
<tr>
<td>Q3</td>
<td>39</td>
</tr>
<tr>
<td>Q4</td>
<td>31</td>
</tr>
</tbody>
</table>

**French Riviera**

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Number of guests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>45</td>
</tr>
<tr>
<td>Q2</td>
<td>39</td>
</tr>
<tr>
<td>Q3</td>
<td>45</td>
</tr>
<tr>
<td>Q4</td>
<td>46</td>
</tr>
</tbody>
</table>

**Hawaiian Club**
You want to obtain the number of guests per quarter for all resorts, rather than the number of guests per resort, as shown in this illustration:

<table>
<thead>
<tr>
<th>FY1998</th>
<th>Number of guests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>127</td>
</tr>
<tr>
<td>Q2</td>
<td>132</td>
</tr>
<tr>
<td>Q3</td>
<td>134</td>
</tr>
<tr>
<td>Q4</td>
<td>125</td>
</tr>
</tbody>
</table>

You obtain this result by deactivating the Resort section for the table. The Slice and Dice Panel enables you to perform this task with mouse clicks. After you have done so, BusinessObjects dynamically recalculates the number of guests for all resorts while leaving the data in the chart unchanged.

**To deactivate a section of a master/detail report**
1. Open a master/detail report, then open the Slice and Dice Panel.
2. Click the tab of the block (table, chart or crosstab) that you want to deactivate.
recalculate.

3. In the Section box, with your right-mouse button click the master of the section that you want to deactivate.

4. Click **Deactivate this section** on the shortcut menu.

5. Click **Apply**.

   The block appears in the section above the section that you deactivated.

**NOTE**

If you select a master that has masters beneath it, the pop-up menu gives you an additional option to deactivate the master plus all its child masters. Similarly, if you select a deactivated master that is not at the top of its master hierarchy, the menu gives the option to reactivate the master plus all its parent masters. See **To deactivate or activate two or more sections at the same time** on page 282.
To reactivate a section of a master/detail report

Once you have deactivated a section of a master/detail report, you can reactivate it in the following way:
1. Open the master/detail report, then open the Slice and Dice Panel.
2. Click the tab of the block (table, chart or crosstab) that you want to recalculate.
3. In the Section box, with your right-mouse button click the master that you now want to activate.
4. Click **Activate this Section** on the shortcut menu.
5. Click **Apply**.

You can also perform this task by selecting and dragging the block back to its original section. To do this, hold down your Alt key and click inside the block. Position the cursor on the block’s border. When the cursor changes to a cross, as shown in the margin, click the border. Hold down your mouse and drag the block to its original section. When you release the mouse button, BusinessObjects dynamically recalculates the data in the block.

To deactivate or activate two or more sections at the same time

In master/master/detail reports, which by definition contain two or more sections, you can deactivate two or more sections at the same time:
1. Open the master/detail report, then open the Slice and Dice Panel.
2. Click the tab of the block (table, chart or crosstab) that you want to recalculate.
3. In the Section box, click the master of the uppermost section that you want to deactivate with your right-mouse button.
4. Click **Deactivate this section and all sections below it** on the shortcut menu.
5. Click **Apply**.

BusinessObjects recalculates the report.
6. To reactivate the sections:
7. Click the master of the lowest section with your right-mouse button (Quarter, in the example above).
8. Click **Activate this section and all sections above it**, then click **Apply**.

Positioning data horizontally in slice-and-dice mode

All reports display data horizontally. In tables and crosstabs, data appears in columns, and in charts, data is plotted on the X-axis that runs from left to right.
In the Slice and Dice Panel the Block Structure box shows dimensions and measures that appear horizontally in a table, crosstab or chart.

You can use the Slice and Dice Panel to move variables horizontally, using drag-and-drop. You can also swap two variables.

- **To drag a variable horizontally to a new position**
  
  Select an icon, hold down your mouse button and drag the icon horizontally, left or right, to its new position. As you move the mouse, the cursor changes, as shown in the margin. Release your mouse button to drop the icon at its new position. Click Apply to display the data in its new position in the report.

- **To swap two variables horizontally**
  
  To swap two variables horizontally, hold down the Shift key and select one of the icons you want to swap. The cursor changes to show that you are carrying out a swap operation, as shown in the margin. Drag the icon horizontally until it is above the other icon you want to swap, then release your mouse button. Click Apply to display the swapped data in the report.
Working with crosstabs and 3-D matrix charts

The Slice and Dice panel makes working with crosstabs and 3-D matrix charts simple. The Block Structure box clearly shows the structure of the crosstab or 3-D matrix chart.
The following illustrations show the crosstab and the 3-D matrix chart that correspond to the representation in the Block Structure box above.

<table>
<thead>
<tr>
<th></th>
<th>FY1998</th>
<th>FY1999</th>
<th>FY2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>287,529.00</td>
<td>307,400.00</td>
<td>376,115.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>225,940.00</td>
<td>282,310.00</td>
<td>252,170.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>479,895.00</td>
<td>519,530.00</td>
<td>480,445.00</td>
</tr>
</tbody>
</table>

### Crosstab

#### 3-D matrix chart

In slice-and-dice mode you can perform the following tasks with crosstabs and 3-D matrix charts:

- build a crosstab or 3-D matrix chart from a table or 2-D chart, respectively
- reduce a crosstab or 3-D matrix chart to a table or 2-D chart, respectively
- reposition data that appears in rows or on the Z-axis
- move data between rows and columns (in crosstabs), and between the Z-axis to the X-axis (in 3-D matrix charts)
- turn crosstabs into 3-D matrix charts and vice versa

The following sections describe how to perform these tasks, with the exception of turning crosstabs into 3-D matrix charts and vice versa. This task is described in “Transforming blocks in Slice-and-Dice mode” on page 289.
To build a crosstab or a 3-D matrix chart from a table or 2-D chart

Here is the procedure:
1. Click inside a table or 2-D chart, then open the Slice and Dice Panel.
2. Select the icon of the data that you want to use to build the crosstab or matrix chart.
3. Drag the icon until it is positioned above the icon that is furthest to the right in the Block Structure box, then release your mouse button.
4. The icon appears above and to the right of the other icons in the Block Structure box, as illustrated on page 284.
5. Click Apply to make the crosstab or matrix chart appear.

Which icons you take depends on whether you want to include existing or new data:

<table>
<thead>
<tr>
<th>To do this...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use data that is already displayed in the report,</td>
<td>take an icon from the Block Structure or Section boxes.</td>
</tr>
<tr>
<td>Use data that is not yet displayed,</td>
<td>take an icon from the Available Variables box.</td>
</tr>
</tbody>
</table>

To reduce a crosstab or 3-D matrix chart to a table or 2-D chart

When you reduce a crosstab or a 3-D matrix chart into a table or 2-D chart, you take data out of the crosstab or matrix chart. You can:
• remove data from the report
• display the data in the 2-D chart or table

Here is the procedure:
1. Click inside a crosstab or 3-D matrix chart, then open the Slice and Dice Panel.
2. Select the icon of the variable whose data appears in rows (if you are working with a crosstab), or on the Z-axis (if you are working with a chart.) This icon is located in the upper right-hand corner of the Block Structure box.
3. If you want to remove the data from the report, press the Delete key.
   If you want to display the data in the table or 2-D chart, drag it down and to the left, until it is at the same level as the other icons. Then, release your mouse button.
4. Click Apply to make the table or 2-D chart appear.
To reposition data vertically
You can place more than one variable in rows in crosstabs and on the Z-axis of 3-D matrix charts. You can reposition these variables, as illustrated here:

In this example, the data for Quarter and Year would appear in columns in a crosstab, or on the Z-axis of a 3-D matrix chart. You can move Year up, or Quarter, down.

To move a variable up or down
Select an icon, hold down your mouse button and drag the icon vertically, up or down, to its new position. As you move the mouse, the cursor changes, as shown in the margin. Release your mouse button to drop the icon at its new position, then click Apply.

To swap two variables vertically
To swap two variables vertically, hold down the Shift key and select one of the icons you want to swap. The cursor changes to show that you are carrying out a “swap” operation, as shown in the margin. Drag the icon vertically until it is above the other icon you want to swap, then release your mouse button. Click Apply.
To move data between columns and rows in crosstabs

You can move data between the columns and rows of a crosstab using the Slice and Dice Panel, by moving icons from and to the upper right-hand corner of the Block Structure box:

- To move data from rows to columns
  Select an icon in the upper right-hand corner of the Block Structure box, hold down your mouse button and drag it to the bottom left-hand corner of the box. As you move the mouse, the cursor changes, as shown in the margin. Release your mouse button and click **Apply**.

- To swap two variables between columns and rows
  Hold down the Shift key and select one of the icons you want to swap. Drag the icon until it is above the other icon you want to swap. The cursor changes to show that you are carrying out a "swap" operation, as shown in the margin. Release your mouse button and click **Apply**.

To move data between the X- and Z-axis in 3-D matrix charts

This task is equivalent to moving data between columns and rows in crosstabs. Follow the procedures described in the preceding paragraphs. For "column" read "X-axis" and for "row" read "Z-axis".

Displaying and removing data in Slice-and-Dice Mode

The Slice and Dice Panel shows the data that is displayed in a report. It also shows any unused data that you can display. More precisely:

- The icons that you can see in the Section box, and in the Block Structure box, show the data that is already displayed in the report.
- The icons in the Available Variables box show all the data you can use,
whether or not it is already displayed.

Using the drag-and-drop technique, you can display unused data in the report, and you can remove data that is already displayed. Once removed, the data remains available for later use.

► To display data in the report
1. In the Available Variables box, drag an icon to the Section box to display it as a master, or double-click it to display it in the block.
2. Click Apply to display the data in the report.

► To remove data from the report
1. Select an icon in the Section box or the Block Structure box.
2. Drag the icon to the Available Variables box, and release the mouse button. Alternatively, press the Delete key.
3. Click Apply to remove the data from the report.

Deleting, renaming and resetting blocks in Slice and Dice mode

You can delete, rename and reset blocks in slice-and-dice mode. To do so:
1. Click the tab of the block you want to work on. To select multiple blocks, hold down the Ctrl key then click each one.
2. Click your right-mouse button. The shortcut menu appears.
3. Click the command that corresponds to the task you want to perform.
   • If you select the Rename command, you must type the new name in the dialog box that appears, then click OK.
   • The Rename command is not available for multiple tabs.
4. Click Apply.

TIP

You can reset one or more blocks using the procedure described above. To reset the whole report, click the Reset button on the Slice and Dice Panel toolbar. Resetting a block or a report removes any formatting you have applied.

Transforming blocks in Slice-and-Dice mode

In slice-and-dice mode, you can turn tables and crosstabs into charts, and vice versa. To do so:
1. Open the Slice and Dice Panel, then click the tab of the block you want to
transform.

2. Click your right-mouse button, then select a command:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Click...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn the selected table or crosstab to a chart</td>
<td>Turn to Chart</td>
</tr>
<tr>
<td>Turn the selected chart to a table</td>
<td>Turn to Table</td>
</tr>
<tr>
<td>Turn the selected matrix chart to a crosstab</td>
<td>Turn to Crosstab</td>
</tr>
</tbody>
</table>

3. Click **Apply**.

### Applying further modifications in slice and dice mode

You can use the Slice and Dice Panel to apply the following features on data that is displayed in a report:

- sorts, which control the order in which the data appears
- ranking, which enable you to view top and bottom values of selected data.
- calculations
- breaks, which break up blocks of data.
- filters, which enable you to view only the data that you need.

There is a button for each feature in the Slice and Dice Panel toolbar:

```
  a. Sorts
  b. Ranking
  c. Calculations
  d. Breaks
  e. Filters
```

None of the above functionalities is specific to slice and dice mode; however, the Slice and Dice Panel’s graphical interface provides a user-friendly way to apply them. Moreover, it is useful to be able to perform slice and dice operations, then apply one or more of these features to the report. For example, if you build a master/detail report that displays revenue by resort, you can quickly calculate the total revenue per resort.

In the sections that follow, you learn how to use the Slice and Dice Panel to apply sorts, ranking, calculations, breaks and filters on report data. References to more information on the features are provided in their respective sections.
Using sorts to control data order
To apply a sort on data in the report:

1. Select an icon in the Section box or the Block Structure box.
2. Click **Apply Sort**.
   A sort icon appears next to or below the icon you selected.

   You can also:
   - Invert the sort. To do this, double-click the sort icon.
   - Remove the sort. To do this, select its icon then press the Delete key.

   For further information on sorts and how to apply them on report data, refer to the *BusinessObjects User’s Guide: Report Techniques and Formatting*.

Using ranking to view only top and bottom values of data
To apply ranking on data in the report:

1. Select an icon in the Section box or the Block Structure box.
2. Click **Apply Ranking**.
   A ranking icon appears next to or below the icon you selected.
3. Double-click the ranking icon.
   The **Select Top/Bottom Variable Name** dialog box appears, where you define the ranking you wish to apply.
4. Click **OK**, then, in the Slice and Dice Panel, click **Apply**.
   The report appears with the ranking you applied.

   You can also:
   - Redefine the ranking by double-clicking its icon in the Slice and Dice Panel, then modifying its attributes in the **Select Top/Bottom Variable Name** dialog box.
   - Remove the ranking. To do this, select its icon then press the Delete key.

   For further information on ranking and how to apply them, refer to the *BusinessObjects User’s Guide: Report Techniques and Formatting*.

Making calculations on data
To make a calculation on data in the report, select an icon in the Section box or the Block Structure box, then click **Insert Calculation**. A calculation icon appears next to or below the icon you selected.
To select the calculation:
1. Double-click the calculation icon.
   The Calculation On dialog box appears.
2. Select the function(s) you wish to apply, then click OK.
3. Click Apply.
   The calculations appear in the report.

For information on how to apply calculations without using slice-and-dice mode, refer to the BusinessObjects User’s Guide: Report Techniques and Formatting.

Applying breaks
You can apply a break on data that is displayed in tables or crosstabs. You cannot apply a break on a master, or on data that is displayed in a chart.

To apply a break:
1. Select an icon in the Block Structure box, then click Apply Break.
   A break icon appears next to or below the icon you selected.
   Click Apply.
   BusinessObjects applies the break to the report.

You can also:
• Redefine a break by double-clicking its icon. In the dialog box that appears, you can select different attributes for the break.
• Remove a break. To do this, select its icon, then press the Delete key.

For further information on breaks and how to apply them, refer to the BusinessObjects User’s Guide: Report Techniques and Formatting.

Using filters to view only the data you need
To apply a filter on data in the report:
1. Select an icon in the Section box or the Block Structure box.
2. Click Apply Filter.
   A filter icon appears next to or below the icon you selected.
3. Double-click the filter icon.
   The Filter On dialog box appears, where you define the filter you wish to apply.
4. Click OK, then, in the Slice and Dice Panel, click Apply.
   The report appears with the filter you applied.

You can also:
• Redefine the filter by double-clicking its icon in the Slice and Dice Panel, then
modify its attributes in the Filter on dialog box.

- Remove the filter. To do this, select its icon then press the Delete key.

For further information on filters and how to apply them, refer to the BusinessObjects User's Guide: Report Techniques and Formatting.
Using Slice and Dice Mode
Customizing Report Data
Filtering and Ranking Data
Overview

A BusinessObjects report can contain a great deal of data. This chapter presents the various ways in which you can organize and present report data to bring certain key information to the instant attention of your readers.

• You can limit the amount of information displayed in a report to focus on a selection only by using filters.
• You can order the information by using ranking or sorting.
• You can highlight interesting data with special formatting by using alerters.
Limiting the data displayed

You might not want to display all the data returned by a data provider in a report. You might want to focus on a selection of it only, which is difficult if there is a lot of data on the screen. A filter enables you to hide data you do not want to view behind the scenes and display only the data you need.

There are two types of filter. A global filter affects the whole report. A block-specific filter filters data for the specified chart, table or crosstab only.

**EXAMPLE**

Filter data to show sales revenue for two regions only

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Region</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>West</td>
<td>$25,000.00</td>
</tr>
<tr>
<td>Q1</td>
<td>South</td>
<td>$19,502.00</td>
</tr>
<tr>
<td>Q1</td>
<td>Mid West</td>
<td>$109,798.00</td>
</tr>
<tr>
<td>Q2</td>
<td>East Coast</td>
<td>$19,124.00</td>
</tr>
<tr>
<td>Q2</td>
<td>West</td>
<td>$71,100.00</td>
</tr>
<tr>
<td>Q2</td>
<td>South</td>
<td>$103,210.00</td>
</tr>
<tr>
<td>Q3</td>
<td>Mid West</td>
<td>$105,330.00</td>
</tr>
<tr>
<td>Q3</td>
<td>West</td>
<td>$85,160.00</td>
</tr>
<tr>
<td>Q3</td>
<td>South</td>
<td>$106,914.00</td>
</tr>
<tr>
<td>Q4</td>
<td>Mid West</td>
<td>$110,582.00</td>
</tr>
<tr>
<td>Q4</td>
<td>West</td>
<td>$85,396.00</td>
</tr>
<tr>
<td>Q4</td>
<td>South</td>
<td>$110,643.00</td>
</tr>
<tr>
<td>Q4</td>
<td>Mid West</td>
<td>$115,404.00</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td>$1,106,257.00</td>
</tr>
</tbody>
</table>

You can filter the data for Region by selecting two of its values: East Coast and Mid West.

BusinessObjects now displays the values for these objects only.

In this example you want to show the sales revenue results for two regions only. To do this, you insert a filter on the Region column and choose to display East Coast and Mid West only.

Notice how the filter also affects calculations. The sum is different in the two tables.

**Inserting a filter**

You can insert a filter on data in tables, crosstabs or charts.
To insert a filter

1. Click the row, column or chart element you want to filter.
2. Click the Insert Filter button on the Report toolbar, or click Filter on the Insert menu.

The Apply a Filter On dialog box appears. It displays the values you can select for the filter:

3. Hold down the Ctrl key and click the values you want to include in the report, then click OK.

The dialog box closes, and the report includes only the values you selected. The Insert Filter button on the toolbar is dimmed.

To remove the filter, click inside the filtered data, then click Insert Filter.

Managing filters

You can manage filters in the Filters dialog box in the following ways:
• Select different values for existing filters.
• Add new filters.
• Insert filters on variables that are not displayed in the report.
• Specify whether a filter be applied on the whole report or on a specific block.
• Remove filters.

The following sections describe these tasks.
Selecting different values for existing filters

Once you have defined a filter by specifying the values you want to display, you can edit it by selecting different values. For example, instead of displaying data for the East and Midwest regions, you can select different values and display data for the West and South.

To select different values for an existing filter:
1. Click inside the block or master cell where the filtered data appears.
2. Click Filters on the Format menu.
   The Filters dialog box appears.

3. In the Filters On list, click the variable whose filter you want to edit.
   In the Values box, the values that are currently displayed in the report are highlighted (East and Midwest, in the illustration above.)
4. To select different values, you can:
   - Click values that are already selected. The highlighting disappears, which shows that the values will not appear in the report.
   - Select previously unselected values. Before you do this, check Show All Values to display all the values for the variable.
   - Click Select All Values. Doing this enables you to keep the filter, yet view all the data for the selected variable.
5. When you are done, click OK or Apply.
Adding filters

The Filters dialog box enables you to add filters to the report. To add a filter:

1. Click inside the report then click Filters on the Format menu.
   The Filters dialog box appears.

2. Double-click a folder in the Filters On box:
   - Filters in the Global folder affect the whole report.
   - Filters in the BlockName folder are block-specific.

3. Click Add.
   The Variable(s) to Filter dialog box appears. It lists all the variables in the document that are not currently filtered, whether or not they are displayed in the current report:

4. Click the variable you want to filter, then click OK.
   You return to the Filters dialog box. The variable(s) you clicked in the Variable(s) to Filter dialog box appear(s) in the Filters on box.

5. In the Values box, hold down the Ctrl key and click the values to display.

6. If you selected more than one variable to filter in the Variable(s) to Filter dialog box, repeat step 5, then click OK or Apply.
TIP
You can drag-and-drop a filter between folders. This enables you to change the way a filter is applied. For example, if you drag a filter from a Block Name folder to the Global folder, then click Apply or OK, you apply the filter on the whole report rather than on one block.

Removing a filter
When you remove a filter, BusinessObjects displays all the data for the variable on which you applied the filter. For example, if you applied a filter that displays revenue for the East and Midwest regions only, then you remove that filter, BusinessObjects displays revenue for all regions.

To remove a filter:
1. Click in the data on which you applied the filter.
2. Click Insert Filter on the Report toolbar.

NOTE
You can also choose the Filters command from the Format menu, highlight the filtered variable in the Filters On box, then click Remove.

Displaying filter names in a special field
To keep track of the global filters you have applied to the whole report, you can automatically insert the names of the filtered values you have placed in the global filters folder as a comment or title to your report.

To do this:
1. Click Special Field, then click Global Filters on the Insert menu. The cursor changes to the insert cell cursor.
2. Draw a box in the area of the report where you want insert the names of the filtered values.
3. When you release the mouse button BusinessObjects inserts the names.

You can also insert the data in an existing cell. To do this:
1. Select the cell.
2. Click Special Field, then click Global Filters on the Insert menu. BusinessObjects displays the names of the filtered values in the selected cell.
Creating more complex filters

Simple filters enable you to restrict the data of a variable in the report. Complex filters enable you to display values that satisfy conditions. You define a complex filter by writing a formula.

For more information on using the Formula Editor to write formulas, see "Formulas, Local Variables and Functions" on page 465.

**EXAMPLE**

Display only those stores with weekly revenue over $200,000

You publish a weekly report on sales revenue per store but only want to show the results of your better performing outlets.

To do this, you define a complex filter that displays only those stores whose revenue is equal to or over $200,000. To do this, you write a formula that states that revenue must be greater than or equal to $20,000. The syntax is as follows:

\[(\text{Revenue} \geq 20000)\]

Each week, when you refresh your report with the new sales data, only those stores with a revenue over $200,000 will be listed in the table.

**Inserting a complex filter**

1. Click inside the section or block that displays the data you want to filter.
2. Click Filters on the Format menu.
3. In the Filters dialog box that appears, click a folder in the Filters On box:
   - To apply the filter on the whole report, click Global.
   - To apply the filter on a specific block, click the folder with that block’s name.
4. Click Add.
   The Variable(s) to Filter dialog box appears.
5. Click the variable you want to filter, then click OK.
6. Click Define.
   The Formula Editor opens.
7. Type the formula in the Formula box, or double-click the function(s),
variable(s) or operator(s) you need.
8. Click **OK** to return to the Filters dialog box.
9. If you wish, deselect values for the variable in the Values box.
   The filter will only be applied on the remaining selected values.
10. Click **Apply** or **OK**.

BusinessObjects identifies a complex filter with an asterisk next to the filter symbol.

**Editing a complex filter**
1. Click inside the block which contains the filtered data.
2. Click **Filters** on the **Format** menu.
3. In the Filters On box, click the filter you want to edit, then click **Define**.
4. Edit the filter formula in the Formula box, then click **OK**.
5. If you wish, deselect values for the variable in the Values box. The filter will be applied on the remaining selected values only.
6. Click **Apply** or **OK**.

**NOTE**
If there is a syntax error in the formula you enter, BusinessObjects displays an error message. Click OK to return to the formula. BusinessObjects automatically selects the incorrect part of the syntax. Correct the error and click OK; you cannot use the formula if it contains an error.

**Ignoring filters**
You can force BusinessObjects to ignore any filters you have inserted on a report so that it calculates on all data, not just the filtered values. To do this, you use the NoFilter function. The syntax is:

```excel
=NoFilter(formula)
```

The example below shows how this works.

<table>
<thead>
<tr>
<th>City</th>
<th>Year</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>2001</td>
<td>$1,307,209</td>
</tr>
<tr>
<td>New York</td>
<td>2002</td>
<td>$2,763,503</td>
</tr>
<tr>
<td>New York</td>
<td>2003</td>
<td>$3,151,022</td>
</tr>
<tr>
<td>Washington</td>
<td>2001</td>
<td>$693,211</td>
</tr>
<tr>
<td>Washington</td>
<td>2002</td>
<td>$1,215,188</td>
</tr>
<tr>
<td>Washington</td>
<td>2003</td>
<td>$1,315,188</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City</th>
<th>Year</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>2001</td>
<td>$1,307,209</td>
</tr>
<tr>
<td>New York</td>
<td>2002</td>
<td>$2,763,503</td>
</tr>
<tr>
<td>New York</td>
<td>2003</td>
<td>$3,151,022</td>
</tr>
<tr>
<td>Washington</td>
<td>2001</td>
<td>$693,211</td>
</tr>
<tr>
<td>Washington</td>
<td>2002</td>
<td>$1,215,188</td>
</tr>
<tr>
<td>Washington</td>
<td>2003</td>
<td>$1,315,188</td>
</tr>
</tbody>
</table>
You filter the City column so that the report displays the data for New York and Washington only.

In the first table, the sum includes New York and Washington revenues only. The formula to calculate this sum is:

\[ \text{Sum(<SalesRevenue>)} \]

In the second table, formula for calculating the sum includes the NoFilter function. As a result, the sum includes revenues for all cities. The formula is as follows:

\[ \text{NoFilter (Sum(<SalesRevenue>))} \]

Notice the difference in the two sums.
Ordering data

You can change the order in which data appears in rows and columns by applying a sort on the data. For example, you can sort a column of city names to have the cities appear in alphabetical order. You can apply a sort on text, dates or numbers. You apply sorts from the Sorts toolbar.

Sorting data

There are three types of sorts that you can apply to data:

<table>
<thead>
<tr>
<th></th>
<th>Text</th>
<th>Numbers</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ascending</td>
<td>A-Z</td>
<td>lowest to highest</td>
<td>past to present</td>
</tr>
<tr>
<td>Descending</td>
<td>Z-A</td>
<td>highest to lowest</td>
<td>present to past</td>
</tr>
<tr>
<td>Custom</td>
<td>Place values in the order you want</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

The default sort order for all types of data is ascending.

You apply sorts from the Sorts toolbar. To access this toolbar, click Sorts on the Report toolbar.

1. Click the cell, column, row or chart element containing the data you want to sort.
2. Click the toolbar button for the sort you want to apply. The data appears in order, and the button you clicked remains pushed, to show that the data has been sorted.

**Applying a sort on report data**

1. Click the cell, column, row or chart element containing the data you want to sort.
2. Click the toolbar button for the sort you want to apply. The data appears in order, and the button you clicked remains pushed, to show that the data has been sorted.

**Inverting a sort**

1. Click the data you have already sorted
2. Click one of the sort buttons on the toolbar. For example, if you want to invert an ascending sort, click **Descending Sort** button.
Removing a sort
The Sort buttons work as toggle buttons turning sorts on and off. To remove a sort:
1. Click the data you have already sorted.
2. Click the button you used to apply the sort.
Applying a custom sort

1. Click the cell, column, row or chart element where the data you want to sort is displayed.
2. Click **Custom Sort** on the toolbar.
   The Custom Sort dialog box opens.

3. Either
   - choose an option from the Sort Option list box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Sorts the values in the order they were in when the Custom Sort dialog box appeared. This option cancels the order you have specified but does not close the dialog box.</td>
</tr>
</tbody>
</table>
Filtering and Ranking Data

or
- drag and drop the values in the list into the desired order
- use the Move Up and Move Down buttons to place the values in the desired order

4. Click OK to close the Custom Sort dialog box.

**NOTE**

You cannot use custom sorts and alerters. For example, if you set up an alerter to highlight months greater than May, BusinessObjects will use the alphabetical sort order instead of the chronological sort order.

The following example shows a table that has two custom sorts.

**EXAMPLE**

**How can I get months to sort correctly?**

In this table the months have been sorted to display in chronological order and the product lines have been manually sorted.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphanumeric</td>
<td>Displays values in alphabetical or numerical order.</td>
</tr>
<tr>
<td>Month</td>
<td>Displays the months of the year in chronological, not alphabetical, order. Only applicable for a variable that returns the months of the year.</td>
</tr>
<tr>
<td>Day</td>
<td>Displays the days of the week in chronological, not alphabetical, order. Only applicable for a variable that returns the days of the week.</td>
</tr>
</tbody>
</table>
By default, BusinessObjects sorts months in alphabetical order. To sort months correctly in chronological order:
1. Click in the column or row that displays the months.
2. Click **Custom Sort** on the Sorts toolbar.
   The Custom Sort dialog box opens.
3. Choose Month from the Sort Option list box and click **OK**.
   BusinessObjects sorts the months correctly.

**REMINDER**
BusinessObjects sorts months and days according to the Regional Settings made in the Windows Control Panel. See the *BusinessObjects User’s Guide: Report Techniques and Formatting* for more information on checking these settings.

**Managing multiple sorts**
You can apply more than one sort on report data and specify the order in which you want to apply the sorts. The following example shows how this can be useful.
EXAMPLE

Sort customers by nationality and then by name in alphabetical order

<table>
<thead>
<tr>
<th>Country of origin</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Dupont</td>
</tr>
<tr>
<td>France</td>
<td>Gantl</td>
</tr>
<tr>
<td>France</td>
<td>Mottin</td>
</tr>
<tr>
<td>France</td>
<td>Michaud</td>
</tr>
<tr>
<td>France</td>
<td>Pajot</td>
</tr>
<tr>
<td>France</td>
<td>Robert</td>
</tr>
<tr>
<td>France</td>
<td>Sarlos</td>
</tr>
<tr>
<td>Germany</td>
<td>Durnstein</td>
</tr>
<tr>
<td>Germany</td>
<td>Fussman</td>
</tr>
<tr>
<td>Germany</td>
<td>Schiller</td>
</tr>
<tr>
<td>Germany</td>
<td>Schulte</td>
</tr>
<tr>
<td>Germany</td>
<td>Talman</td>
</tr>
<tr>
<td>Germany</td>
<td>Weintra</td>
</tr>
<tr>
<td>Japan</td>
<td>Arai</td>
</tr>
<tr>
<td>Japan</td>
<td>Kamata</td>
</tr>
<tr>
<td>Japan</td>
<td>Kusamura</td>
</tr>
<tr>
<td>Japan</td>
<td>Makino</td>
</tr>
<tr>
<td>Japan</td>
<td>Mikuimoto</td>
</tr>
<tr>
<td>Japan</td>
<td>Okumura</td>
</tr>
<tr>
<td>Japan</td>
<td>Onoishi</td>
</tr>
</tbody>
</table>

This table sorts data first by applying an ascending sort to the country column and then by applying a secondary ascending sort to the customer column.
Define sort priority in a report
You can change the order in which sorts are applied.

1. Click inside the block or master cell containing sorted data.
2. Click the Sorts on the Format menu.
   The Sorts dialog box appears:

   ![Sorts dialog box]

   Primary sort -
   BusinessObjects sorts the table by the country first.

   Secondary sort -
   BusinessObjects sorts the table by customer name after sorting by country.

If your report contains a table or a 2-D chart, the dialog box contains one tab only. If your report contains a crosstab or a matrix chart, the dialog box has both tabs, Across Edge and Down Edge. The Down Edge tab displays the sorts applied to columns. The Across Edge tab displays the sorts applied to rows or on the Z-axis of the chart.

3. Click the tab you want to work in, then click the sort that you want to prioritize.
4. Click Move Up to give the sort higher priority, or Move Down to give it lower priority.
   The sort icon moves up or down one place in the list of sorts.
5. Click OK or Apply.
   The sort priority you defined is applied to the report.
Adding sorts from the sorts dialog box
You can add, remove and re-organize sorts in the Sorts dialog box.
1. Click inside a block or a master cell, then click the Sorts command on the Format menu.
2. Click the tab you want to work in.
3. Click Add.
4. Choose the variable to sort and click OK.
   The new sort appears in the Current Sorts box.
5. Set the sort options you want to apply and click OK.

To remove a sort
• Click the Sort in the Current sorts list and click Remove.
Using ranking to view the top and bottom values

You might want to show just the extreme ranges of your data, for example your top ten customers.

Ranking enables you to look at the largest and smallest numbers in a report. Like filtering, it hides the data you do not want to display. BusinessObjects does not delete the data from the report; you can view it again whenever you like by removing the ranking.

Ranking also sorts the data in descending order. Thus, the largest value of the ranking is always at the top of the ranked column and the smallest value at the bottom.

You can rank data in tables, crosstabs or master cells in master/detail reports.

**EXAMPLE**

Display the three top-selling product lines only

In the example below, the table shows sales revenue for the product lines in the efashion retail chain. The table on the left shows revenue for all product lines. In the table on the right, the product lines column shows only the top three-selling lines.

<table>
<thead>
<tr>
<th>Lines</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>$9,914,846</td>
</tr>
<tr>
<td>City Skirts</td>
<td>$347,715</td>
</tr>
<tr>
<td>City Trousers</td>
<td>$264,734</td>
</tr>
<tr>
<td>Coats</td>
<td>$2,165,020</td>
</tr>
<tr>
<td>Jackets</td>
<td>$977,618</td>
</tr>
<tr>
<td>Leather</td>
<td>$167,413</td>
</tr>
<tr>
<td>Outerwear</td>
<td>$11,162,993</td>
</tr>
<tr>
<td>Overcoats</td>
<td>$436,529</td>
</tr>
<tr>
<td>Shirt Waist</td>
<td>$4,018,220</td>
</tr>
<tr>
<td>Sweaters</td>
<td>$2,839,095</td>
</tr>
<tr>
<td>Sweat Pant</td>
<td>$12,675,090</td>
</tr>
<tr>
<td>Trousers</td>
<td>$903,300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lines</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweat T Shirts</td>
<td>$12,675,090</td>
</tr>
<tr>
<td>Accessories</td>
<td>$9,914,846</td>
</tr>
<tr>
<td>Shirt Waist</td>
<td>$4,018,220</td>
</tr>
</tbody>
</table>

Applying a ranking on report data

To apply a ranking:

1. Click to select the data you want to rank. For example, if you want to rank the
data for customers, click the column where this data appears.

2. Click **Apply Ranking** on the Report toolbar, or click **Ranking** on the **Format** menu.

The title of this dialog box depends on the data you select in the report. Here, data for Month was clicked, so the title is "Select Top/Bottom Lines"
3. The Select Top/Bottom Variable Name dialog box appears. It displays the options you can select for the ranking:

<table>
<thead>
<tr>
<th>To...</th>
<th>Do this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the largest $n$ values,</td>
<td>Click Top and select the value of $n$. (BusinessObjects selects 3 by default when you click Top.)</td>
</tr>
<tr>
<td>Select the smallest $n$ values,</td>
<td>Click Bottom and select the value of $n$. (BusinessObjects selects 3 by default when you click Bottom.)</td>
</tr>
<tr>
<td>Select the top $n%$ of values,</td>
<td>Click Top, click In percentage of total number of values, then select the value of $n$.</td>
</tr>
<tr>
<td>Select the bottom $n%$ of values,</td>
<td>Click Bottom, click In percentage of total number of values, then select the value of $n$.</td>
</tr>
<tr>
<td>Display subtotals relating to the values that appear in the report and the values that are omitted from the report,</td>
<td>Click Display subtotals.</td>
</tr>
<tr>
<td>Display percentages relating to the values that appear in the report and the values that are omitted from the report,</td>
<td>Click Display percentages.</td>
</tr>
<tr>
<td>Select the measure on which the ranking is based,</td>
<td>Choose the measure in the Based On combo box.</td>
</tr>
</tbody>
</table>

4. Click the ranking values you want to display in your report, then click OK or Apply. The report displays ranked data for the values you selected only. The Apply Ranking button on the toolbar is dimmed.

**Displaying subtotals**

When you click Display subtotals BusinessObjects adds the following calculations to the report:

- the sum of all the rows included in the report by the ranking
- the sum of all the rows excluded from the report by the ranking
- the overall sum of all the rows in these two categories
Displaying percentages

When you click Display Percentages BusinessObjects adds the following calculations to the report:

- each included row expressed as a percentage of all included rows
- the total number of included rows expressed as a percentage of all rows (included and excluded)
- the total number of excluded rows expressed as a percentage of all rows
- the total number of included and excluded rows expressed as a percentage (which is always 100%)

Ranking in master/detail reports

In master/detail reports, subtotals and percentages can only be shown for cells selected for ranking that are in the table. Master cells that have ranking applied will not display subtotals and percentages.

In master/detail reports, data is ranked for each section.
Ranking and breaks

In a table or crosstab in which breaks have been inserted, data is ranked separately for each break level.

NOTE

If you have created a local variable using values from different data providers, you will not be able to rank data based on this variable. The variable will not be displayed in the list in the Ranking dialog box.

EXAMPLE

Show sales revenue for top three months, compare with overall revenue

<table>
<thead>
<tr>
<th>Month Name</th>
<th>Sales revenue</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>September</td>
<td>$4,081,152</td>
<td>11.45 %</td>
</tr>
<tr>
<td>January</td>
<td>$3,440,910</td>
<td>10.65 %</td>
</tr>
<tr>
<td>March</td>
<td>$3,727,371</td>
<td>10.41 %</td>
</tr>
<tr>
<td>Sum</td>
<td>$11,659,433</td>
<td>32.12 %</td>
</tr>
<tr>
<td>Sum Other</td>
<td>$24,600,729</td>
<td>67.78 %</td>
</tr>
<tr>
<td>Sum All</td>
<td>$36,260,162</td>
<td>100.00 %</td>
</tr>
</tbody>
</table>

This report displays the top three revenue-making months. It shows the combined revenue for September, January and March (Sum) and the total revenue for the other nine months of the year (Sum Other).

TIP

To remove the ranking, click inside the ranked data, then click Apply Ranking.

Editing an existing ranking

Once you have defined a ranking by specifying the ranking values you want to display, you can edit it by selecting different ranking values. For example, if you have applied a ranking that enables you to view the top and bottom 3 ranking of revenue for customers, you can edit the ranking to show the top 10 instead.

To select different values for an existing ranking:

1. Click inside the data where the ranking was applied.
2. Click **Ranking** on the **Format** menu.
   The Select Top/Bottom **Variable Name** dialog box appears, with the current
3. Alter the ranking settings as desired, then click **OK** (or **Apply**). When you exit from the Dialog box, the values with the new ranking criteria are displayed.

**Removing a ranking**

When you remove a ranking, all the data that it excluded re-appears in the report. For example, if you applied a ranking that displays the top 10 customers by revenue, then you remove that ranking, the report displays the revenue for all customers.

To remove a ranking, first click inside the data on which you applied the ranking. You can then:

- Click the dimmed **Apply Ranking** button on the Report toolbar.
- Click **Ranking** on the **Format** menu, remove the Top and Bottom check marks, then click **OK** or **Apply**.

**Managing ranking with filters and sorts**

To rank data in a report, you must remove any sorts or filters currently applied to that data. If any sorts or filters exist when you try to apply a ranking, BusinessObjects displays the following message:

![Warning Conflict](image)

Click **Yes** to remove the existing sort or filter.
Hiding columns and rows of data

You can hide columns or rows of data so that the data is not displayed in your table but still remains in the report. To do this:

1. Right-click inside the table or crosstab.
2. Click Table or Crosstab on the Format menu.
3. Click the Pivot tab in the dialog box that appears.

**Hiding Data**

- In the Used Variables box, click the variable you want to hide, then click Hide.
- To hide more than one variable at the same time, hold down the Ctrl key, click the variables, then click Hide.

Hidden variables are grayed in the Used Variables box.

**NOTE**

When you hide a dimension, BusinessObjects does not recalculate measures. Hiding a dimension is not the same as removing it from a report.
Showing data
• To display a previously hidden variable, click it in the Used Variables list, then click Show.
Highlighting data

You can highlight data in a BusinessObjects report using alerters. Alerters use special formatting to make data that fits certain conditions stand out from the rest of the data. This helps draw attention to trends and exceptions in report data.

**EXAMPLE**

**Which sales representatives generate revenue over $500 000**

You want to identify excellent salespeople who generate revenue over $500 000, and poor salespeople who generate revenue below $50 000.

You define a condition for your good salespeople: "Revenue >= 500000" and define a format that will be applied for values that fit this condition: "Good job!" in green.

<table>
<thead>
<tr>
<th>Sales Person</th>
<th>Revenue</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fischer</td>
<td>$233 888.00</td>
<td></td>
</tr>
<tr>
<td>Galagher</td>
<td>$743 139.00</td>
<td>Good job!</td>
</tr>
<tr>
<td>Ishimoto</td>
<td>$544 883.00</td>
<td>Good job!</td>
</tr>
<tr>
<td>Nagata</td>
<td>$4,700.00</td>
<td>Work harder!</td>
</tr>
</tbody>
</table>

You then define a condition for your not-so-good salespeople: "Revenue <= 50000". and define a format that will be applied for values that fit this condition: "Work harder!" in red.

You insert an extra column after revenue and name it Performance. You then apply the alerter to the performance column.

**NOTE**

You cannot use alerters on charts.

**Creating an alerter**

There are 4 steps involved in creating an alerter:

1. Select the data you want to use.
2. Name and describe the alerter.
3. Define the range of values.
4. Define how the alerter will be displayed in the cell.

Once you have created an alerter you can apply it, hide it or display it.
Step 1: select the data
1. In a report, click a cell, row or column of data.
2. Click Alerter on the Format menu.
3. In the Alerter dialog box that appears, click Add.

Step 2: name and describe the alerter
1. In the Name box in the Definition tab, type a name for the alerter.
2. In the Description box, type a help text on the alerter to remind you and others what the Alerter is set to highlight.

3. Click the Conditions tab.
**Step 3: Set the conditions**

1. Choose a variable from the Variable to Compare listbox.

2. Choose an operator from the Operator 1 listbox.
   - Type a value (number, character or date), or
   - Click the down arrow button to the right of the Value 1 box, then click **Variables** or **List of Values**. Select a variable or a value.
3. If operator 1 requires the use of a second operator, click the operator to use in the Operator 2 box.
4. Click inside the Value 2 box and repeat step 2.

**Step 4: Set the formatting**

Specify different formats for each range, using fonts, colors and borders, for example. Or you can enter a text or select a variable that will mask the data that satisfies the conditions you set.

1. Click the arrow next to the Cell Content box.
   A shortcut menu appears.
2. To have the alerter display text, click **Text**.
3. Type the text, then click **OK**.
4. To have the alerter display a variable or a formula, click **Variables**.
5. To format the cell contents, click **Format**.
6. In the Cell Format box, format the text, then click **OK**.
7. Click **OK** to return to the Edit Alerter dialog box.
Switching alerters off and on

Once you have created alerters, you can apply or deactivate them whenever you like. For example, you can create an alerter, apply it to a report, print the report, then deactivate the alerter. In this case, the alerter appears on the printed report, but no longer appears on your screen.

▲ Turn an alerter on
1. Click the cell, column or row where you want to display the alerter.
2. Click Alerters on the Format menu.
3. Click the check box next to the alerter that you want to activate.
4. Click Apply or OK.

▲ Turn an alerter off
1. Click the cell, column or row where the alerter is displayed.
2. Click Alerters on the Format menu.
3. Click the check box next to the alerter that you want to deactivate.
4. Click Apply or OK.
Display or hide all alerters in the current report
You can also display or hide all alerters that you have applied to the current report.
1. Click Options on the Tools menu.
2. Click the Display tab.
3. Click Alerters under Report Options.
   If this option is already checked and you want to hide the alerters in your report, click it again to remove the check.
4. Click OK to apply the option and to close the dialog box.

Working with existing alerters
Once you have created an alerter, you can use it as the basis for other new alerters. You copy the existing alerter, then modify its conditions and their corresponding formats.

Copy an alerter
1. Click Alerters on the Format menu.
2. Click the alerter you want to copy.
3. Click Copy.
   The Edit Alerters dialog box appears.
   In the Definition tab, the name of the initial alerter appears in the Name box followed by no2. If you renamed your first alerter so its name is not the same as the variable name BusinessObjects gives the new alerter the variable name.
4. Type a new name and description for the new alerter.
5. In the Conditions tab, define the first range of values for the alerter.
6. Define the way the result appears in the cell.
7. Repeat Step 5 and Step 6 to define a second range of values if you wish, then click Apply or OK.

Edit an alerter
1. Click Alerters on the Format menu.
2. Click the alerter you want to edit.
3. Click Edit.
4. Change the definition of the alerter.
5. Click Apply or OK.
Delete an alerter
1. Click Alerters on the Format menu.
2. Click the alerter you want to delete.
3. Click Remove.
Customizing Queries on Universes
Overview

This chapter describes how to benefit from the most powerful query features in BusinessObjects. You learn how to
• create your own objects
• apply complex conditions
• work with multiple conditions
• combine the results of multiple queries into one data set
• view, edit and reuse the SQL generated by BusinessObjects queries
• use one query as input to another
Creating user objects

A universe consists primarily of classes and objects created by the universe designer. If the objects in a universe do not meet your needs, you can create your own additional user objects.

User objects appear in the User Objects class in the universe. You include them in queries in the same way that you include regular objects. You do not need to define a connection to a database to define a user object.

Why create a user object?

Based on one or more existing objects, user objects enable you to:

• make additional calculations beyond those provided by the base universe objects.
• apply functions to text, for example to capitalize data.
• group data.

Here’s an example of a user object.

**EXAMPLE**

**Obtaining total ordered revenue by creating a user object**

To obtain the revenue generated by an order, you create the Total user object with the following formula:

\[ \text{Total Revenue} = \text{Quantity Ordered} \times \text{Product Price} \]

where Quantity Ordered and Product Price are objects in your universe. When you include the Total user object in a query, BusinessObjects makes the calculation and places the results in the report.

What does a user object consist of?

A user object has a name, a type (character, date or numeric), a qualification (dimension, measure or detail) and a formula. The formula contains a combination of functions, objects, user objects, operators, and text.

User objects are end-user personal objects that are not shared with other end-users. User objects are defined for each universe and stored on a local file inside the “Universe” folder. For example, if you create a user object in the BEACH.UNV universe, BusinessObjects stores it locally in the file BEACH.UDO in the Universe folder.
What are the restrictions on user objects?

You can work only with the user objects that you create yourself, and you cannot move user objects from the User Objects class. Also, user objects are available only in the universe in which they were created.

User objects are not shared. Reports that include user objects can only be viewed by other end-users. This is because user objects are stored locally in a user object definition file. Other end-users, who do not have the same user object definition file, are not able to access the user object definitions. If an end-user tries to refresh or edit a query that contains another user's user objects, BusinessObjects removes the objects from the query and report.

Despite these restrictions, the universe designer can convert user objects into regular objects that can be made available in other universes and for other users.

**NOTE**

You cannot schedule reports that contain user objects and keep the user objects; they are removed when the report is refreshed. For more information refer to the *InfoView User's Guide*.

How can an end-user share user objects with other users?

If you want to share user objects with other users, you should ask the universe designer to include these user objects in the related universe in order to make them available to all BusinessObjects end-users.

Creating, editing and deleting user objects

BusinessObjects lets you create user objects in two different ways.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>create user objects that you can use only in the universe that contains the current query,</td>
<td>click <strong>User Objects</strong> on the Query Panel toolbar. You cannot delete user objects this way.</td>
</tr>
<tr>
<td>create, edit, or delete user objects in any of the universes available to you,</td>
<td>click <strong>Universes</strong> on the <strong>Tools</strong> menu.</td>
</tr>
</tbody>
</table>

The following sections describe how to create, edit and delete user objects using the Universes command.
To create a user object in any universe available to you

1. Click Universes on the Tools menu.
   The Universes dialog box appears.
2. Select the universe in which you want to create the user object, then click User Objects.
   The User Objects dialog box appears.
3. Click Add.
   The User Object dialog box appears:

4. Type the name of the user object in the Name field in the Definition tab.
5. Select the type of the user object (Character, Number or Date) in the Type list.
6. Type a description of the object in the Type list.
   The descriptive text appears when you select the user object in the Query Panel.
7. Under Qualification, click Dimension, Measure or Detail.
8. Click the Formula tab to write the formula for the user object.
   For information on how to write the formula, refer to To write the formula of a user object on page 334.
9. Click OK.
   The user object you have created appears in the User Objects dialog box.
next time you build or edit a query on the universe, the user object you have created will appear in the User Objects class.

REMINDER

You can create a user object by clicking User Objects on the Query Panel toolbar. Using this method you can create user objects only in the universe you selected for the query you are working on. The workflow is the same as the one described in this section.

To write the formula of a user object

1. Display the User Object dialog box by following steps 1 to 3 under To create a user object in any universe available to you on page 333.
2. Click the Formula tab, then check Get Assistance on Functions:
3. Double-click the objects, functions and operators you want to use in the user object formula.
   - When you double-click an object or an operator, it appears in the Formula box. The functions and operators you can select depend on the database at your site.
   - When you double-click a function, the Function[FunctionName] dialog box
4. In the Function[FunctionName] dialog box, type an argument in each field.
   - If the function's arguments include objects, user objects, functions or operators, you can double-click them to insert them in the function formula.
   - The arguments appear in the Formula box.
5. If necessary, type text, numbers or dates in the formula.
   You must type quotes (") before and after text and numbers, but type a single quote (') before and after dates.
6. Click OK, then click Test to check the syntax of the formula.

To edit a user object

Editing a user object enables you to change the object’s name, type, and definition (formula). To edit a user object:

1. Click Universes on the Tools menu.
   The Universes dialog box appears.
2. Select the universe that contains the user object that you want to edit, then
click **User Objects**. The User Object dialog box appears.
3. Select the user object you want to edit, then click **Edit**.
4. In the Definition tab of the User Objects editor, change the name, the type and/or the help text of the user object.
5. In the Qualification box, click a radio button to change the user object qualification.
6. Click the **Formula** tab if you want to edit the user object’s formula.
7. Click **OK**.

**To delete a user object**
1. Click **Universes** on the **Tools** menu.
2. In the Universes dialog box, select the universe that contains the user object you want to delete, then click **User Objects**.
3. In the User Objects dialog box, select the user object you want to delete, then click **Delete**.

**Creating a time hierarchy for a user object**
When one of your user objects is a date-type dimension, you can create a time hierarchy for the object.

1. In the User Object dialog box, make certain that the object is a date-type dimension.
2. Click **Automatic Time Hierarchy**. The Automatic Time Hierarchy dialog box appears.
3. In the Automatic Time Hierarchy dialog box, click **Year, Quarter, and/or Month**. When you click Year, Quarter, and/or Month, you create a new user object that will appear below the initial user object in the hierarchy.
4. If you want, enter a name and help text for each new user object in the Name box and Description box respectively, then click **OK**.
Applying complex conditions on queries

You can limit the data that queries return by applying conditions. You apply complex conditions by combining an object with an operator (for example greater than), and an operand (for example, values that you type, or another object).

In addition to complex conditions, you can use predefined conditions and simple conditions. Here are some guidelines for choosing among the three types of conditions:

<table>
<thead>
<tr>
<th>If you want...</th>
<th>Then use...</th>
</tr>
</thead>
<tbody>
<tr>
<td>to use conditions defined by the universe designer using Designer that you can apply to a query, but that you cannot edit or delete from a universe,</td>
<td>predefined conditions.</td>
</tr>
<tr>
<td>to use conditions that you apply by selecting values from an object list of values and that let you limit the data returned only by result objects,</td>
<td>simple conditions.</td>
</tr>
<tr>
<td>to use conditions that let you choose the exact operator and operand on any object in the universe,</td>
<td>complex conditions.</td>
</tr>
</tbody>
</table>

To illustrate how you can benefit from complex conditions, here’s an example.

**EXAMPLE**

**Which customers made reservations for 2001 and 2002?**

You want to market new products to customers who made reservations for 2001 and 2002. All you need is that list of names - and you obtain it by applying a complex condition on Reservation Year, without using Reservation Year as a result object in the query. Here’s how to do it:

1. Insert the Customer object in a query on the Island Resorts Marketing universe.
2. Drag the Reservation Year object to the Conditions box and click the <select...
an operator> text next to the object name.
The Classes and Objects list turns into the Operators list.
3. Double-click the In list operator.
The Operators list turns into the Operands list.
4. Double-click the Show list of values operand.
The List of Values of Reservation Year dialog box appears.

6. Click OK
7. Click Run.
The list of customers appears in the report.

Applying a complex condition on a query
Applying a complex condition requires three steps. First, you select the object you want, then the operator (for example, greater than), then the operand (for example, values that you type, or another object). The following procedure explains how to do it, and gives information to help you choose the operator and operand you need:
1. Drag the object you want to use from the Classes and Objects list to the Conditions box in the Query Panel.
The Classes and Objects list becomes the Operators list:
The following table helps you to select the operator you need:

<table>
<thead>
<tr>
<th>To obtain data that...</th>
<th>For example...</th>
<th>Double-click...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is equal to one given value</td>
<td>A particular Year</td>
<td>Equal to</td>
</tr>
<tr>
<td>Is different from one given value</td>
<td>Countries not including France</td>
<td>Different from</td>
</tr>
<tr>
<td>Is greater than a given value</td>
<td>Revenue over $100,000</td>
<td>Greater than</td>
</tr>
<tr>
<td>Is greater than or equal to a given value</td>
<td>Customers who are 60 or over</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>Is lower than a given value</td>
<td>Revenue under $100,000</td>
<td>Less than</td>
</tr>
<tr>
<td>Is lower than or equal to a given value</td>
<td>Customers who are 60 or under</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>Falls between two given values</td>
<td>Weeks between 25 and 36</td>
<td>Between</td>
</tr>
<tr>
<td>Falls outside two given values</td>
<td>All the weeks of the year excluding weeks 25 to 36</td>
<td>Not between</td>
</tr>
<tr>
<td>Is the same as any of a list of values</td>
<td>Revenues from only two resorts</td>
<td>In list</td>
</tr>
<tr>
<td>Is different from given values</td>
<td>Non-European customers</td>
<td>Not in list</td>
</tr>
<tr>
<td>Contains empty rows</td>
<td>Customers who have not paid (i.e., without invoice dates)</td>
<td>Is null</td>
</tr>
<tr>
<td>Does not contain empty rows</td>
<td>Customers who have paid (i.e., their invoice dates are in the database)</td>
<td>Is not null</td>
</tr>
<tr>
<td>All contains the same letter or letters</td>
<td>Customers whose names begin with the letters S</td>
<td>Matches pattern</td>
</tr>
</tbody>
</table>
2. Double-click the operator you want to use.
The Operators list turns into the Operands list:
The operands in the list depend on the operator you selected.

3. Double-click the operand you want.
The following table helps you select the operand you need and tells you what to do next:

<table>
<thead>
<tr>
<th>To obtain data that...</th>
<th>For example...</th>
<th>Double-click...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not contain a given letter or pattern of letters</td>
<td>Customers whose names do not begin with S</td>
<td>Different from pattern</td>
</tr>
<tr>
<td>Satisfies two conditions on one object</td>
<td>Customers who settled an invoice in June and in July</td>
<td>Both</td>
</tr>
<tr>
<td>Excludes a given value</td>
<td>Customers who stayed at resorts other than French Riviera</td>
<td>Except</td>
</tr>
</tbody>
</table>

If you want to compare the condition object with...

<table>
<thead>
<tr>
<th>Double-click...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values that you type</td>
<td>Type a new constant</td>
</tr>
<tr>
<td>Values that you select from the object’s list of values</td>
<td>Show list of values</td>
</tr>
<tr>
<td>Values that you will select when you run the query</td>
<td>Type a new prompt - or - Show list of prompts</td>
</tr>
</tbody>
</table>

Select a prompt from the dialog box that appears.
### Tips for applying complex conditions

This section provides some tips on getting the most out of BusinessObjects complex conditions.

**Using wildcard characters**

Conditions with the *Match pattern* and *Different from pattern* operators are great for finding lists of similar values, such as customer names beginning with S.

<table>
<thead>
<tr>
<th>If you want to compare the condition object with...</th>
<th>Double-click...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Another object (which can be a user object)</td>
<td>Select an object</td>
<td>Double-click the object in the Classes and Objects box.</td>
</tr>
<tr>
<td>Any value returned by another query</td>
<td>Create a subquery (ANY)</td>
<td>Build a query in the new query tab that appears. For more information, refer to One invoice date per customer appears in the report. on page 346.</td>
</tr>
<tr>
<td>All values returned by another query</td>
<td>Create a subquery (ALL)</td>
<td>For more information, refer to Applying a condition with a calculation on page 344.</td>
</tr>
<tr>
<td>The result of a calculation (sum, minimum, maximum, average or count)</td>
<td>Calculation</td>
<td>Follow the screens of the wizard that appears. For more information, refer to Applying a condition with a calculation on page 344.</td>
</tr>
<tr>
<td>The values returned by an existing query</td>
<td>Select Query Results</td>
<td>Follow the procedure described in Using an existing query in a condition on page 356.</td>
</tr>
</tbody>
</table>
Wildcards are special characters that can denote any single character, or any number of characters. BusinessObjects supports the standard wildcard characters, which are:

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>Replaces several characters, or in the response to a prompt. For example, N% returns all values beginning with an N (New York, Nevada)</td>
</tr>
<tr>
<td>_</td>
<td>Replaces a single character in a constant. For example, GR_VE returns Grave, Grove, Greve.</td>
</tr>
</tbody>
</table>

**Using In list**

The *In list* operator lets you select multiple values for a document. These multiple values can be a condition on a query that you want to build or can be the basis for an interactive document in which BusinessObjects prompts other users to select values from the list you created to view data limited to their needs.

You type your list of values in the text field of the Enter or Select Values dialog box or if you click Values in this dialog box you can select them from a list. When you type values, separate each value with a comma (,).

The maximum number of values allowed in a list is 256.

**Using Different From, Not in list and Except**

*Different from*, *Not in list* and *Except* are all operators that exclude certain data from your query results. Does this mean that you could use this condition to obtain a list of customers who have not stayed at Bahamas Beach?

In fact, you cannot. In situations like this you need to think carefully about the query you are building and how your data is structured. The result of this query includes those customers who have stayed at Bahamas Beach and elsewhere. Why? Because reservations exist for these customers for resorts other than Bahamas Beach. These reservations alone are enough to satisfy the condition ‘Resort differs from Bahamas Beach’.

Furthermore, this condition excludes customers who have made no reservations. BusinessObjects checks these customers’ records against reservations and determines that no reservations satisfy the condition ‘Resort differs from Bahamas Beach’.
Applying complex conditions on queries

Bahamas Beach’ - because there are no reservations! Nevertheless, it is clear that a report showing customers who have not stayed at Bahamas Beach should include customers who have not stayed anywhere.

You solve this problem by using the Except operator instead of Different from. When you use Except, BusinessObjects builds two queries:

• All customers.
• Customers who have reservations for Bahamas Beach.

BusinessObjects then subtracts the customers given by the second query from those given by the first. This returns the result you want.

Note also that:

• You can only specify one value with Different from, but multiple values with Not in list.
• You can only specify one value with Except. However, you can build combined queries using MINUS to exclude, for example, Bahamas Beach customers and 1996 customers.

For more information, refer to Building combined queries on page 365.

More on Not in List

You should take care when using the Not in list operator. In this type of query each record in the outer query must be checked against every record in the inner query (the list referenced by Not in list) to determine whether it should appear in the report. If the inner list is small, this is not a issue. If it is large, it is. Why? What if you set up a condition along the lines of ‘Customers who are not in the list of customers who own more than two cars’, and your database contains 10,000 customers? The query needs to check 10,000 * 10,000 rows (that’s 1,000,000,000 rows!) to generate the report.

Editing complex conditions

You can edit a complex condition by changing its object, operator and/or operand. To do this:

1. Click the part of the condition that you want to change in the Conditions box of the Query Panel.
2. Depending on the element you clicked, select a different element in either the Classes and Objects list, or the Operators list, or the Operands list.

NOTE

If you use a different operator, you might also have to use a different operand.
Deleting complex conditions
1. Right-click the condition’s icon in the Conditions box.
2. Click Delete on the shortcut menu.

Applying a condition with a calculation
BusinessObjects lets you limit query results with calculations in complex conditions. This type of condition is useful in answering questions such as “Which products generated above average revenue?” at the query level.

You apply a condition with a calculation by using the Calculation operand. BusinessObjects then displays a wizard which guides you through the steps required to make the calculation.

To apply a condition with a calculation
Use the following procedure to apply a condition with a calculation.
1. Move an object to the Conditions box in the Query panel. The Classes and Objects box becomes the Operators box.
2. Double-click Equal to.
   The Operators box becomes the Operands box.
3. Double-click Calculation.
4. The Complex Condition wizard appears. The first screen asks you to select a calculation object.
5. Open the folder containing the object, click on the object, then click Begin. The next screen asks you to select a function to apply on the object.
6. Select the function from the list, then click Next. The next screen asks you to define the level of calculation.
7. Select a level of calculation.

<table>
<thead>
<tr>
<th>If...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>you want to obtain a single result row,</td>
<td>click Globally, then click Next.</td>
</tr>
<tr>
<td>you want to obtain several result rows,</td>
<td>click By one or more objects, select the</td>
</tr>
<tr>
<td></td>
<td>objects from the list, then click Next.</td>
</tr>
</tbody>
</table>

The next screen asks you to choose between making an independent calculation and comparing the result of the calculation with the values of one
or more objects.

8. Select how you want to synchronize your calculation.

<table>
<thead>
<tr>
<th>If...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>you want to make a calculation independently of your objects,</td>
<td>click <em>Independently of your objects</em>, then click <em>Next</em>.</td>
</tr>
<tr>
<td>you want to make a calculation for each value of one or more of your objects, which allows you to limit the calculation to particular objects,</td>
<td>click <em>For each value of one or more objects</em>, select the objects from the list, then click <em>Next</em>.</td>
</tr>
</tbody>
</table>

The next screen asks you to set the number of values to compare.

9. Select whether you want to compare the object with at least one value or with all values.

<table>
<thead>
<tr>
<th>If...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>you want to compare the object with at least one value, which allows you to limit the values compared with the object,</td>
<td>click <em>At least one value</em>, then click <em>Finish</em>.</td>
</tr>
<tr>
<td>you want to compare the object with all values,</td>
<td>click <em>All values</em>, then click <em>Finish</em>.</td>
</tr>
</tbody>
</table>

The Query Panel reappears with the query defined with a condition on a calculation.

10. Click *Run*.

**EXAMPLE**

*When did each customer last pay for a product?*

You want to find out the date of each customer’s last invoice so that you can contact those customers who have not been buying your products. To obtain this data, you need to apply a complex condition with a calculation. The calculation compares the invoice dates for each customer, then returns only the last date. Here’s how to do it:

1. Insert the *Customer* and *Invoice Date* objects in a query on the Island
Resorts Marketing universe.

2. Drag the **Invoice Date** object to the Conditions box.
3. Double-click the **Equal to** operator.
4. Double-click the **Calculation** operand.

The Complex Condition Wizard appears:

5. Open the Sales class, click the Invoice Date object, then click **Begin**.
6. Click **Maximum** (you want the last invoice date), then click **Next**.
7. In the next dialog box, click **Globally**, then click **Next**.
8. In the next dialog box, click **For each value of one or more objects**, then **Customer**.

This option forces the calculation to return the last invoice date per customer.

9. Click **Next**, click **Next** again, then, in the Query Panel, click **Run**.

One invoice date per customer appears in the report.

Let's look in detail at the calculation you have just created in detail to make what is happening clear.

When you use a calculation, BusinessObjects builds SQL that contains a subquery. A subquery is an inner query. The database that receives the SQL generated by BusinessObjects evaluates the result of the inner query against each row of the outer query to determine if the row should appear in the result.

**NOTE**

You can create subqueries explicitly. See Applying a condition with a subquery on page 352.

As you move through the wizard, you specify:
• the object to use in the calculation

This is the object in the outer query whose value is compared against the result of the inner query. In the example, the object is Invoice Date.

• the aggregate function to apply to the object.

In the example you applied the Maximum function because you were...
interested in the most recent invoice date.
• The level of calculation.

This determines the grouping in the subquery. In the example you chose Globally because you were interested simply in the customer’s latest invoice date, not a latest invoice date by some other criteria.
- Synchronization..

This determines the subquery links to the main query in the example you chose the Customer object because you were interested in each customer’s latest invoice date.
• The number of values to compare.

This determines how many values in the subquery the database compares against the values in the outer query. In the example you can choose either option because the subquery returns one row only for each customer.
Examining the SQL

The following figure shows and explains the SQL generated by BusinessObjects from the calculation in the example.

a. Invoice Date is the object used in the calculation, so the SQL joins the two queries using the Sales.Invoice_date field.
b. The calculation level is global so the sub-query has no grouping.
c. The queries are synchronized via the Customer object so the SQL creates a correlated subquery on Customer.last_name.
d. The query compares all the values in the inner query against the outer query.

Applying a condition with a subquery

A subquery is a query within a query. It returns a single column of data which is compared with the data retrieved by the main query. You use subqueries for situations such as finding a single individual in a list of individuals who meet the conditions of the query. For example, of all customers who made reservations, what is the name and address of the customer who made the first reservation?

Complex conditions on queries consist of three elements: an object, an operator and an operand. Conditions that include the Create a subquery (ALL) operand, or the Create a subquery (ANY) operand, generate a subquery. The operator (for
example greater than, less than) that you include in the condition determines whether the data returned by the subquery is, for example, excluded from the main query result.

The operands that generate a subquery are described here:

<table>
<thead>
<tr>
<th>This operand...</th>
<th>Compares...</th>
<th>And answers questions such as...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a subquery (ALL)</td>
<td>All the values returned by the subquery with the values returned by the main query</td>
<td>Which customers have not bought my latest product?</td>
</tr>
<tr>
<td>Create a subquery (ANY)</td>
<td>Any values returned by the subquery with the values returned by the main query</td>
<td>Which products generated above average revenue?</td>
</tr>
</tbody>
</table>

**To apply a condition with a subquery**

Here’s how to apply a condition with a subquery.

1. Drag an object from the Classes and Objects box and drop it in the Conditions box in the Query Panel.
2. Double-click the operator you want to use.
   - Some operators (for example Both, Between, Match pattern) cannot be used with subqueries.
   - For information on which operator to choose, refer to the table on page 339.
3. Double-click the Create a subquery (ALL) operand or the Create a subquery (ANY) operand.
   A tab is created for the subquery (Subquery 1.1).
4. In the Subquery 1.1 tab, insert an object in the Result Objects box.
   Note that you can only include one result object in the subquery.
5. If necessary, apply a condition on the subquery.
6. Click Run.

**NOTE**

The subquery’s tab always appears to the right of the main query. A number appears in the tab, (Subquery n.n). BusinessObjects increments the subquery number to show the relationship between a query and its subquery or subqueries.
EXAMPLE

Which customer made the earliest reservation?

You want to offer a bottle of champagne to the customer who made the earliest reservation. To obtain the name and address of this customer:

1. Insert the Customer object and the Address object in a query on the Island Resorts Marketing universe.
2. Drag the Reservation Date object to the Conditions box.
3. Double-click the Less than or equal to operator.
4. Double-click the Create a subquery (ALL) operand.
   The Subquery 1.1 tab now appears in the Query Panel.
5. In the Subquery 1.1 tab, insert the Reservation Date object in the Result Objects box.
   The Query Panel now looks like this:

6. Click Run.
   The name and address of the customer who made the earliest reservation appears in the report:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schultz</td>
<td>Am Holzbach 17</td>
</tr>
</tbody>
</table>
Deleting a subquery

A subquery is generated by a condition containing the \textit{Create a subquery (ALL)} operand or the \textit{Create a subquery (ANY)} operand. Therefore, you delete a subquery by deleting the condition in the main query.

To delete the condition:
1. Right click the condition icon in the Conditions box.
2. Click \textbf{Delete} on the shortcut menu.

Subqueries and calculations

The example \textit{When did each customer last pay for a product? on page 345} explains that complex conditions generate subqueries behind the scenes. You can see this by repeating the example above using a complex condition with a calculation.

\textbf{EXAMPLE}

\textbf{Which customer made the earliest reservation? (using calculation)}

To discover this using a complex condition with a calculation:
1. Insert the Customer and Address objects in a query on the Island Resorts Marketing universe.
2. Drag the Reservation Date object to the Conditions box.
3. Double-click the \textit{Equal to} operator.
4. Double-click the \textit{Calculation} operand.
   
   The Complex Condition wizard appears.
5. Select the Reservation Date object then click \textbf{Begin}.
6. Select the Minimum function then click \textbf{Next}.
   
   The Minimum returns the earliest (smallest) reservation date.
7. Select \textit{Globally} then click Next.
8. Select \textit{Independently of your objects} then click Next.
   
   Note that you are interested in the earliest overall reservation date so you do not link the date to an object in the main query.
9. Select \textit{All Values} then click \textbf{Finish}.
10. Run the query.

The query returns the same result as the query defined using a subquery to answer the same question. Examine these steps in relation to the description of complex conditions in the example \textit{When did each customer last pay for a product? on page 345} if you are still unclear why this is so.
Using an existing query in a condition

You can use the values returned by an existing query in a condition in another query. This is similar in some ways to creating a subquery. (You can express the same queries either as subqueries or as queries that use values returned by existing queries). The difference here is that BusinessObjects does not build an SQL statement containing a subquery. It returns the data from both queries, then performs the 'subquery' processing on the client machine.

EXAMPLE

Return list of resorts/revenues where resort country revenue > $1000000

In this example you have a report containing a data provider that lists all countries whose revenue is greater than or equal to $1,000,000. You can use this data provider to build the new data provider that lists all resorts within these countries and the resort revenues. To do this:

1. Click Table on the Insert menu.
2. With your mouse, draw a rectangle where you want the new block to appear.
3. When you release the mouse button, the New Table wizard appears.

4. Click *Build a new query on the universe currently in use*. The Query Panel appears.
5. Drag the Resort and Revenue objects to the Result Objects window.
6. Drag the Country object to the Conditions window.
7. Double-click *In list* in the list of operators. The list of operators changes to a list of operands.
8. Click *Select Query Results* in the list of operands.
The list of data providers in the document appears.

9. Expand the data provider and select the Country dimension.

10. Click **OK**. The condition appears in the Conditions window.
11. Click **Run** to run the query.

The data appears in a block in the report.

<table>
<thead>
<tr>
<th>Resort</th>
<th>Revenue(Resort)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>971,444.00</td>
</tr>
<tr>
<td>Hawaiian Ckt</td>
<td>1,479,660.00</td>
</tr>
</tbody>
</table>
Applying groups of conditions

You use conditions to limit the data retrieved by queries. A group of conditions consists of two or more conditions (predefined, simple or complex) applied on the same query.

You use groups of conditions when a single condition does not enable you to obtain the data that you need.

Organizing groups of conditions

A group of conditions consists of two or more conditions applied on the same query. In the Conditions box in the Query Panel, conditions are linked by an operator (AND or OR).

- **Groups of two conditions**
  In a group that contains only two conditions, you double-click the operator to change it from AND to OR, or vice versa.

- **Groups of three or more conditions**
  When you double-click the operator in a group that contains at least three conditions, you create an indentation in the group, as illustrated here:

  ![Indentation Example]

  In groups that contain three or more conditions, you can also:
  - Create indentations without replacing the operator, by dragging the operator horizontally.
  - Move conditions within the group.
  - Delete a condition from a group.
The following table explains how to organize groups of conditions in different ways:

<table>
<thead>
<tr>
<th>To...</th>
<th>Do this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change a group's operator from AND to OR</td>
<td>Double-click the AND or OR operator.</td>
</tr>
<tr>
<td>and vice versa</td>
<td></td>
</tr>
<tr>
<td>Indent or outdent conditions within a</td>
<td>Click the AND or OR operator with your right mouse button.</td>
</tr>
<tr>
<td>group of conditions</td>
<td>Click <strong>Shift right</strong> or <strong>Shift left</strong> on the shortcut menu</td>
</tr>
<tr>
<td>Move a condition from one group to another</td>
<td>Click the condition and hold down your mouse button.</td>
</tr>
<tr>
<td></td>
<td>Drag the condition to another group of conditions, then release your mouse button.</td>
</tr>
</tbody>
</table>

**AND and OR**

BusinessObjects automatically links multiple conditions using operators in the Conditions box in the Query Panel:

- **AND** specifies a result that is true for both conditions.  
  - For example, the following group of predefined conditions on a query containing the Customer object specifies customers who stayed at Bahamas Beach in 1999:

  ![AND example](image1.png)

- **OR** specifies a result that is true for either the first or the second condition.  
  - For example, the following group of predefined conditions on a query containing the Customer object specifies any customers from 1998 and any customers who stayed at Bahamas Beach:

  ![OR example](image2.png)
You can replace AND with OR, and vice versa, by double-clicking it. When you apply three or more conditions on a query, double-clicking the operator creates an indentation in the group of conditions.

**TIP**

Avoid groups of conditions such as *Year Equal to 2001 AND Year Equal to 2002*. This example would return no data, because *Year* cannot be equal to two different values. To obtain, for example, the list of customers from both 2001 and 2002, you would have to build a combined query using the INTERSECT operator. For information on combined queries and how to build them, refer to Building combined queries on page 365.

**Order of precedence**

When you have a group of conditions, BusinessObjects objects evaluates them in order. For example, in the following list of conditions, BusinessObjects first determines whether the sales revenue \( \geq 2000000 \) or the quantity sold is \( > 1000 \). Only after it has evaluated whether either of these conditions are true does it compare this evaluation with the condition that checks whether the margin \( \geq 1000000 \).

BusinessObjects shows this order of precedence by indenting conditions that are evaluated first. and returns the following list of stores that satisfy these conditions:

<table>
<thead>
<tr>
<th>Store name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 e-Fashion Austin</td>
</tr>
<tr>
<td>2 e-Fashion Chicago 33rd</td>
</tr>
<tr>
<td>3 e-Fashion Highlands</td>
</tr>
<tr>
<td>4 e-Fashion Los Angeles</td>
</tr>
<tr>
<td>5 e-Fashion New York Magnolia</td>
</tr>
<tr>
<td>6 e-Fashion New York Sundance</td>
</tr>
<tr>
<td>7 e-Fashion San Francisco</td>
</tr>
<tr>
<td>8 e-Fashion Washington Tower</td>
</tr>
</tbody>
</table>
However, the following group of conditions returns the following list of stores:

<table>
<thead>
<tr>
<th>Store name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 e-Fashion, Austin</td>
</tr>
<tr>
<td>2 e-Fashion, Chicago 33rd</td>
</tr>
<tr>
<td>3 e-Fashion, Colorado Springs</td>
</tr>
<tr>
<td>4 e-Fashion, Houston 5th</td>
</tr>
<tr>
<td>5 e-Fashion, Houston Heights</td>
</tr>
<tr>
<td>6 e-Fashion, Los Angeles</td>
</tr>
<tr>
<td>7 e-Fashion, New York Magnolia</td>
</tr>
<tr>
<td>8 e-Fashion, New York Sundance</td>
</tr>
<tr>
<td>9 e-Fashion, San Francisco</td>
</tr>
<tr>
<td>10 e-Fashion, Washington, Telbooth</td>
</tr>
</tbody>
</table>

because BusinessObjects now determines which stores have a quantity sold $\geq 10,000$ or margin $\geq 1,000,000$, and then determines which of these has sales revenue $\geq 2,000,000$.

**To apply groups of conditions**

1. In the Query Panel, apply a condition (predefined, simple or complex).
2. Apply a second condition. The conditions are automatically linked by the AND operator.
3. Apply more conditions if necessary.

**EXAMPLE**

Which customers bought a given product in a given time period?

You need the list of customers who stayed at the Bahamas Beach resort in 2001 or 2002. This requires two conditions: \textit{Resort=Bahamas Beach} and \textit{Year = 2001 or 2002}. Here’s how to do it:

1. Include the Customer object in a query on the Island Resorts Marketing.
2. Click *Predefined Conditions* below the Classes and Objects box. The list of predefined conditions in the universe appears.

3. Click the + sign to the left of the Resort class, and double-click *Bahamas resort*.

4. Click the + sign to the left of the Sales class, then double-click *Year 2001*.

5. Double-click *Year 2002*.
   The conditions are linked by an AND operator.

6. Double-click the AND that links *Year 2001* with *Year 2002*.
   BusinessObjects changes AND to OR, and indents the group of conditions.

7. Click **Run**.

**Deleting groups of conditions**

1. Click the operator (AND or OR) that links the group of conditions that you want to delete.

2. Click your right mouse button.

3. Click **Delete** on the shortcut menu.
Building combined queries

BusinessObjects lets you combine the data returned by up to eight queries as one set of results. These combined queries enable you to:

• obtain a single column of data from multiple objects
• obtain data common to two sets of results, such as customers from a given region and a given age group
• exclude the results of one query from the results of another

NOTE

Building combined queries allows you to combine or exclude data from the query result by using operators to combine the results of multiple queries. You can contrast this with using the In list and Not in list operators when you apply a complex condition on a query. The In list and Not in list operators include or exclude data from a query result based on a list of values that you enter.
Building a combined query

Here’s how to build a combined query:

1. In the Query Panel, build a query.

2. Click the Combine Queries button on the Query Panel toolbar.
   - The existing query appears in the Query 1 tab.
   - A second tab, Query 2, also appears and is now active:

3. If you want to use a different operator, click the Query 2 tab with your right-mouse button, then click the operator you want on the shortcut menu:
The following table describes the options:

<table>
<thead>
<tr>
<th>Use...</th>
<th>To...</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNION</td>
<td>Combine the data from two objects in a single column in the report. UNION is especially useful for working with incompatible objects. For example, you need to find out the dates on which your customers made reservations or paid their invoices. This query requires two incompatible objects: Invoice Date and Reservation Date. If you include these objects in a regular query, BusinessObjects returns two blocks of data. By building a combined query with UNION, with Invoice Date in one tab and Reservation Date in the other, the data appears in one column in the report. UNION is the default operator. Here is the symbol for the UNION operator:</td>
</tr>
<tr>
<td>INTERSECT</td>
<td>Obtain data common to two sets of results, such as customers from a given region and a given age group. The example, Which customers bought a given product in a given time period? on page 363, illustrates the use of INTERSECT. Here is the symbol for the INTERSECT operator:</td>
</tr>
<tr>
<td>MINUS</td>
<td>Exclude the results of one query from the main query result. For example, you can use MINUS to find out which customers bought product A but not product B. You cannot obtain this data with a condition such as Product different from B, because the condition would include customers who bought A and B. Here is the symbol for the MINUS operator:</td>
</tr>
</tbody>
</table>
4. Build the rest of the query in the Query 2 tab.
5. If necessary, repeat the above steps to include more queries.
   You can include up to eight queries in a combined query.
6. Click Run.
   The data from the combined query appears in the report.

**NOTE**
You can delete one of the queries you have combined by clicking its tab with your right-mouse button, then by clicking the **Delete** command on the pop-up menu that appears.

---

**Restrictions on combined queries**

You need to be aware that:
- Queries that you combine must contain the same number of objects in order to return the same number of columns of data.
- These objects must be of the same data type.
- You can include up to eight queries in a combined query.

To illustrate how you can benefit from combined queries, the following example explains how to obtain data common to two sets of results.

**EXAMPLE**

**Which customers bought products in both 2001 and 2002?**

Finding customers who match two criteria is a common business goal. You cannot obtain the list of paying customers from two years by
- Applying a complex condition with the **In list** operator; in this case, you would obtain customers from either 2001 or 2002, or possibly both.
- Two conditions linked by AND (**Year equal to 2001** and **Year equal to 2002**) because this will return no data.
- Two conditions linked by OR (**Year equal to 2001** or **Year equal to 2002**) because this does not guarantee that the customer bought in both years.

You need to build a combined query that returns the intersection of customers from 2001 and 2002. Here's how to do it:
1. Include the Customer object in a query on the Island Resorts Marketing
universe.

2. Drag the Year object to the Conditions box.

3. Double-click Equal to, then Type a new constant, then type 2001.

4. Press Enter, then click Combine Queries on the Query Panel toolbar.
   - The existing query appears in the Query 1 tab.
   - A second tab, Query 2, also appears and is now active.
   - Customer is already a result object in Query 2.

5. In the Query 2 tab, drag the Year object to the Conditions box.

6. Double-click Equal to, then Type a new constant, then type 2002.

7. Press Enter, then click the Query 2 tab with your right-mouse button.
   The operators you can use to combine the queries appear on a pop-up menu:
   - Union
   - Intersect
   - Minus
   - Delete

8. Click Intersect.
   The INTERSECT symbol appears on the Query 2 tab as shown here:

9. Click Run.
   A list of customers who bought products in both years appears in the report.

Using SQL from BusinessObjects queries

When you build a query in the Query Panel, BusinessObjects writes the query’s corresponding SQL. If you know SQL and want to find out how BusinessObjects resolves a query, you can view the SQL script.

You can also reuse the SQL that BusinessObjects generates by saving it to a file. And because BusinessObjects also lets you edit your query SQL scripts, you can build complex queries in the Query Panel, change the SQL to suit your needs, then save the script. You can then run the saved script using another application.
To view, edit or save a query’s SQL script:

1. Click the **View SQL** on the Query Panel toolbar. The query’s SQL script appears in the SQL Viewer dialog box:

   ![SQL Viewer](image)

   - **NOTE**
   
   It is possible that BusinessObjects will have created a complex SQL query to resolve the query, for example if the query contains incompatible objects or certain types of conditions. Such SQL queries can be split into several SELECT statements, which BusinessObjects indicates as folders in the left pane of the SQL Viewer dialog box.

2. To edit the script, click inside it and type the changes you want.
3. Click **Regenerate** to go back to the SQL of the original query.
4. Check **Do Not Generate SQL before running** if you want to keep the changes you have made.
   - This option automatically parses the script when you click **OK**.
   - If you do not click this option, any SQL changes you have made will be lost when you click **OK**.
5. Click **Parse** to check the validity of the script.
NOTE

If the universe designer has set up the universe with a restrictive connection, BusinessObjects prompts you to enter your database username and password before parsing the script. For more information on restrictive connections, see Restrictive connections on page 40.

6. Click Save to save the edited script to a file.
7. Click OK to return to the Query Panel.
Customizing Queries on Universes
Using and Customizing Lists of Values
Overview

This chapter describes lists of values (LOVs) and how you can use them to make your reports more efficient.
What is a list of values?

A list of values contains the values returned by an object. You use lists of values to select the value(s) you need when defining conditions on a query or when selecting the value(s) in a prompt.

When you use or view a list of values for the first time, BusinessObjects creates a .lov file that contains the query definition and the values it returns. By default, .lov files are located in sub-folders inside the UserDocs folder.
How are lists of values created?

In Designer, the universe designer decides whether to associate a list of values with an object. Once associated, the list of values can be viewed or edited in Designer or BusinessObjects.

The first time you view an object’s list of values, BusinessObjects runs a query and retrieves the values from the database; the object’s default list of values is generated by the object query.
Customizing lists of values in BusinessObjects

You can customize lists of values in the following ways:

- By editing the list’s corresponding query.
  For example, you can limit the data returned by a list of values by applying a condition.
- By assigning data from personal data files.
  This feature is especially useful if you always use the same subset of the values available in the database when applying conditions. You can view only the values you need without having to connect to your remote database. For more information, refer to Assigning personal data to a list of values on page 380.
Editing lists of values

The universe designer decides whether a list of values is editable in BusinessObjects. You can edit a LOV only if the designer has granted you the right to do so.

Because the list of values is a query, you edit it by editing its corresponding query in the Query Panel. You can edit the query by:

• applying conditions to restrict the values returned
• applying sorts to order the values
• building a combined query
• including additional objects in the query

The following example shows you how to edit a list of values.

**EXAMPLE**

**Showing cities and regions in a list of cities**

The Island Resorts Marketing universe has a City object. However, it is possible for cities in different regions to have the same name. To identify a city precisely, you want to display the city’s region along with the city’s name in a list of values. To do this:
1. Click **Universes** on the **Tools** menu. The Universes dialog box appears.

![Universes dialog box](image)

2. Select the Island Resorts Marketing universe and click **List of Values**.
3. Navigate to the City object beneath the Customer class.

![Lists of Values dialog box](image)
4. Click **Edit**.  
The Query Panel appears showing the query for the City object’s list of values.

5. Add the Region object to the query.

6. Click **Run**.  
Now when you see the City list of values dialog box it shows the cities and their regions.
If you select Hierarchical View, the List of Values dialog box shows the cities organized hierarchically within their regions.
Assigning personal data to a list of values

If you always choose from the same subset of values when applying conditions, you can limit your choices by assigning personal data to an object's list of values. This decreases the time required for the query because it is quicker to retrieve values from a list than it is to query the database.

You can assign personal data to a list of values from three sources:

• text files
• Microsoft Excel files
• dBase files

The following sections describe how to assign a list of values from a personal data file by associating a file of cities with the City object in the Island Resorts Marketing universe.

Assigning personal data from a text file

To do this:
1. Create a text file containing the values you want to include in the list (for example cities):
   Cities
   Los Angeles
   San Diego
   San Francisco
   (The first entry in the text file is the name of column of data, which you can display in the list of values dialog box.)
2. In BusinessObjects, click Universes on the Tools menu, select the Island Resorts Marketing universe and click Lists of Values.
3. Open the Customer class and select the City object.

4. Select Personal Data.
   The Access Personal Data dialog box opens.
5. Click **Browse** to locate the text file containing the values.

6. Select **First Row Contains Column Names**. (In this case, the first row is called ‘Cities’.)

7. Click **Run**.

8. To view the list of values, click **Display**.

![List of Values of City](image)

9. Click **OK**.

**Assigning personal data from an Excel file**

To do this:

1. Create an Excel file containing the list of values.

2. In BusinessObjects, click **Universes** on the **Tools** menu, select the Island Resorts Marketing universe and click **List of Values**.

3. Open the Resort class and highlight the City object.

4. Select **Personal Data**.
   The Access Personal Data dialog box appears.

5. Select Microsoft Excel Files from the Format dropdown list.
   The bottom half of the dialog box now contains controls that you use to select
the cells containing the values in the Excel file.

6. Click **Browse** and select the Excel sheet containing the values.
7. Select the worksheet containing the values in the Sheet Name dropdown list.
8. Select the range of cells (for example A1:A4) containing the list of values in the Range Definition box  
   or  
   Select the named range containing the list of values in the Named Range dropdown list.
9. Select First Row Contains Column Names if the range contains the column name.
10. Click **Run**.

**Assigning personal data from a dBase file**

To do this:

1. Click **Universes** on the **Tools** menu, select the Island Resorts Marketing
1. Click universe and click List of Values.
2. Open the Resort class and select the City object.
3. Select *Personal Data*.
   The Access Personal Data dialog box appears.
4. Select *dBase Files* from the Format dropdown list.
5. Click *Browse* and select the dBase (.dbf) file that contains the list of values.
6. Click *Run*. 
Displaying, refreshing and purging lists of values

BusinessObjects lets you display, refresh and purge lists of values at any time. To do so:

1. Click **Universes** on the **Tools** menu.
   The Universes dialog box appears.
2. Select the universe that contains the list of values you want, then click **Lists of Values**.
   The Lists of Values dialog box appears.
3. Open a class by clicking its + sign, then select the object whose list of values you want to view, refresh or purge.
4. Click the button you want:

<table>
<thead>
<tr>
<th>Button</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display</strong></td>
<td>BusinessObjects displays the values in the List of Values of <strong>Object Name</strong> dialog box. To view the list in table or hierarchy format, click <strong>Tabular View</strong> or <strong>Hierarchical View</strong> respectively. These options are useful if the list of values contains more than one object; that is, it combines two or more columns of values:</td>
</tr>
<tr>
<td></td>
<td>• In <strong>Tabular View</strong>, the columns appear next to each other.</td>
</tr>
<tr>
<td></td>
<td>• In <strong>Hierarchical View</strong>, the values from the first column appear as a folder. The folder contains the values of the second column. If there is a third column, the second column appears as a folder that contains these, and so on.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>BusinessObjects runs the query for the list of values, and a refreshed list appears.</td>
</tr>
<tr>
<td><strong>Purge</strong></td>
<td>BusinessObjects empties the .lov file corresponding to the list of values.</td>
</tr>
</tbody>
</table>

5. Click **OK** to close the dialog box.
TIP

You can populate a purged list of values by clicking **Refresh**.
Creating Calculations
Overview

This chapter explains how to create and display simple calculations in tables and crosstabs. The final section of this chapter describes all you need to know about converting currencies to and from euros in BusinessObjects.

The following chapters give further information on using calculations in BusinessObjects:

- **Formulas, Local Variables and Functions on page 465** explains how you can use the power of the BusinessObjects formula editor to write your own calculations.
- **Calculation Contexts and Extended Syntax on page 407** explains the more sophisticated aspects of writing formulas and gives some background about how the BusinessObjects calculation engine works.
- **Calculation Troubleshooting on page 437** explains the error messages and other problems you may encounter when inserting formulas and calculations in your reports and explains how to fix these problems.
- For information on using calculations in charts, see the *BusinessObjects User’s Guide: Report Techniques and Formatting*. 
Calculations

BusinessObjects has standard calculation functions that enable you to make quick calculations on the data in reports. These calculations are available directly from a menu. The most commonly used calculations are also available on the Report toolbar. This section describes how to make calculations using the menu and toolbar functions.

You can also add calculations to your reports by writing your own formulas. This is described in Formulas, Local Variables and Functions on page 465.

Adding simple calculations to reports

To add a simple calculation to a report:
1. Select the row or column on which you want to make the calculation.
2. Click Data, then the calculation you want, on the Calculations menu.

<table>
<thead>
<tr>
<th>To...</th>
<th>Choose...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate the sum of the selected data,</td>
<td>Sum</td>
</tr>
<tr>
<td>Count all rows for a Measure object.</td>
<td>Count</td>
</tr>
<tr>
<td>Count distinct values for a Dimension or Detail object.</td>
<td>Count</td>
</tr>
<tr>
<td>Count the number of rows in the body including all duplicate and empty rows.</td>
<td>Count All</td>
</tr>
<tr>
<td>Calculate the average of the selected data.</td>
<td>Average</td>
</tr>
<tr>
<td>Display the maximum value of the selected data,</td>
<td>Maximum</td>
</tr>
<tr>
<td>Display the minimum value of the selected data,</td>
<td>Minimum</td>
</tr>
</tbody>
</table>
Accessing Data and Data Analysis

Creating Calculations

The result of the calculation is inserted in a new cell in the table or crosstab.

**NOTE**

Variance syntax containing the Where operator will work in all cases except when you have two cascading Wheres (a variance using a Where operator that contains a formula that also uses a Where operator) or in certain contexts, for example the variance of the variable "<Revenue>" in Report.

### Count and Count All

In the example below, if you insert a Count on the Resort column, the result of the calculation is 3 because there are three different resorts, Bahamas Beach, Hawaiian Club and French Riviera. The Count function counts values of a dimension object that are the same only one time. This is called a distinct count.

If you insert Count All on the same column, the result is 12 because there are twelve rows of data in the resort column. The Count All function counts all rows including empty and duplicate rows.

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Bahamas Beach</td>
<td>$234,891.00</td>
</tr>
<tr>
<td>Q1</td>
<td>French Riviera</td>
<td>$206,665.00</td>
</tr>
<tr>
<td>Q1</td>
<td>Hawaiian Club</td>
<td>$159,470.00</td>
</tr>
<tr>
<td>Q2</td>
<td>Bahamas Beach</td>
<td>$237,672.00</td>
</tr>
<tr>
<td>Q2</td>
<td>French Riviera</td>
<td>$312,905.00</td>
</tr>
<tr>
<td>Q2</td>
<td>Hawaiian Club</td>
<td>$341,790.00</td>
</tr>
<tr>
<td>Q3</td>
<td>Bahamas Beach</td>
<td>$363,422.00</td>
</tr>
<tr>
<td>Q3</td>
<td>French Riviera</td>
<td>$236,128.00</td>
</tr>
<tr>
<td>Q3</td>
<td>Hawaiian Club</td>
<td>$356,150.00</td>
</tr>
<tr>
<td>Q4</td>
<td>Bahamas Beach</td>
<td>$345,369.00</td>
</tr>
<tr>
<td>Q4</td>
<td>French Riviera</td>
<td>$158,665.00</td>
</tr>
<tr>
<td>Q4</td>
<td>Hawaiian Club</td>
<td>$356,250.00</td>
</tr>
</tbody>
</table>

**Count:** 3

**Count All:** 12

---

Creating Calculations
If you look at the Revenue column, Count and Count All return 12. For a measure object, the Count function counts all rows.

**Using the Calculation toolbar**

You can also use the buttons on the Report toolbar for certain calculations.

![Calculation buttons](image)

- a. Sum
- b. Percentage
- c. Count
- d. Variance
- e. Variance percentage

**Deleting calculations from a table or a crosstab**

To delete calculations from a table or crosstab:

1. Select the row or column where the calculation is displayed.
2. Click **Calculations** then the calculation on the **Data** menu.

You can see which calculations have been applied to a row or column of data. On the Data menu, the icon next to the calculation is dimmed or has a check mark next to it to show it has been used. Buttons on the toolbar are dimmed to show they have been used on the selected data.

**NOTE**

The calculation commands available on the menu and toolbar depend on the data you have selected in the table or crosstab.

**Making calculations on dimension and detail objects**

You can use the following calculations on dimension and detail objects:

- **Count**
- **Minimum**
- **Maximum**
- **Sum**
- **Percentage**
- **Variance**
- **Variance percentage**
Calculation examples

The following section contains several examples of making calculations using the Calculations menu.

**EXAMPLE**

**Displaying total revenue and subtotals**

You want to calculate and display the total revenue in a table. To do this:

1. Click in the Revenue column.
2. Click **Sum** on the Calculations toolbar.
   - BusinessObjects displays the total revenue in the footer at the end of the table.
   - The sum button on the Calculations toolbar is dimmed.

You now decide you want to insert a break on this table to display subtotals for each resort.

3. Click inside the Resort column and click **Insert Break**.
   - The data is broken up and an empty row is inserted at the end of each resort section.
4. Click in the Revenue column.
5. Click **Sum** once to remove the existing calculation.
6. Click **Sum** again to insert the subtotals.
   - BusinessObjects displays a subtotal for each resort and a total at the end of the table.
BusinessObjects inserts a subtotal for each resort, and a grand total for all resorts.
Displaying average, maximum and minimum revenue

The following table has four different calculations for revenue. Notice that the Average revenue, the Maximum and Minimum revenues and the Sum are all displayed in separate rows in the order they were applied. Each figure is identified by the name of the calculation. When you add a Percentage calculation, BusinessObjects adds an extra column that shows each row as a percentage of the total.

<table>
<thead>
<tr>
<th>Resort</th>
<th>Quarter</th>
<th>Service Line</th>
<th>Revenue</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>Q1</td>
<td>Accommodation</td>
<td>$197,236</td>
<td>21.16 %</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>Q1</td>
<td>Food &amp; Drinks</td>
<td>$25,585</td>
<td>3.01 %</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>Q1</td>
<td>Recreation</td>
<td>$30,870</td>
<td>3.47 %</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Q1</td>
<td>Accommodation</td>
<td>$136,470</td>
<td>15.64 %</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Q1</td>
<td>Food &amp; Drinks</td>
<td>$23,492</td>
<td>3.47 %</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Q1</td>
<td>Recreation</td>
<td>$14,545</td>
<td>2.07 %</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Q1</td>
<td>Accommodation</td>
<td>$237,910</td>
<td>26.12 %</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Q1</td>
<td>Food &amp; Drinks</td>
<td>$64,160</td>
<td>7.12 %</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Q1</td>
<td>Recreation</td>
<td>$55,220</td>
<td>6.06 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average</td>
<td>$97,046</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum</td>
<td>$237,910</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum</td>
<td>$23,492</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sum</td>
<td>$730,416</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Percent</td>
<td>100.00 %</td>
<td></td>
</tr>
</tbody>
</table>

Calculating the difference in revenue between two quarters

In this example you have a crosstab that displays revenue per resort per quarter and total revenue per quarter. You want to show in the table the difference in revenue between two quarters. To do this:

1. Select Q2 with the mouse, then, holding down the Ctrl key, select Q1.
2. Click **Variance Percentage** on the Calculations toolbar. BusinessObjects displays the difference in revenue between Q1 and Q2 for each resort as a percentage in a new cell called Q2-Q1.
3. Select Q4 with the mouse.
4. Hold down the Control key and select Q3. BusinessObjects displays the difference in revenue between Q1 and Q2 for...
Accessing Data and Data Analysis

Calculations

You can also use the Shift key to select columns and rows. If you use the Shift key, BusinessObjects does not take into account the order in which you selected the columns and rows and always selects cells from top to bottom and from left to right. If you carry out the calculation above using the Shift key, BusinessObjects will calculate Q1-Q2.

---

**NOTE**

You can also use the Shift key to select columns and rows. If you use the Shift key, BusinessObjects does not take into account the order in which you selected the columns and rows and always selects cells from top to bottom and from left to right. If you carry out the calculation above using the Shift key, BusinessObjects will calculate Q1-Q2.

---

**Re-using a calculation elsewhere in a report**

You can drag a calculation from a table or a crosstab and place it in a different position in a report. BusinessObjects keeps the formula with the cell. To do this:

1. Click inside the cell containing the calculation.
2. Click inside the cell again and, holding down your mouse button, drag the cell to the desired position.
3. Release the mouse button.

---

**NOTE**

When you move a cell containing a calculation to a different part of the report, the calculation result depends on where exactly the cell is positioned as this can change the context in which the calculation is made. For more information on this point see Calculation Contexts and Extended Syntax on page 407.

---

**EXAMPLE**

**Displaying total revenue as a table title**

You have a report showing overall total and maximum revenues, and total and maximum revenue by resort and service line. You want the total revenue to appear at the top of the report. To do this:

---

<table>
<thead>
<tr>
<th></th>
<th>Bahamas Beach</th>
<th>French Riviera</th>
<th>Hawaiian Club</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>$240,881.00</td>
<td>$2,089,805.00</td>
<td>$357,170.00</td>
</tr>
<tr>
<td>Q2</td>
<td>$297,872.00</td>
<td>$2,412,165.00</td>
<td>$341,780.00</td>
</tr>
<tr>
<td>Q4-Q1</td>
<td>-5.78 %</td>
<td>16.11 %</td>
<td>-4.31 %</td>
</tr>
<tr>
<td>Q3</td>
<td>$368,422.00</td>
<td>$2,251,125.00</td>
<td>$395,180.00</td>
</tr>
<tr>
<td>Q4</td>
<td>$2,450,689.00</td>
<td>$1,582,305.00</td>
<td>$385,500.00</td>
</tr>
<tr>
<td>Q4-Q1</td>
<td>-6.89 %</td>
<td>-29.88 %</td>
<td>-2.43 %</td>
</tr>
</tbody>
</table>
1. Select the Sum cell at the bottom of the report and drag it to the top of the report.

```
<table>
<thead>
<tr>
<th>Report</th>
<th>Service Line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>Accommodation</td>
<td>$973,964.00</td>
</tr>
<tr>
<td></td>
<td>Food &amp; Drinks</td>
<td>$169,960.00</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>$170,100.00</td>
</tr>
<tr>
<td>Bahamas Beach</td>
<td>Sum</td>
<td>$1,214,444.00</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>$673,664.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Accommodation</td>
<td>$563,050.00</td>
</tr>
<tr>
<td></td>
<td>Food &amp; Drinks</td>
<td>$107,400.00</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>$184,770.00</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Sum</td>
<td>$839,220.00</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>$563,250.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Accommodation</td>
<td>$987,210.00</td>
</tr>
<tr>
<td></td>
<td>Food &amp; Drinks</td>
<td>$277,750.00</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>$220,700.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Sum</td>
<td>$1,479,660.00</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>$901,210.00</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
<td>$3,206,324.00</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>$901,210.00</td>
</tr>
</tbody>
</table>
```

2. Click **Cell** on the **Insert** menu and type “Total Revenue” in the inserted cell, then align it next to the cell you placed at the top of the report.
3. Select the block at the bottom of the report that contains the overall sum and maximum calculations and press the Delete key to delete it.

<table>
<thead>
<tr>
<th>Resort</th>
<th>Service Line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>Accommodation</td>
<td>$673,664.00</td>
</tr>
<tr>
<td></td>
<td>Food &amp; Drinks</td>
<td>$169,880.00</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>$101,100.00</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>$1,944,644.00</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td>$673,664.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resort</th>
<th>Service Line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>French Riviera</td>
<td>Accommodation</td>
<td>$963,250.00</td>
</tr>
<tr>
<td></td>
<td>Food &amp; Drinks</td>
<td>$107,400.00</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>$864,770.00</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>$1,735,420.00</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td>$963,250.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resort</th>
<th>Service Line</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hawaiian Club</td>
<td>Accommodation</td>
<td>$861,210.00</td>
</tr>
<tr>
<td></td>
<td>Food &amp; Drinks</td>
<td>$777,760.00</td>
</tr>
<tr>
<td></td>
<td>Recreation</td>
<td>$230,760.00</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>$1,579,730.00</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td>$861,210.00</td>
</tr>
</tbody>
</table>
Converting to and from Euros

The following section describes how BusinessObjects uses built-in functions to help you quickly and accurately convert a currency to or from euros.

What is the euro?

The euro is the official currency unit of the European countries that belong to the EMU (European Monetary Union).

Each one of these countries has a conversion rate between the euro and its former national currency. EMU regulations stipulate how the conversion must be carried out.

BusinessObjects euro conversion functions adhere to these stipulations.

Displaying the euro symbol

The euro is designated by an official symbol. To display this currency symbol in BusinessObjects, you need to have a Windows operating system that can display it or you need to install a euro font upgrade to your Windows operating system.

If your operating system cannot display the euro symbol, you can use the official ISO three-letter code, EUR, instead.

How does the conversion work?

Each currency has a fixed conversion rate of six significant digits. The number of digits after the decimal point depends on the number of digits before the decimal point.

EXAMPLE

Converting to euros: six-digit conversion rates

The conversion rate for Dutch guilders (NLG) is 2.20371, with 1 digit before the decimal and five after.

The conversion rate for Belgian francs (BEF) is 40.3399, with 2 digits before the decimal point and four after.

To convert to euros from an EMU-compliant currency, you divide the local currency amount using the six-digit conversion rate for that currency and then round the result to display the appropriate number of decimal digits.

To convert from euros to an EMU-compliant currency, you multiply the amount in euros by the six-digit conversion rate for the target currency and then round the result to display the appropriate number of decimal digits.
Conversion errors
When you convert an EMU-compliant currency to or from euros, you use the fixed six-digit conversion rate and then round the result to the appropriate number of decimal digits. Rounded numbers are less accurate than the original numbers and the round error on a given amount may be fairly negligible or quite substantial. BusinessObjects allows you to display round errors after you have converted amounts to and from euros to assess the importance of the difference.

Displaying currency formats in BusinessObjects
BusinessObjects uses the default currency defined for your operating system defined in the Windows Regional Settings in the Windows Control Panel.

You can display custom currencies in your BusinessObjects reports in addition to the default currency provided by your operating system. For example, if your default currency is $, all the amounts in your report will be in US dollars. If you also want to display a column with revenue in euros, you can add your own currency format in BusinessObjects. See the BusinessObjects User’s Guide: Report Techniques and Formatting for details.

Converting to euros
You can convert an amount from one of the eleven EMU currencies into euros using one easy mouse click.

To do this:
1. Select the column, row or cell that contains the data you want to convert.
2. Click Euro, then click Convert To Euro on the Data menu.
   - If BusinessObjects recognizes the currency, it carries out the conversion and displays the result.
   - If BusinessObjects does not recognize the currency, it displays the Select Currency dialog box.
3. Choose the currency from the list box and click OK.
   The data is converted. If the source data was formatted as currency, the converted data is formatted with the euro currency formatting defined in the Conversion Rates table.

How does BusinessObjects recognize the currency of the selected data?
BusinessObjects recognizes the currency of selected data if:
• it is formatted as currency using the standard Windows currency symbol
• and the selected currency is in the Conversion Rates list
For example, 47 DM will be recognized by BusinessObjects as German marks.
If the data is not formatted as a currency BusinessObjects recognizes, the Select Currency dialog box is displayed.

If the currency you want to convert to euros is not in the list, you can add it as described below.

**Converting from euros**

You can convert a figure from euros into one of the eleven EMU currencies using one easy mouse click.

To do this:

1. Select the column, row or cell that contains the data you want to convert.
2. Click **Euro**, then **Convert From Euro** on the **Data** menu.

   The Select Currency dialog box opens.

3. Choose the currency you want to convert the euros to and click **OK**.

   The data is converted. If the data in euros was formatted as currency, the converted data is formatted with the currency formatting defined in the Conversion Rates table.

**Displaying rounding errors**

A rounding error occurs because when an amount is converted to or from euros, the result is rounded to show only the appropriate number of decimal digits. The rounding error returns the difference between the rounded number and the number before rounding.

To display rounding errors:

1. Select the column, row or cell that contains the converted data.
2. Click **Display Rounding Errors** on the **Data** menu.

   The rounding errors are displayed in a new column, row or cell which is inserted after the selected column, row or cell.
NOTE

The Display Rounding Errors command is only available if the selected cell contains a number that has been converted to or from euros.

Conversion rates

BusinessObjects stores the conversion rates and other information for the EMU currencies and uses this information to carry out conversions to and from euros. You can edit this information and add other currencies to this list.

Viewing conversion rates

1. Click Euro, then click Display Conversion Rates on the Data menu. The Conversion Rates dialog box appears.

This dialog box shows the following information about each currency:
Editing information about a currency
1. Click Euro, then click Display Conversion Rates on the Data menu.
2. The Conversion Rates dialog box appears.
3. Select the currency you want to edit from the list.
4. Click Edit.
The Edit Conversion Rate dialog box appears.

This dialog box displays the information that BusinessObjects currently uses for the selected currency.
5. Make any necessary changes by typing in the new information over the old information in the text boxes.
6. Click OK when you have finished.
The Edit Conversion Rate dialog box closes.
The new information concerning the selected currency is now displayed in the list in the Conversion Rates dialog box.
7. Click OK to close the Conversion Rates dialog box.
Adding a new currency

1. Click Euro, then click Display Conversion Rates on the Data menu. The Conversion Rates dialog box appears.

2. Click Add. The Add Conversion Rate dialog box appears.

3. Enter the required information in each text box. You must enter information in each text box.

4. Click OK to close the Add Conversion Rate dialog box. The new currency is now displayed in the list in the Conversion Rates dialog box.

5. Click OK to close the Conversion Rates dialog box.

EXAMPLE

Adding US dollars to the currency list

You want to add US dollars to the list of currencies since you regularly use dollar to euro conversion in your business. You have today's dollar/euro exchange rate. Since the price of the dollar against the euro fluctuates daily, you will have to update dollar currency information regularly.
To add US dollars to the currency list:
1. Click Euro, then click Display Conversion Rates on the Data menu. The Conversion Rates dialog box appears.
2. Click Add. The Add Conversion Rate dialog box appears.
3. Enter the information in the Add Conversion Rate dialog box:
   - Enter USD in the Currency box.
   - Enter 0.9 in the Rate box.
   - Enter 2 in the Decimal box.
   - Enter $ in the Format box.
   - Enter US Dollars in the Label box.
4. Click OK.
US dollars now appears in your list and will be recognized by BusinessObjects so that you can convert between US dollars and euros using the Convert to Euro and Convert from Euro menu commands.

Fixed and fluctuating currency rates
The conversion rates between the euro and EMU-compliant currencies are fixed. When converting euros to any other currency you must use the daily rate set by financial institutions.

Triangulation
If you want to convert from one EMU-compliant currency to another EMU-compliant currency, you cannot simply use cross rates. You have to use a procedure called triangulation. To triangulate, you convert the first currency into euros using a six-digit conversion rate and then convert the euros into the second currency using another six-digit conversion rate.

EXAMPLE
Triangulation: converting between EMU-compliant currencies
In the following example, you want to convert an amount in German marks into Belgian francs. To do this, you first have to convert the German marks into euros using the six-digit conversion rate for German marks. You then round the euros to no less than three decimal digits. Finally you convert the euros into Belgian francs using the six-digit conversion rate for Belgian francs. You can carry out this conversion by writing a BusinessObjects formula.

The following BusinessObjects formula converts 100 DEM to 2063 BEF.
EuroConvertFrom(EuroConvertTo(100, "DEM", 3), "BEF", 0)
Creating Calculations
Calculation Contexts and Extended Syntax
Overview

This chapter is designed to help you use the powerful calculation capabilities delivered with BusinessObjects.

Who should read this chapter

This chapter especially concerns those users whose business requires that they perform advanced calculations. The information here is also useful for any user who has experienced computation errors in their BusinessObjects reports.

What's in this chapter

The chapter provides information on how BusinessObjects performs calculations in reports. It describes the concepts behind the BusinessObjects calculation engine. Most specifically, its aim is to explain the extended syntax that enables you to manipulate complex aggregations in reports.

You can find calculation troubleshooting information on error messages such as #COMPUTATION in "Calculation Troubleshooting" on page 437.
Introduction to contexts and extended syntax

This section introduces you to calculation contexts and extended calculation syntax. Each calculation that you place in a report has a default context determined by where you place it. With extended syntax you can override these defaults and control the context yourself.

Semantically-dynamic calculations

In BusinessObjects, you create a report by building a query that retrieves data from a database. Typically, your query contains:

- **Dimensions**, which retrieve character-type data (customer names, product names), or dates (years, quarters, reservation dates).
- **Measures**, which retrieve numeric data that is the result of calculations. For example, in the BusinessObjects demo universe, Revenue is the calculation of number of items sold multiplied by item price.

When you run the query, BusinessObjects retrieves the data corresponding to the dimensions and measures you selected. BusinessObjects calculates measures dynamically, based on the dimensions with which they appear. Here's an example.
EXAMPLE

Revenue per region per year, and revenue per region

The report illustrated below contains two tables: revenue per region per year, and revenue per region.

BusinessObjects dynamically calculates the Revenue measure according to the dimensions in the table. If you remove the Year column, BusinessObjects returns revenue per region:

<table>
<thead>
<tr>
<th>Region</th>
<th>Year</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coast</td>
<td>FY 1998</td>
<td>$19,124.00</td>
</tr>
<tr>
<td>Mid West</td>
<td>FY 1998</td>
<td>$239,962.00</td>
</tr>
<tr>
<td>Mid West</td>
<td>FY 1999</td>
<td>$191,956.00</td>
</tr>
<tr>
<td>Mid West</td>
<td>FY 2000</td>
<td>$162,556.00</td>
</tr>
<tr>
<td>South</td>
<td>FY 1998</td>
<td>$129,330.00</td>
</tr>
<tr>
<td>South</td>
<td>FY 1999</td>
<td>$135,550.00</td>
</tr>
<tr>
<td>South</td>
<td>FY 2000</td>
<td>$136,851.00</td>
</tr>
<tr>
<td>West</td>
<td>FY 1998</td>
<td>$996,200.00</td>
</tr>
<tr>
<td>West</td>
<td>FY 1999</td>
<td>$1,000,310.00</td>
</tr>
<tr>
<td>West</td>
<td>FY 2000</td>
<td>$1,200,950.00</td>
</tr>
</tbody>
</table>

Revenue by region by year. Revenue by region.

Understanding input and output contexts

BusinessObjects defines an input context and an output context to determine the result of an aggregate calculation. Remember that a context is made up of one or more dimensions. The following table defines which dimensions in a report make up which context:

<table>
<thead>
<tr>
<th>The...</th>
<th>Consists of one or more dimensions that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input context</td>
<td>Go into the calculation</td>
</tr>
<tr>
<td>Output context</td>
<td>Determine the result of the calculation</td>
</tr>
</tbody>
</table>

(BusinessObjects also supports reset contexts, which are used in cumulative aggregations such as running totals. To find out about reset contexts, refer to Reset contexts on page 420).
This section explains how BusinessObjects defines input and output contexts in different parts of a report. This information is important if you

• want to understand the different results BusinessObjects returns from the same formula in different parts of a report
• cannot obtain the results you need from the default calculation behavior
• need to fix errors such as #COMPUTATION

How BusinessObjects defines input and output contexts

To understand how BusinessObjects defines input and output contexts, you must first understand the terms body and local context. The following table provides a definition of these terms, and shows how they map to input and output contexts in a report:

<table>
<thead>
<tr>
<th>The...</th>
<th>Consists of one or more dimensions that...</th>
<th>And by default is the same as the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>Are present in the part of the report (for example a block) where the calculation is inserted</td>
<td>Input context.</td>
</tr>
<tr>
<td>Local context</td>
<td>Govern the part of the report where the calculation is inserted (For example, a master variable in a section).</td>
<td>Output context.</td>
</tr>
</tbody>
</table>

Below is an illustrated example that helps you to understand the dimensions in the body and the local context—and thus the default input and output contexts—in different parts of a report.

EXAMPLE

Calculate revenue in various default contexts

The following report displays revenue per city per quarter in 2000. The user has placed calculations in different parts of the report:

• Total revenue at the top of the report
• Revenue in the Year section
• Revenue in the table, and
• Average revenue in the break footer.
The table below the illustration explains the default input and output contexts BusinessObjects uses to obtain the result of each calculation.

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Contexts and Extended Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Revenue:</td>
<td>$3,286,524</td>
</tr>
<tr>
<td>2000 Revenue Total:</td>
<td>$1,063,554</td>
</tr>
<tr>
<td>Q1 Revenue Total:</td>
<td>$266,960</td>
</tr>
<tr>
<td>City</td>
<td>Revenue</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Chicago</td>
<td>$23,070</td>
</tr>
<tr>
<td>Dallas</td>
<td>$30,145</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>$23,620</td>
</tr>
<tr>
<td>Average:</td>
<td>$27,580</td>
</tr>
<tr>
<td>Q2 Revenue Total:</td>
<td>$272,490</td>
</tr>
<tr>
<td>City</td>
<td>Revenue</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Chicago</td>
<td>$25,160</td>
</tr>
<tr>
<td>Dallas</td>
<td>$35,890</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>$26,540</td>
</tr>
<tr>
<td>Average:</td>
<td>$28,397</td>
</tr>
<tr>
<td>Q3 Revenue Total:</td>
<td>$286,993</td>
</tr>
</tbody>
</table>

- a. Total revenue for the report.
- b. Total revenue by year.
- c. Total revenue by year and quarter.
- d. Average revenue by year, quarter.
Using your understanding of input and output contexts

Once you understand input and output contexts, you are ready to use extended syntax.

The extended syntax of an aggregate formula contains not only the basic formula, but also the dimensions that make up the aggregation’s calculation contexts. Here’s an example.

**EXAMPLE**

**The extended syntax of an aggregate formula**

The report shown here calculates running total resort revenue per country:

<table>
<thead>
<tr>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>$97,144.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>$1,479,860.00</td>
</tr>
<tr>
<td><strong>Sum:</strong></td>
<td>$2,525,144.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>French Riviera</td>
<td>$3,200,524.00</td>
</tr>
<tr>
<td><strong>Sum:</strong></td>
<td></td>
</tr>
</tbody>
</table>
You obtain the running totals by applying the RunningSum function on the Revenue column. The basic formula for the calculation is

\[ \text{RunningSum}(<\text{Revenue}>) \]

The extended syntax of the formula, in which the explicit input and output contexts are shown, is

\[ \text{RunningSum}(<\text{Revenue}> \text{ In Body}) \text{ In } <\text{Country}> \]

where Body is the input context, and Country, the output context.

You can use your knowledge of extended syntax to:

- understand the results BusinessObjects returns by default (see below), and
- change the default input and output contexts of a formula to get the results you need. For information on how to do this, refer to Using extended syntax for advanced calculations on page 416.

Viewing the extended syntax of a formula

BusinessObjects provides two simple ways of viewing a formula’s extended syntax, to find out the dimensions in the input and/or output context of an aggregate calculation. You can view extended syntax by

- using the Formula Bar, or
- using the Define As Variable command on the Data menu

Using the Formula Bar

1. If the Formula Bar is not displayed, click Formula Bar on the View menu.
2. Click the cell containing the calculation.
3. Hold the mouse pointer over the Formula Bar.
4. The extended syntax of the formula appears in a tooltip.
Using the Define As Variable command
1. Click the cell containing the calculation.
2. Click Define as Variable on the Data menu. The Define As Variable dialog box appears.
3. Click Evaluate the formula in its context. The extended syntax of the formula appears in the dialog box:

![Define the variable dialog box](image)

4. Click Cancel to close the dialog box.
Using extended syntax for advanced calculations

When you insert a simple aggregation in a report, for example to calculate average revenue, BusinessObjects writes a simple formula, in this case
\[ \text{Average}(<\text{Revenue}>) \]

By default, BusinessObjects evaluates the formula in its default context, as explained in the previous section, Understanding input and output contexts on page 410.

What if you do not want the default context? What if you need average revenue per city in a table containing region and city? By default, BusinessObjects returns results based on region and city, that is, both dimensions from the local context.

What if you need a result based on a dimension that is available in the document but which you do not want to display in the report?

BusinessObjects provides extended syntax so that you can specify the dimensions to use in your calculations. Here’s an example.

**EXAMPLE**

Calculating the number of cities per region

In a table containing the dimensions City and Region, if you insert a count, BusinessObjects counts the cities one by one, using the following formula:
\[ \text{Count}(<\text{City}>) \]

By default, BusinessObjects makes the calculation based on the dimensions in the table (Region, City). There’s only one city per city, so BusinessObjects returns 1 every time.

Extended syntax enables you to specify that you want BusinessObjects to count the cities *per region*. To obtain this result, you must extend the formula:
\[ \text{Count}(<\text{City}>) \text{ In } <\text{Region}> \]
Now you get the result you were looking for:

<table>
<thead>
<tr>
<th>Region</th>
<th>City</th>
<th>Number of Cities</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coast</td>
<td>Boston</td>
<td>3</td>
</tr>
<tr>
<td>East Coast</td>
<td>New York City</td>
<td>3</td>
</tr>
<tr>
<td>East Coast</td>
<td>Washington D.C.</td>
<td>3</td>
</tr>
<tr>
<td>Mid West</td>
<td>Chicago</td>
<td>2</td>
</tr>
<tr>
<td>Mid West</td>
<td>Memphis</td>
<td>2</td>
</tr>
<tr>
<td>South</td>
<td>Dallas</td>
<td>2</td>
</tr>
<tr>
<td>South</td>
<td>Houston</td>
<td>2</td>
</tr>
<tr>
<td>West</td>
<td>Los Angeles</td>
<td>3</td>
</tr>
<tr>
<td>West</td>
<td>San Diego</td>
<td>3</td>
</tr>
<tr>
<td>West</td>
<td>San Francisco</td>
<td>3</td>
</tr>
</tbody>
</table>

Defining calculation contexts with extended syntax

Using extended syntax, you specify the dimensions you need for your calculation. In the example above, we specified the dimension Region in order to return the number of cities per region. When you use extended syntax to specify dimensions in this way, you define calculation contexts other than the default contexts assigned by BusinessObjects.

BusinessObjects lets you define your own input and output contexts. A quick reminder of the difference between them:

<table>
<thead>
<tr>
<th>The...</th>
<th>Consists of one or more dimensions that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input context</td>
<td>Go into the calculation.</td>
</tr>
<tr>
<td>Output context</td>
<td>Determine the result of the calculation.</td>
</tr>
</tbody>
</table>

Thus, in the example on page 416 (number of cities per region), we defined an output context, Region. Here’s an example of a calculation with an input context defined by the user.

**EXAMPLE**

Calculating the minimum revenue per city for each region

You display Region, City and Revenue in a table. When you remove City from the table, BusinessObjects automatically calculates revenue per region, because Region is now the only dimension in the body.

The only way to display minimum revenue per city for each region is to define an input context, because the local context does not contain the City dimension. BusinessObjects allows you to do this, because even though you have removed City from the report display, it is still available in the document.

Using extended syntax for advanced calculations...
The formula and the result are as follows:

\[\text{Min}(<\text{Revenue}> \text{ In } (<\text{Region},<\text{City}>))\]

<table>
<thead>
<tr>
<th>Region</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coast</td>
<td>$19,124.00</td>
</tr>
<tr>
<td>Mid West</td>
<td>$441,294.00</td>
</tr>
<tr>
<td>South</td>
<td>$430,299.00</td>
</tr>
<tr>
<td>West</td>
<td>$324,640.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Region</th>
<th>Revenue</th>
<th>Min Rev per City</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coast</td>
<td>$19,124.00</td>
<td>$6,425.00</td>
</tr>
<tr>
<td>Mid West</td>
<td>$441,294.00</td>
<td>$441,294.00</td>
</tr>
<tr>
<td>South</td>
<td>$430,299.00</td>
<td>$430,299.00</td>
</tr>
<tr>
<td>West</td>
<td>$324,640.00</td>
<td>$3,385.00</td>
</tr>
</tbody>
</table>

**How to define input and output contexts**

When you insert an aggregation such as Sum, BusinessObjects writes a simple formula, for example:

\[\text{Sum}(<\text{Revenue}>)\]

To define contexts, you have to edit formulas, so the first thing to do is to display the Formula Bar. To do this, click Formula Bar on the View menu. Now, when you click a cell in a report, its formula appears in the Formula Bar:

**Syntax for input and output contexts**

To define contexts, you add arguments to a formula. The syntax for input and output contexts is as follows:

\[\text{AggregateFunction}(<\text{measure}> \text{ In } <\text{input context}>) \text{ In } <\text{output context}>\]

The following example explains this in more detail.
EXAMPLE

A formula containing input and output contexts

The following formula returns the minimum revenue per city per region:

\[ \text{Min(<Revenue> In (<Region>,<City>)) In <Region>} \]

The input context consists of Region and City, while the output context is Region.

To add an input and output context to a formula

This procedure shows you step-by-step how to add an input and/or output context to a simple calculation you have already inserted. The procedure is based on the example on page 417, “Calculating the minimum revenue per city for each region”.

1. If the Formula Bar is not displayed, click **Formula Bar** on the **View** menu.
2. Click inside the cell containing the calculation you want to change, in this case a column containing the Revenue measure.
   
   The formula appears in the Formula Bar, for example

   \[ \text{Min(<Revenue>)} \]

3. If you’re adding input context, click to the left of the closing parenthesis. For an output context, click to the right of the closing parenthesis.
4. Type a space, type In, then type another space.
5. Type the name of the dimension you want to specify as the context.
6. Type < > on either side of the dimension, for example <City>
7. If you need to add dimensions to the context, separate each one with a comma, and place the list of dimension in parentheses, like this:
   
   \[ (<Region>,<City>,<Year>) \]
8. Press Enter to validate the formula.

TIP

You can also use the Formula Editor to add contexts to a formula. The advantage is that you can double-click variables to add them to the formula, instead of having to type them.
To display the Formula Editor, click **Formula Editor** on the left of the Formula Bar.
Reset contexts

You use a reset context in a *cumulative* aggregation, such as running total revenue per quarter. The reset context consists of one or more dimensions which reset the value of the calculation to zero each time a dimension value changes. This is best explained with an example.

**EXAMPLE**

Calculating running total revenue per country

You want to calculate running total revenue per country per year, and naturally you want the calculation to be reset for each country. When the value of Country changes, you want the calculation to begin at 0.

To obtain this result, you display Country, Year and Revenue in a table, and apply a break on Country. You then add the cumulative aggregation

```
=RunningSum(<Revenue>;<Country>)
```

in which you specify Country as the reset context. Here’s what you get:

<table>
<thead>
<tr>
<th>Country</th>
<th>Year</th>
<th>Revenue</th>
<th>Running Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>FY1996</td>
<td>$220,310.00</td>
<td>$220,310.00</td>
</tr>
<tr>
<td></td>
<td>FY1997</td>
<td>$269,172.00</td>
<td>$589,482.00</td>
</tr>
<tr>
<td></td>
<td>FY2000</td>
<td>$325,840.00</td>
<td>$915,322.00</td>
</tr>
<tr>
<td>France</td>
<td></td>
<td></td>
<td>$915,322.00</td>
</tr>
<tr>
<td>US</td>
<td>FY1996</td>
<td>$767,814.00</td>
<td>$767,814.00</td>
</tr>
<tr>
<td></td>
<td>FY1997</td>
<td>$825,390.00</td>
<td>$1,593,204.00</td>
</tr>
<tr>
<td></td>
<td>FY2000</td>
<td>$856,990.00</td>
<td>$2,451,194.00</td>
</tr>
<tr>
<td>US</td>
<td></td>
<td></td>
<td>$2,451,194.00</td>
</tr>
</tbody>
</table>

**How to define reset contexts**

You define a reset context by specifying one or more dimensions in the cumulative formula. The syntax is

```
=RunningAggregateFunction(<measure>;<dimension>)
```

giving, for example

```
=Sum(<Revenue>;<Year>;<Region>)
```
To define a reset context:
1. If the Formula Bar is not displayed, click \textbf{Formula Bar} on the View menu.
2. Click inside the cell containing the calculation you want to change.
   The formula appears in the Formula Bar.
3. Type a parenthesis before the function name.
4. Type a semi-colon (;) after the measure.
5. Type the name of the dimension you want to use as the reset context, for example, \texttt{<Year>}
6. Add other dimensions if necessary; separate each with a comma and place the list in parentheses.
   A reset context with more than one dimension looks like this: \texttt{;<Year>,<Region>}
7. Type a closing parenthesis at the end of the formula, then press Enter.

\textbf{Using reset contexts in crosstabs}

A crosstab displays data in rows and columns, as opposed to a table which displays data in columns only. Measures are typically placed in the body of a crosstab at the intersection of rows and columns. The crosstab illustrated here shows revenue per region in 1998, 1999 and 2000:

\begin{tabular}{|l|c|c|c|}
\hline
Region & FY1998 & FY1999 & FY2000 \\
\hline
Mid West & $120,362.00 & $150,666.00 & $162,566.00 \\
South & $120,330.00 & $136,560.00 & $136,969.00 \\
West & $86,360.00 & $130,210.00 & $150,350.00 \\
\hline
\end{tabular}

Now, to calculate running total revenue, you insert a break on Year, then change Revenue to \texttt{=RunningSum(<Revenue>)}

Here's the result:

\begin{tabular}{|l|c|c|}
\hline
Region & FY1998 & FY1999 \\
\hline
Mid West & $120,362.00 & $270,782.00 \\
South & $120,330.00 & $407,350.00 \\
West & $86,360.00 & $437,310.00 \\
\hline
\end{tabular}
\begin{tabular}{|l|c|c|}
\hline
Region & FY1999 & FY2000 \\
\hline
Mid West & $270,782.00 & $279,072.00 \\
South & $407,350.00 & $547,930.00 \\
West & $437,310.00 & $747,570.00 \\
\hline
\end{tabular}
Notice that BusinessObjects calculates the running totals for both Year and Region. In other words, the figures are added from left to right, and from top to bottom.

To reset the running sum for either Year or Region, you add a reset context to the formula. The formula to reset the running sum to 0 for each year is

\[ \text{=RunningSum(<Revenue>;<Year>)} \]

and the result is as shown:

<table>
<thead>
<tr>
<th></th>
<th>FY1996</th>
<th>FY1997</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid West</td>
<td>$138,000</td>
<td>$128,367</td>
</tr>
<tr>
<td>South</td>
<td>$121,500</td>
<td>$256,892</td>
</tr>
<tr>
<td>West</td>
<td>$38,300</td>
<td>$353,072</td>
</tr>
<tr>
<td></td>
<td>$196,800</td>
<td>$580,660</td>
</tr>
</tbody>
</table>

**NOTE**

Users of previous versions of BusinessObjects may have used the keywords Col and Row to define reset contexts in crosstabs. These keywords are no longer necessary. You can use them, but they have no added value. For example

\[ \text{=RunningSum(<Revenue>; Col <Year>)} \]

and

\[ \text{=RunningSum(<Revenue>; <Year>)} \]

return the same result.

**Syntax for combining reset, input and output contexts**

When input and output contexts are specified, the syntax for reset contexts is as follows:

\[ \text{=RunningAggregateFunction(<measure> In <input context>;<reset context>) In <output context>} \]

**Modifying contexts with the operators ForEach and ForAll**

When you define a context, you specify the dimension(s) it must contain. The syntax requires that you use the In operator to introduce the context:

\[ \text{=Min(<Revenue> In (<Region>,<City>) In <Region>)} \]
In is the default operator for defining a context. However, you can modify the context you’re defining by replacing In by ForEach or ForAll. The following table provides a definition of ForEach and ForAll:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ForEach</td>
<td>Adds dimensions to the input or output context.</td>
</tr>
<tr>
<td>ForAll</td>
<td>Removes dimensions from the input or output context.</td>
</tr>
</tbody>
</table>

The following example illustrates how ForEach and ForAll work.

**EXAMPLE**

**Using ForEach and ForAll**

A table containing Region, City and Revenue returns the revenue per city per region.

The Year dimension is also available in the document. You want to display maximum revenue per city per year, but you don’t want to add Year to the table. Instead, you *add* Year to the local context by using ForEach:

\[ \text{Max}(<\text{Revenue}> \text{ForEach} <\text{Year}>) \]

Finally, you want to display maximum revenue per region, but want City to remain in the table. In other words, you have to *remove* City from the local context without physically removing the City column. You achieve this by using ForAll. Here’s the formula and the result:
Accessing Data and Data Analysis

Calculation Contexts and Extended Syntax

=Max(<Revenue>) ForAll <City>

<table>
<thead>
<tr>
<th>Region</th>
<th>City</th>
<th>Revenue</th>
<th>Max Rev per Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coast</td>
<td>New York</td>
<td>$4,230</td>
<td>$10,704</td>
</tr>
<tr>
<td>East Coast</td>
<td>Washington</td>
<td>$10,704</td>
<td>$10,704</td>
</tr>
<tr>
<td>Mid West</td>
<td>Chicago</td>
<td>$411,594</td>
<td>$411,594</td>
</tr>
<tr>
<td>South</td>
<td>Dallas</td>
<td>$400,099</td>
<td>$400,099</td>
</tr>
<tr>
<td>West</td>
<td>Los Angeles</td>
<td>$301,545</td>
<td>$301,545</td>
</tr>
<tr>
<td>West</td>
<td>San Diego</td>
<td>$16,715</td>
<td>$301,545</td>
</tr>
<tr>
<td>West</td>
<td>San Francisco</td>
<td>$1,380</td>
<td>$301,545</td>
</tr>
</tbody>
</table>

Getting the same result: ForAll <City> vs. In <Region>

In the previous example, we calculated maximum revenue per region by using ForAll to remove City from the local context. You could obtain the same result by making Region the output context.

Here’s how this works. The local context is City and Region. Unless you specify otherwise, BusinessObjects returns maximum revenue per city per region. You want maximum revenue per region. To calculate this, you must “tell” the formula to “ignore” City, either by
- removing City from the local context, or
- defining an output context, Region

So, these two formulas,

=Max(<Revenue>) ForAll <City>
=Max(<Revenue>) In <Region>

where the local context is Region and City, obtain the same result.

Now try ForEach <City> vs. In (<Region>,<City>)

In the example entitled Calculating the minimum revenue per city for each region on page 417, you used

=Min(<Revenue> In (<Region>,<City>))

to calculate minimum revenue per city in a table containing only Region and Revenue. The default input context is Region, but you added City to it.

You could obtain the same result by adding City with the ForEach operator, the formula being

=Min(<Revenue> ForEach <City>)

Calculation Contexts and Extended Syntax
Using the Rank function and extended syntax

The Rank function allows you to rank the values of a dimension based on a measure. The syntax is:

\[ \text{Rank} \text{(dimension ,measure)} \]

The first table below shows revenue per quarter per country—that is, quarters based on their total revenues, ignoring countries. (You can see this more clearly if you look at the table below the first table, which breaks on quarters and shows total quarter revenues.) The rank formula in this table is:

\[ \text{Rank} \text{(<Quarter> ,<Revenue>)} \]
But what if you put a break or a section on country? The result is that the quarters are ranked for each country separately, as shown in the table to the right.

<table>
<thead>
<tr>
<th>Country</th>
<th>Quarter</th>
<th>Revenue</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Q1</td>
<td>$208,855.00</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$242,165.00</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$238,125.00</td>
<td>1</td>
</tr>
<tr>
<td>US</td>
<td>Q1</td>
<td>$859,652.00</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$682,051.00</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$503,829.00</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$310,829.00</td>
<td>4</td>
</tr>
</tbody>
</table>

This is because, by default, BusinessObjects includes country in the calculation context. Since there are two values for Country, BusinessObjects makes a calculation for each value.

You can modify the calculation context by using the ForAll context operator. This tells BusinessObjects to ignore the break (or section) on Country. The syntax is:

```
=Rank(<Quarter>,<Revenue>) ForAll <Country>
```
The result is:

![Table 1](image1)

BusinessObjects now calculates the rank according to revenue generated per quarter and for all values of the Country dimension.

What if you now want to rank all eight quarters in the block according to revenue generated?

To do this, you have to remove the Country break from the rank calculation context, but keep it in the revenue calculation context. The best way to achieve this is to create a new variable to calculate the revenue and then include this in the formula inserted in the Rank column.

Here’s how to do it:

1. Create a new measure variable using the following syntax:

   `<Revenue> ForEach <Country>`

   This variable calculates the revenue for each country.

2. Name the variable `Revenue ForEach Country`.

3. Insert the following formula in the rank column:

   `=Rank(<Quarter>, <Revenue ForEach Country>) ForAll <Country>`
The result is:

<table>
<thead>
<tr>
<th>Country</th>
<th>Quarter</th>
<th>Revenue</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Q1</td>
<td>$332,966</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$342,466</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$328,156</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$188,566</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Country</th>
<th>Quarter</th>
<th>Revenue</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>Q1</td>
<td>$352,261</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$679,452</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$659,572</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$930,829</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Using the Rank function in crosstabs**

In the following example, you have a crosstab that displays revenue per quarter per resort. You have inserted a column to display rank after each Resort column and inserted the following syntax:

=RANK(<Quarter>,<Revenue>)

The result is:

<table>
<thead>
<tr>
<th>Resort</th>
<th>Quarter</th>
<th>Revenue</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahama Beach</td>
<td>Q1</td>
<td>$224,881</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$277,872</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$303,423</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$246,261</td>
<td>4</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Q1</td>
<td>$308,566</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$342,195</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$236,125</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$159,566</td>
<td>4</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Q1</td>
<td>$387,170</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$341,780</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$395,150</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$386,510</td>
<td>2</td>
</tr>
</tbody>
</table>

The problem is that the rank is the same for all quarters in all resorts.

What you want to do is rank the quarters for each resort in the crosstab. To do this, you have to specify that you want to calculate rank for each resort in the output context. The syntax is:

=RANK(<Quarter>,<Revenue>) ForEach <Resort>

And the result is:

<table>
<thead>
<tr>
<th>Resort</th>
<th>Quarter</th>
<th>Revenue</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahama Beach</td>
<td>Q1</td>
<td>$224,881</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$277,872</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$303,423</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$246,261</td>
<td>2</td>
</tr>
<tr>
<td>French Riviera</td>
<td>Q1</td>
<td>$308,566</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$342,195</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$236,125</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$159,566</td>
<td>4</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td>Q1</td>
<td>$387,170</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$341,780</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$395,150</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$386,510</td>
<td>2</td>
</tr>
</tbody>
</table>
Defining contexts with keywords

Calculation contexts consist of one or more dimensions. In the examples so far, we've defined contexts by writing the names of dimensions inside the formula. BusinessObjects provides keywords that enable you to define contexts without listing the specific dimensions you need. A keyword corresponds to the dimension or dimensions in a specific part of the report. Keywords can define all types of context in extended syntax—input, output or reset.

Keywords offer you the following advantages:

• It is often quicker to add one keyword to a formula than to write a list of dimensions.
• The syntax for keywords is simple: you add the keyword to the formula, avoiding things like missing parentheses and misspelled dimensions.
• If you add or remove dimensions from the report, perform drag-and-drop or slice-and-dice, you do not have to rewrite formulas containing keywords.

This section provides:

• definitions of the keywords you can use
• an example of how to use a keyword in a formula
• a step-by-step procedure for writing formulas with keywords
• information on how BusinessObjects assigns keywords to formulas, and
• an explanation of how report filters behave when you use keywords.
Keywords: definitions and example

The four keywords you can use in formulas are Report, Block, Body and CurrentPage. Each keyword corresponds to the dimension or dimensions in a specific part of the report. The definitions in the following table indicate
• in which part of a report you can use each keyword, and
• the dimensions the keywords correspond to.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Can be used in formulas...</th>
<th>And corresponds to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>Inside a block</td>
<td>The dimensions in the block.</td>
</tr>
<tr>
<td></td>
<td>Outside a block</td>
<td>The dimensions in the current section.</td>
</tr>
<tr>
<td>Block</td>
<td>Inside a block</td>
<td>The dimensions in the current section.</td>
</tr>
<tr>
<td>Report</td>
<td>Anywhere in the report</td>
<td>All the dimensions in the document.</td>
</tr>
<tr>
<td>CurrentPage</td>
<td>Inside the current page</td>
<td>The dimensions in the current page.</td>
</tr>
</tbody>
</table>

To illustrate how keywords can be used, here’s an example.

EXAMPLE

Calculating a grand total by using the Report keyword

You want to display the grand total revenue across all dimensions in a report. This calculation does not require extended syntax if the formula is placed in a cell at the very top of the report, because in this case the default output context contains all the dimensions in the report. But you want to display the information elsewhere, for example inside a table footer.

To obtain this result, you could define an output context by listing all the dimensions in the report. A much simpler solution is to use the Report keyword in the following formula:

=Sum(<Revenue>) In Report

Wherever you display this formula in the report, you obtain the grand total.
NOTE
A formula in which the Report keyword is used to define the output context always returns a single value.

How to define a context using a keyword
Here’s the step-by-step procedure for using a keyword to define a context:
1. If the Formula Bar is not displayed, click Formula Bar command on the View menu.
2. Click the cell in which you want to place the formula, then write the basic formula, for example, =Sum(<Revenue>) in the Formula Bar.
   1. If you’re adding input context, click to the left of the closing parenthesis. For an output context, click to the right of the closing parenthesis.
   2. Type a space, type In, then type another space.
   3. Type the keyword (Block, Body, Report, or CurrentPage), then press Enter.

How BusinessObjects uses keywords
You use keywords in extended syntax when you need to define contexts in aggregate formulas. BusinessObjects defines contexts in all aggregate formulas, however simple, and often uses keywords to do so. The following example illustrates this behavior.

EXAMPLE
The formula BusinessObjects writes for a simple calculation
You create a master/detail report that shows revenue per resort per year. You insert a sum on Revenue. This is a simple calculation requiring no extended syntax.
BusinessObjects, however, writes a full formula in which it specifies input and output contexts. The formula is
=Sum(<Revenue> In Body) In (<Year>)
where Body corresponds to Resort (the dimension in the table containing the calculation, the local context), and Year is the output context.
You can view the full formula by clicking the cell containing the sum, then resting the mouse pointer over the Formula Bar. The formula appears in a tooltip.
You can also view the full formula by clicking Define As Variable on the Data menu.

**Benefits**

In simple aggregations such as the one in the example above, you rarely need to understand the extended syntax BusinessObjects uses. But what if you don’t understand the numbers BusinessObjects displays in a report? What if you’re having a hard time fixing an error such as #COMPUTATION?

In these cases, you can use the extended syntax BusinessObjects provides to:

- Understand the computation behind the report display, then
- Edit your formula to get the result you want. The formula displayed in the tooltip is a base for you to work from.

**Filters and keywords**

BusinessObjects still applies filters when you use keywords in aggregations, but you can force BusinessObjects to ignore them.

**How to force BusinessObjects to ignore filters**

You have a master/detail report and a filter on the master variable. You want to calculate a grand total using the Report keyword. You don’t want to compute the filter but you don’t want to delete it either.

The solution is to use the NoFilter function. The syntax is

=NoFilter(formula)

an example being

=NoFilter(Sum(<Revenue>) In Report)
Quick reference

This section provides a quick reference to the terms and concepts discussed in this chapter.

The information below is organized by category:
- Basic terms such as document, microcube, aggregation, etc.
- Calculation contexts (local, body, input, output and reset)
- Context operators (In, ForEach, ForAll)
- Keywords (Report, Block, Body, CurrentPage).

Frequently used terms
A selection of useful terms often used in this context.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation</td>
<td>A calculation that returns totals, percentages, etc. in which any of the following functions are used: Average, Count, Max, Min, StdDev, StdDevP, Sum, Var, VarP.</td>
</tr>
<tr>
<td>Cumulative aggregation</td>
<td>A calculation that returns running totals, percentages, etc. in which any of the following functions are used: RunningAverage, RunningCount, RunningMax, RunningMin, RunningSum.</td>
</tr>
<tr>
<td>Dimension</td>
<td>Qualification of an object, variable or formula that returns text (names, IDs, etc.) or dates. In a report, dimensions make up calculation contexts.</td>
</tr>
<tr>
<td>Extended syntax</td>
<td>The syntax of a formula for an aggregation, in which the input and output contexts for the aggregation are displayed. You can use extended syntax to define your own input and output contexts. You have to use extended syntax to define reset contexts.</td>
</tr>
<tr>
<td>Formula</td>
<td>The definition of the content of a cell. Can contain functions, operators, variables and text.</td>
</tr>
</tbody>
</table>
**Accessing Data and Data Analysis**

**Calculation Contexts and Extended Syntax**

**Calculation contexts**

A calculation context consists of one or more dimensions that determine the input and output of aggregate calculations. Important: In BusinessObjects 5.1.x, the

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
<td>Qualification of an object, variable or formula that returns numeric data, such as revenue. The result of a measure is by default determined by the dimensions in the context in which the measure is placed in a report.</td>
</tr>
<tr>
<td>Microcube</td>
<td>Set of data returned by a query. Because documents can contain data from different sources in BusinessObjects, any given document can contain multiple microcubes.</td>
</tr>
<tr>
<td>Report</td>
<td>Pages in a document where data is displayed. A document can contain many reports; each report has a tab at the bottom of the application window. A report can display only a subset of data from the document. You can use any dimension - displayed or not - as input or output for a calculation.</td>
</tr>
<tr>
<td>Variable</td>
<td>A named formula. BusinessObjects always stores query results as variables.</td>
</tr>
</tbody>
</table>

**Context** | **Definition**
--- | ---
Local | Dimension or dimensions that govern the part of the report where the calculation is inserted (e.g., a master variable in a section). |
Body | Dimension or dimensions that are present in the part of the report (e.g., a block) where the calculation is inserted. |
NOTE

In BusinessObjects 5.1.x and later, the SQL for contexts is generated differently from previous versions. Therefore, if you run a query that was built in a previous version to 5.0.x, you receive #SYNTAX, #COMPUTATION and #ERROR messages in the upgrading report.

Context operators

In the formula for an aggregation using extended syntax, an operator introduces the input and/or output context.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>The default operator. Includes the specified dimension(s) in the context.</td>
</tr>
<tr>
<td>ForEach</td>
<td>Modifies a context by including the specified dimension(s).</td>
</tr>
<tr>
<td>ForAll</td>
<td>Modifies a context by excluding the specified dimension(s).</td>
</tr>
</tbody>
</table>
Keywords

In extended syntax, a keyword stands for the dimension(s) in a specific part of a report. The keyword enables you to define contexts in aggregate formulas by using one word rather than listing the dimensions in the context.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Block</strong></td>
<td>Corresponds to the dimensions in the current section.</td>
</tr>
<tr>
<td></td>
<td>Can only be used in a formula placed inside a block.</td>
</tr>
<tr>
<td><strong>Body</strong></td>
<td><strong>When the formula is placed...</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Corresponds to...</strong></td>
</tr>
<tr>
<td>Outside a block</td>
<td>The dimensions in the current section.</td>
</tr>
<tr>
<td>Inside a block</td>
<td>The dimensions in the block.</td>
</tr>
<tr>
<td><strong>Report</strong></td>
<td>Anywhere in the report</td>
</tr>
<tr>
<td></td>
<td>All the dimensions in the document.</td>
</tr>
<tr>
<td><strong>CurrentPage</strong></td>
<td>Inside the current page</td>
</tr>
<tr>
<td></td>
<td>The dimensions in the current page.</td>
</tr>
</tbody>
</table>
Calculation Troubleshooting
Overview

This chapter describes solutions for computation errors that might occur in BusinessObjects. A computation error always starts with # and appears in the cells where the data should be.

The bulk of the information in this chapter focuses on the most common errors, #COMPUTATION and #MULTIVALE. For the full list of errors covered here, please refer to the table of contents or the index.

If you need help with #COMPUTATION and #MULTIVALUE errors in your reports, you are strongly advised to first read Calculation Contexts and Extended Syntax on page 407, because these errors often result from the use of extended syntax.
This section provides descriptions, examples and solutions for the following errors in BusinessObjects:

- #COMPUTATION in cumulative aggregations such as running average revenue per city
- #COMPUTATION in non-aggregate formulas, for example using IF THEN ELSE statements.

#COMPUTATION in cumulative aggregations

The expression “cumulative aggregations” refers to any aggregation containing a running aggregate function such as RunningMax or RunningAverage. In reports, cumulative aggregations let you answer questions such as “What’s the running percentage of revenue per city for each year?”.

**NOTE**

The information presented here requires that you understand BusinessObjects extended syntax, which is described in Using extended syntax for advanced calculations on page 416.

**Description of #COMPUTATION in a cumulative aggregation**

#COMPUTATION can occur in cumulative formulas where a reset context is defined. A reset context consists of a dimension, and resets a running calculation to zero when the value of the dimension changes. For further information on reset contexts, refer to Reset contexts on page 420.

#COMPUTATION occurs in cumulative aggregations for the following reason: the reset context is not included in the output context.

Here’s an example.

**EXAMPLE**

#COMPUTATION resulting from a running sum with a reset context

You’ve displayed running totals per country per year, and reset the calculation per country. The formula is as follows:

=RunningSum(<Revenue>;<Country>)
You now set a break on Year, because you want to display the running total for each year. You copy the formula from the running total column and paste it in the break footer. \#COMPUTATION appears:

<table>
<thead>
<tr>
<th>Year</th>
<th>Country</th>
<th>Running Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>FY1998</td>
<td>France</td>
<td>$295,340.00</td>
</tr>
<tr>
<td></td>
<td>US</td>
<td>$172,514.00</td>
</tr>
</tbody>
</table>

Why? Because the reset context in the formula you pasted is Country, but you placed the formula in the footer of the break on Year. Thus, the reset context (Country) is not in the output context (Year).

To fix this error, you need to display the Formula Bar and change the reset context from Country to Year. The correct formula is:

$$=\text{RunningSum(<Revenue>;<Year>)}$$

Alternatively, you could set the break on Country, but you would obtain a different result.

Solution for \#COMPUTATION in a cumulative aggregation

To avoid \#COMPUTATION in a cumulative aggregation, the reset context must be included in the output context. In other words, the dimension or dimensions after the semi-colon (;) in the formula must also be listed after the operator (\text{In}, \text{ForEach} or \text{ForAll}) on the right of the formula.

REMINDER

If you do not define an output context in the formula, BusinessObjects makes the calculation using the local context. For further information, refer to Understanding input and output contexts on page 410.

To fix your formula:

1. Display the Formula Bar (Formula Bar command, View menu), then click the
cell containing the formula.

2. Check that the dimension in the reset context is also specified in the output context. The following table will help you determine this:

<table>
<thead>
<tr>
<th>When the reset context is Region and...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only the output context is defined, for example, =RunningSum(&lt;Revenue&gt;;&lt;Region&gt;) In (&lt;Region&gt;,&lt;City&gt;))</td>
<td>Region must also be specified in the output context.</td>
</tr>
<tr>
<td>Only the input context is defined, for example, =RunningSum(&lt;Revenue&gt; In (&lt;Region&gt;,&lt;City&gt;);&lt;Region&gt;)</td>
<td>Region must be present in the local context.</td>
</tr>
<tr>
<td>Neither the input nor output contexts are defined, e.g., =RunningSum(&lt;Revenue&gt;;&lt;Region&gt;)</td>
<td>Region must be present in the local context.</td>
</tr>
<tr>
<td>Both the input and the output contexts are defined, e.g., =RunningSum(&lt;Revenue&gt; In (&lt;Region&gt;,&lt;City&gt;);&lt;Region&gt;) In (&lt;City&gt;;&lt;Region&gt;)</td>
<td>Region must also be specified in the output context.</td>
</tr>
</tbody>
</table>

3. Edit the formula in the Formula Bar, then press Enter.

#COMPUTATION in non-aggregate formulas

#COMPUTATION can occur in formulas that do not contain aggregate or running aggregate functions.

- **Description of #COMPUTATION in a non-aggregate formula**

  In non-aggregate formulas, #COMPUTATION occurs because the output context is not included in the input context. Here’s an example.

**EXAMPLE**

#COMPUTATION caused by a conditional formula in a break footer

You can use an IF THEN formula to set a condition for displaying data in a report. The following formula

= If (<Year>="FY1998") Then <Revenue>

displays the Revenue measure only when the year is 1998.
In the report illustrated here, the formula is inserted in a break footer, and this causes #COMPUTATION:

<table>
<thead>
<tr>
<th>Region</th>
<th>City</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coast</td>
<td>New York City</td>
<td>#MULTIVALE</td>
</tr>
<tr>
<td>East Coast</td>
<td>Washington D.C.</td>
<td>#MULTIVALE</td>
</tr>
<tr>
<td>Mid West</td>
<td>Chicago</td>
<td>#MULTIVALE</td>
</tr>
<tr>
<td>South</td>
<td>Dallas</td>
<td>#MULTIVALE</td>
</tr>
<tr>
<td>West</td>
<td>Los Angeles</td>
<td>#MULTIVALE</td>
</tr>
<tr>
<td>West</td>
<td>San Diego</td>
<td>#MULTIVALE</td>
</tr>
<tr>
<td>West</td>
<td>San Francisco</td>
<td>#MULTIVALE</td>
</tr>
</tbody>
</table>

Why? The input context required to display revenue for 1998 is Year, whereas the output context of the break footer is Resort. To fix this error, you add Year to the output context using the ForEach operator:

```plaintext
=Max((If (<Year>="FY1998") Then <Revenue>) ForEach <Year>)
```

Finally, you add Max (or Min) to return the single value you need:

```plaintext
=Max((If (<Year>="FY95") Then <Revenue>) ForEach <Year>)
```

BusinessObjects now displays Revenue for 1998 in the break footer.

**Solution for #COMPUTATION in a non-aggregate formula**

You must:

1. Edit the formula so that the output context is included in the input context, as described in the example above.
2. Add Max or Min to the beginning of the formula, followed by an opening parenthesis.
3. Add a closing parenthesis at the end of the formula, then press Enter.
#MULTIVALEU

This section provides descriptions, examples and solutions for #MULTIVALEU in BusinessObjects:
- #MULTIVALEU in aggregations such as maximum revenue per year
- #MULTIVALEU in break headers and footers

#MULTIVALEU in aggregations

The expression "aggregations" refers to any calculation containing an aggregate function such as Sum, Count, Min, Max etc. In reports, aggregations let you answer questions such as “What is the minimum revenue per city for each region?”. This section provides a description of why #MULTIVALEU occurs in aggregations, with an example, and offers a solution.

NOTE

The information presented here requires that you understand BusinessObjects extended syntax, which is described in Using extended syntax for advanced calculations on page 416.

Description of #MULTIVALEU in an aggregation

#MULTIVALEU occurs in aggregations because

The output context is not included in the local context.

EXAMPLE

#MULTIVALEU in an aggregation

Look at the following table:

<table>
<thead>
<tr>
<th>Region</th>
<th>City</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>East Coat</td>
<td>New York City</td>
<td>#MULTIVALEU</td>
</tr>
<tr>
<td>East Coat</td>
<td>Washington D.C.</td>
<td>#MULTIVALEU</td>
</tr>
<tr>
<td>Mid West</td>
<td>Chicago</td>
<td>#MULTIVALEU</td>
</tr>
<tr>
<td>South</td>
<td>Dallas</td>
<td>#MULTIVALEU</td>
</tr>
<tr>
<td>West</td>
<td>Las Angeles</td>
<td>#MULTIVALEU</td>
</tr>
<tr>
<td>West</td>
<td>San Diego</td>
<td>#MULTIVALEU</td>
</tr>
<tr>
<td>West</td>
<td>San Francisco</td>
<td>#MULTIVALEU</td>
</tr>
</tbody>
</table>

The desired result in the right-hand column is revenue for a dimension that is not present in the table, for example, Year. The formula
Accessing Data and Data Analysis

==<Revenue> In <Year>

returns #MULTIVALUE because Year is specified as the output context but it is not present in the local context (the table).

**NOTE**

The formula in this example does not begin with an aggregate function (Min, Max, etc.). If you omit the function in this way, BusinessObjects calculates a sum by default. However, if you had specify Sum or another aggregate function in the formula above, for example

=Sum(<Revenue>) In <Year>

BusinessObjects returns #COMPUTATION.

If the Year dimension is available in the document, you can, using extended syntax, calculate revenue per year without displaying Year in the table. To do this, you must define a formula with an input and an output context, like this:

=Sum(<Revenue> In (<Region>,<City>,<Year>) In <Year>)

**Solution for #MULTIVALUE in an aggregation**

To avoid #MULTIVALUE in an aggregation, the output context must be included in the local context, which means that the dimension(s) specified on the right of the formula after In or ForEach must be present. To fix your formula:

1. Display the Formula Bar (Formula Bar command, View menu), then click the cell containing the formula.
2. Look for the following:
   - Does the output context contain dimensions that are not in the block or section in which the formula is inserted? For example, if the following formula appears in a block that does not contain Year, you’ll get a #MULTIVALUE:
     =Min(<Revenue>) In (<Year>)
   - Does the output context contain more dimensions than the local context?
     - For example
     =Min(<Revenue>) In(<Region>,<City>)
   - returns #MULTIVALUE in a table containing only one of these dimensions.
3. Edit the formula in the Formula Bar, then press Enter.

**#MULTIVALUE in break headers and footers**

#MULTIVALUE can occur when you insert a variable in a break header or footer in a table or crosstab.
Description of #MULTIVALUE in a break header or footer

A break splits up the values of a variable and thus enables you to make calculations.

A break footer is a cell at the bottom of each value of the break. Users typically display text or calculations such as running totals in break footers.

#MULTIVALUE occurs in a break header or footer:

- **If You...**
- **Then You...**
- **And...**

<table>
<thead>
<tr>
<th>if you...</th>
<th>then you...</th>
<th>and...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set a break on a variable</td>
<td>insert a second variable in the break footer</td>
<td>These two variables have a 1:1 relationship, as is the case with Customer and Age.</td>
</tr>
</tbody>
</table>

Here’s an example.

**EXAMPLE**

#MULTIVALUE in a break footer

The table below shows the running total revenue for two customers. When you insert Age in the break footer, BusinessObjects returns #MULTIVALUE:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>#MULTIVALUE</td>
<td>#MULTIVALUE</td>
<td>$6,036.00</td>
</tr>
</tbody>
</table>

This error occurs because variables with a 1:1 relationship, (Customer and Age), are inserted at the same break level. By default, a break is based on one variable only.

**Solution for #MULTIVALUE in a break header or footer**

You fix this problem by including the variable from the header or footer in the break definition. Here’s how to do it:

1. Click inside the table or crosstab containing the break, then select click...

   #MULTIVALUE
Breaks on the Format menu.
The Breaks dialog box appears.
2. Click the icon of the break concerned, then click Edit.
3. A dialog box listing all the variables in the report appears.
4. Click the check box next to the variable you want to display in the break footer (Age in the example on page 445), then click OK.
5. Click OK in the Breaks dialog box.
BusinessObjects displays the variable in the break footer.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aria</td>
<td>Bahamas Beach</td>
<td>$8,036.00</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>$6,036.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer</th>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
<td>Bahamas Beach</td>
<td>$141,684.00</td>
</tr>
<tr>
<td></td>
<td>French Riviera</td>
<td>$141,300.00</td>
</tr>
<tr>
<td>Hawaiian Club</td>
<td></td>
<td>$158,710.00</td>
</tr>
<tr>
<td>64</td>
<td></td>
<td>$441,594.00</td>
</tr>
</tbody>
</table>
#MULTIVALE in a cell at the section level

Description
A report section displays data in a master cell and in a block or blocks. If you use two variables at the section level, #MULTIVALE can occur. Here's an example.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
<td>$9,588.00</td>
</tr>
<tr>
<td></td>
<td>$27,640.00</td>
</tr>
<tr>
<td></td>
<td>$27,180.00</td>
</tr>
<tr>
<td></td>
<td>$35,412.00</td>
</tr>
<tr>
<td></td>
<td>$34,714.00</td>
</tr>
<tr>
<td></td>
<td>$36,954.00</td>
</tr>
<tr>
<td></td>
<td>$36,614.00</td>
</tr>
<tr>
<td></td>
<td>$36,654.00</td>
</tr>
<tr>
<td></td>
<td>$36,744.00</td>
</tr>
<tr>
<td></td>
<td>$40,085.00</td>
</tr>
<tr>
<td></td>
<td>$41,588.00</td>
</tr>
<tr>
<td></td>
<td>$38,916.00</td>
</tr>
<tr>
<td></td>
<td>$42,026.00</td>
</tr>
</tbody>
</table>

#MULTIVALE in a section containing name and address
You want to display customer names and addresses in a section, and the customers’ revenue details in a table. You build a table containing Customer, Address, Invoice Date and Revenue, then drag Customer out of the block to create a section.

The next step is to drag Address out of the block, and drop it next to Customer. Here’s what you get:

Why? Because by default, there is only one master variable per section.
Solution #1 for #MULTIVALUE at the section level

The way round this problem is to turn the variable that returns the error into a measure. Here’s how to do it:

1. Click the cell containing the #MULTIVALUE error.
2. Click Variables on the Data menu.
   
   The Variables dialog box appears:

3. In the dialog box, click the variable that returns the error.
4. The next step depends on the type of variable you just clicked:

<table>
<thead>
<tr>
<th>If the Edit button is grayed</th>
<th>If the Edit button is available</th>
</tr>
</thead>
<tbody>
<tr>
<td>This means that the variable you need to turn into a measure comes from a query on a universe. You cannot edit variables that come from queries on universes, so you have to create a new variable instead.</td>
<td>You can change the variable into a measure.</td>
</tr>
</tbody>
</table>

- Click **Add**. The Variable Editor appears.
- In the **Definition** tab, type a name for the new variable.
- In the Qualification box, click **Measure**.
- Click the **Formula** tab.
- In the Variables box, double-click the name of the variable that returned the error. For example if you are creating a new variable to replace Address, double-click Address in the Variables box.
- Click **OK**.

5. In the Variables dialog box, click **Replace**. You replace the erroneous variable with the one you have just either created or modified. The data appears instead of the error.

**Solution #2 for #MULTIVALUE at the section level**

You can also fix #MULTIVALUE at the section level by applying the Min or Max function, which forces BusinessObjects to display only one value. This solution works for master variables with a 1:1 relationship, such as Customer and Address (unless your customers have more than one address). If the variable returning #MULTIVALUE contains more than two values, you will only be able to display the first and last of these by applying Min or Max.

1. Click the master cell displaying #MULTIVALUE.
2. In the Formula Bar, type Min or Max after =, then add parentheses, like this: \(-\text{Min}(\text{<Address>})\)
3. Press Enter.
The cell is not wide enough to display the data it contains in full.

Double-click the cell’s right border. BusinessObjects widens the cell to autofit the data.
#ALERTER

**Description**

This error occurs when an alerter contains a missing variable. For example, if the definition of the alerter contains the Revenue variable, #ALERTER appears if Revenue does not exist in the report.

The definition of the alerter itself may also be the cause of this error. For example, if you try to compare a measure with a dimension (Revenue greater than Country), #ALERTER is returned.

**Solution**

You can:

- Obtain the missing data by adding the corresponding object to the query (Edit Data Provider command, Data menu)
- Edit the alerter so that its definition contains only available data (Alerters command, Format menu)
- Deactivate the alerter (Alerters command, Format menu, then uncheck the alerter in the dialog box).
- Check that the definition of the alerter does not contain a comparison such as that of a measure with a dimension.

**TIP**

To use the Alerters command on the Format menu, first click any cell containing data.
#DICT.ERROR

**Description**
In BusinessObjects, you can format a report by applying a template. You do this by choosing Report, then clicking **Apply Template** on the **Format** menu. The **Apply Template** dialog box appears.

A template contains a set of variables, also referred to as the variable dictionary. The report you wish to format also contains a variable dictionary. When you apply a template, unless you click **Options** button in the **Apply Template** dialog box in order to specify the correspondence between the two variable dictionaries, BusinessObjects automatically replaces the variables in the template with the variables in the report. In some cases, BusinessObjects cannot match the variable dictionaries and returns #DICT.ERROR.

**Solution**
To fix this problem:
1. Click the cell containing #DICT.ERROR, then click **Variables** on the **Data** menu.
   The Variables dialog box appears.
2. Click the variable or formula that you want to display in the selected cell.
3. Click **Replace**.
   The Variables dialog box closes, and the variable or formula appears in the report.

**How to avoid #DICT.ERROR**
If the template you select has many more variables than the report you are working on, you may well end up with #DICT.ERROR. To avoid the problem:
1. Click **Report**, then click **Apply Template** on the **Format** menu.
   The Apply Template dialog box appears.
2. Click the template you want to apply, then click Options.
   The Template Options dialog box appears.
3. Uncheck **Replace Variables Automatically**, then click **Define**.
   The Replace Variables dialog box appears.
4. In the Report Variables box, click a variable, then click a corresponding variable in the Template Variables box, making sure that the two variables are of the same type (dimension, measure or detail).
5. Click **Replace**, then repeat the previous step until you have replaced all
variables from the template with variables from the report.

6. Click **OK** in the Replace Templates dialog box.
   You return to the Template Options dialog box.

7. Ensure that *Delete Undefined Template Variables* is checked, then click OK.

8. Click **OK** in the Apply Template dialog box.

9. BusinessObjects applies the template to the report.
#DIV/0

**Description**
Occurs when a formula performs a division by 0. For example, the formula
=Revenue/Quantity Sold
returns 20 if Revenue is 100 and Quantity Sold is 5. But if Quantity Sold is 0, then the result is #DIV/0.

**Solution**
Using an IF THEN ELSE statement, you can set up a value or text, for example, “No Sale”, which will appear when a division by zero occurs. To do this:
1. Click Formula Bar on the View menu.
2. Click inside the cell where #DIV/0 appears.
3. Write the following formula in the Formula Bar:
   =If IsError (VariableName) Then "No Sale" Else (VariableName)
4. Press the Enter key.

**TIP**
You can use an IF THEN ELSE statement such as the one in the above procedure to return default values for errors other than #DIV/0.
#ERROR

**Description**

This error occurs when the definition of a formula or a variable within a formula is incorrect. For example, the formula that returns percentages based on a measure, such as

\[
\frac{{\text{Nb Customers}}}{{\text{Sum(Nb Customers)}}}
\]

returns #ERROR if the measure, in this case Nb Customers, itself contains an error.

**Solution**

You need to break down the formula into its component parts in order to find which part contains the error. Here’s how:

1. Insert a new cell in the report by clicking **Cell** on the **Insert** menu.
2. Select the cell containing the error, then click **Copy** on the **Edit** menu.
3. Select the new cell, then click **Paste** on the **Edit** menu.
4. The formula containing the error appears in the new cell.
5. Click the new cell, then in the Formula Bar, select one part of the formula.
6. Copy and paste the selection into the cell where the error first occurred, then press Enter.
7. Repeat this step until you find the part of the formula that contains the error.
8. Fix the error, then paste the whole corrected formula back into the cell where the error first occurred.
#IERR

Occurs in complex formulas within formulas. The three common causes are:

- Formulas combining measures and dimensions, where a dimension is missing from the calculation context
- Aggregations containing multiple formulas
- Formulas with complex WHERE clauses.

TIP

When you make calculations by combining formulas, #IERR might occur because the formulas within the formulas contain errors. If none of the solutions in this section remove #IERR, try breaking down the formula into its component parts, and test each one. This procedure is described under “#ERROR” on page 455.

#IERR in a formula combining measures and dimensions

Description

BusinessObjects supports aggregate formulas that contain both dimensions and measures. For example, the following formula displays revenue for customers called Prince:

=If(<Customer>="Prince") Then <Revenue>

BusinessObjects qualifies any formula containing a measure (for example, Revenue) as a measure. This qualification requires that all dimensions are present in the calculation context - the local context if none is specified. #IERR may occur when a dimension required to compute the formula is missing from the context.

TIP

For information on calculation contexts, refer to “Calculation Contexts and Extended Syntax” on page 407.

Solution

You add the missing dimension to the context using the ForEach operator, and apply the Sum function. So, if

=If(<Customer>="Prince") Then <Revenue>

returns #IERR, the formula you need is as follows:
Accessing Data and Data Analysis

#IERR in an aggregation containing a complex formula

**Description**
An aggregation such as Min, Max etc., used on a formula that already contains a formula, may produce #IERR.

**Solution**
Turn the formula within the formula into a variable, then rewrite the whole formula using the new variable.

A variable is a formula with a name. Once you have made your variable, to include it in your formula, all you have to do is write its name, rather than a complex formula within a formula. Here’s an example.

**EXAMPLE**
Solving #IERR by turning part of a formula into a variable

You want to know the week your top ten customers placed their first order. In a table containing the list of customers, you add a column and insert the following formula:

=Min(Week(<Order Date>))

If #IERR occurs, the first thing to do, using the Variables command on the Data menu, is to create a variable called WeekOrderDate from the formula

=Week(<Order Date>)

Then, rewrite the original formula using the new variable, as follows:

=Min(<WeekOrderDate>)

Note that the original formula might work. The purpose of this example is to show how to fix #IERR should it occur in similar formulas.

#IERR in a formula using WHERE

**Description**
The WHERE operator lets you specify values of a dimension to include in a calculation. For example

=(<Revenue>*2) WHERE (<Customer>="Prince")

shows revenue at 200% for customers named Prince. #IERR can occur in WHERE clauses that contain complex formulas.
Solution

Turn conditions specified after WHERE into variables. Then, you can rewrite the whole formula, using the variables instead of the original formulas in the WHERE clause.

This is the same solution as for #IERR in an aggregation containing a complex formula. Refer to “Solving #IERR by turning part of a formula into a variable” on page 457 for more information.
#OVERFLOW

**Description**
The calculation returns a number that is too big for BusinessObjects to compute. The maximum is $17e +/\sim 308$, or 15 digits.

**Solution**
Check the maximum value of the function used in the formula. For example, the maximum value for the Fact function is 709.
The BusinessObjects online help on functions includes maximum values where appropriate.
#SYNTAX

**Description**
#SYNTAX occurs when a variable used in a formula no longer exists in the document. For example, the formula
\[ =\text{Product Price} \times \text{Quantity Sold} \]
returns #SYNTAX if the user deletes either object from the query.

The missing variable can correspond to
- an object returned by a query, or
- a local variable that you have defined in the report.

**Solution**
The way you fix this problem depends on the data available in the report:

<table>
<thead>
<tr>
<th>If the variable you need is...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available as an object in the universe you are using,</td>
<td>Edit the query (Edit Data Provider command, Data menu), add the object you need, then click Run.</td>
</tr>
<tr>
<td>A local variable that you have defined in the report,</td>
<td>Create it in the Formula Editor (Variables command, Data menu, Add button).</td>
</tr>
</tbody>
</table>
#UNKNOWN

## Description

Occurs when the object corresponding to a variable displayed in the report has been removed from the query.

When you remove an object from a query BusinessObjects does not always remove it from the report. More specifically:

<table>
<thead>
<tr>
<th>If the variable is displayed in...</th>
<th>Then BusinessObjects...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• A table or simple crosstab</td>
<td>Removes it from the report.</td>
</tr>
<tr>
<td>• A master cell</td>
<td></td>
</tr>
<tr>
<td>• A free-standing cell</td>
<td>Returns #UNKNOWN.</td>
</tr>
<tr>
<td>• A break header or footer</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

If the missing variable is used in a formula, BusinessObjects returns #SYNTAX.

## Solution

You have to add the missing object to the query. Here’s how to do it:

1. Click **Edit Data Provider** on the **Data** menu. If the List of Data Providers dialog box appears, click the query you want to edit, then click **OK**. The Query Panel appears.
2. Double-click the missing object in the Classes and Objects box. The object appears in the Result Objects box.
3. Click **Run**. BusinessObjects replaces #UNKNOWN with the variable corresponding to the object you added.

**NOTE**

If you do not want to add the missing data to your report, simply clear the cell containing the error. Select the cell, then press the Del key on your keyboard. To delete the cell, click **Delete** on the **Edit** menu.
## Tips and tricks

This section provides tips for writing formulas and avoiding #COMPUTATION, #MULTIVALEU and other errors.

<table>
<thead>
<tr>
<th>Tip</th>
<th>What to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. All formulas must begin with =</td>
<td>Type = before your formula.</td>
</tr>
<tr>
<td>If you forget =, BusinessObjects displays the formula as text!</td>
<td></td>
</tr>
</tbody>
</table>
| 2. The extended syntax of a formula shows you the contexts BusinessObjects applies by default. There are two easy ways to see the extended syntax of a formula. | 1. Click the cell containing the formula.  
2. Rest your cursor over the Formula Bar. The extended formula appears in a tooltip.  
-or-  
1. Click the cell containing the formula.  
2. Click Define as Variable on the Data menu.  
3. Click Evaluate the formula in its context. The extended formula appears in the dialog box. |
| 3. If you remove a dimension from a table, a calculation that needs that dimension can return an error. To avoid this, you can hide the dimension: it won’t appear in the report, but it will be included in the calculation. | 1. Click the table containing the dimension you want to hide.  
2. Click Table on the Format menu.  
3. In the Pivot tab, click the dimension.  
4. Click Hide, then click OK. |
<p>| 4. The best way to fix #MULTIVALEU is to include the variable returning the error in the current break. | Click Break on the Format menu. For more information, refer to Solution for #MULTIVALEU in a break header or footer on page 445 |</p>
<table>
<thead>
<tr>
<th>Tip</th>
<th>What to do</th>
</tr>
</thead>
</table>
| 5. If you get #MULTIVALE when you insert a variable in a column header of a table, try fixing it by applying a sort. | 1. Click the cell containing #MULTIVALE.  
2. Click Sort on the Insert menu. |
| 6. When you create or edit a formula in the Formula Bar or the Formula Editor, you press enter to validate the formula. If there's a syntax error, BusinessObjects tells you so and highlights the anomaly in the formula itself. This helps you fix the specific problem. | • If a variable name is highlighted in full, add parentheses and check the spelling.  
• If one end of a variable name is highlighted, add a parenthesis. |
| 7. You can edit formulas directly in the report, without using the Formula Bar or the Formula Editor. | 1. Double-click the cell containing the formula you want to edit.  
2. Edit the formula in the cell, then press Enter when you’re done. |
| 8. You can display the Formula Editor from the Formula Bar. | • Click the Formula Editor button: |
| 9. You can view and edit all the formulas displayed in a report by switching to Structure view. | • Click Structure on the View menu. |
| 10. BusinessObjects lists all the formulas you create in a document in the Data tab of the Report Manager. | Click Report Manager on the View menu. Click the Data tab and expand the Formulas folder. |
| 11. BusinessObjects also lists formulas in the Variables dialog box. In this dialog box, you can view, edit or insert any formula you have created. | 1. Click Variables on the Data menu.  
2. Double-click the Formulas folder.  
3. To edit a formula, select it then click Edit.  
4. To insert a formula, select it then click Insert or Replace. |
Formulas, Local Variables and Functions
Overview

This chapter gives you an introduction to using BusinessObjects formulas, local variables and functions. It explains how to set up your own formulas and variables in BusinessObjects reports using BusinessObjects syntax, how to use BusinessObjects functions, and also includes several examples of business calculations.

Calculation Contexts and Extended Syntax on page 407 describes how to build more powerful calculations by using extended syntax to define calculation contexts. This chapter also has a calculations Quick reference on page 433 that recaps some of the key BusinessObjects terms.

Calculation Troubleshooting on page 437 gives solutions to computation errors that may occur in BusinessObjects when you are writing or using formulas.
Formulas

When you run a query, BusinessObjects makes calculations on the data at the query level and returns the results as variables. You can also make calculations on report data using the built-in calculations available on the calculations menu or toolbar. This section explains how you can set up your own personalized calculations on data in your reports by writing BusinessObjects formulas.

Why use formulas?

You use formulas to carry out calculations locally in the report to set conditions on filters and data display.

A BusinessObjects formula is made up of functions, variables, and operators and always begins with an "equal to" sign. The examples below show two very simple formulas:

- \( \text{Sum} (<\text{Sales Revenue}> \) \\
- \( <\text{Margin}> / <\text{Sales revenue}> \times 100\% \)

Making local calculations

You might want to carry out personal calculations in your reports or compare database figures to figures from a spreadsheet for example. Writing a formula allows you to do this.

In addition, there are certain types of calculation that some SQL servers cannot carry out. For example, standard SQL does not allow you to use decision logic such as the IF statement.

Setting conditions

You can use BusinessObjects formulas to set conditions. For example, you can set up a condition to hide sections in a report if the sales revenue falls below a certain level.

You can also set up conditional filters. For example, you want to display just the outlets that have generated weekly revenue above a certain sum. By setting a filter with a condition, when you refresh the report with the new weekly data, only those outlets that satisfy the condition will be displayed in the report.

You can use conditions in Broadcast Agent when you are setting up reports for scheduled processing to specify under what conditions your documents should be processed.
Creating formulas

There are three ways of creating formulas. You can:
  • type your formula directly into a cell
  • type your formula in the Formula Bar
  • use the Formula Editor

Displaying the Formula Bar

Once you are familiar with BusinessObjects syntax you can type simple formulas directly into a cell or into the Formula Bar. Using the Formula Bar allows you to see more clearly what you are doing as the whole formula is displayed more easily than in a cell. If the Formula Bar is not displayed:
  • Click Formula Bar on the View menu.

Displaying the Formula Editor

If you are not familiar with BusinessObjects syntax or are writing more complex formulas, the best method is to use the Formula Editor.

1. Click inside the cell where you want the formula to appear.
   - If the Formula Bar is open, click Formula Editor on the Formula Bar or click Edit Formula on the Data menu.

   The Formula Editor appears.
Using the Formula Editor

The Formula Editor allows you to create your formula by choosing variables, functions and operators directly from the lists.

The Formula Editor has four main parts:

Formulas box
Displays the formula. You use this box to write or edit formulas.

Variables box
Displays a list of all the variables in the document, which can be local variables or variables returned by the data provider. You can include these variables in your formula.

Functions
Displays a list of all BusinessObjects functions.

Operators
Operators define the relationship between elements in a formula. Operators include mathematical operators such as addition (+) and division (/), relational operators such as greater than (>) and between, logical operators such as If
Then Else and context operators such as ForAll, ForEach and In. The list of operators in this window is updated as you add elements to the formula so that only the operators that are compatible with the current syntax are displayed.

To understand how the Formula Editor works, let's look at an example.

**EXAMPLE**

**Calculating a running total**

In the table below you want to display a running total for the monthly sales revenue.

<table>
<thead>
<tr>
<th>Month</th>
<th>Sales revenue</th>
<th>Running Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>$3,840,310</td>
<td>$3,840,310</td>
</tr>
<tr>
<td>February</td>
<td>$2,102,538</td>
<td>$5,942,848</td>
</tr>
<tr>
<td>March</td>
<td>$2,287,012</td>
<td>$8,229,061</td>
</tr>
<tr>
<td>April</td>
<td>$2,165,680</td>
<td>$12,395,759</td>
</tr>
<tr>
<td>May</td>
<td>$3,561,847</td>
<td>$16,477,606</td>
</tr>
<tr>
<td>June</td>
<td>$2,378,516</td>
<td>$19,856,122</td>
</tr>
<tr>
<td>July</td>
<td>$2,576,172</td>
<td>$21,431,294</td>
</tr>
<tr>
<td>August</td>
<td>$1,564,215</td>
<td>$22,995,509</td>
</tr>
<tr>
<td>September</td>
<td>$4,081,152</td>
<td>$27,076,451</td>
</tr>
<tr>
<td>October</td>
<td>$3,931,396</td>
<td>$30,967,840</td>
</tr>
<tr>
<td>November</td>
<td>$2,639,086</td>
<td>$33,606,576</td>
</tr>
<tr>
<td>December</td>
<td>$3,080,225</td>
<td>$36,687,250</td>
</tr>
<tr>
<td>Sum</td>
<td>$36,307,201</td>
<td>$39,840,310</td>
</tr>
</tbody>
</table>

To do this:

1. Insert a new column after the Sales Revenue column and name it Running total.
2. Click inside the Running total column. This is where you are going to insert the formula.
3. Click **Edit Formula** on the **Data** menu. The Formula Editor appears.
4. Double-click the equal sign (=) sign in the Operators list. The equal (=) sign is displayed in the Formulas box.
5. In the Functions box, open the All functions & aggregates folder.
6. Scroll down until the function RunningSum is displayed in the Functions box.
7. Double-click **RunningSum**. BusinessObjects displays RunningSum in the Formulas box. Notice that the cursor is inside the brackets. This is where BusinessObjects will insert the
variable.

8. In the Variables box, double-click **Sales Revenue**.
   Sales revenue is added to the formula. Notice the angle brackets which are automatically added to a variable by the Formula Editor.

9. Click **OK**.
   The Formula Editor closes and the calculation is displayed in the RunningSum column.

   BusinessObjects stores the formula in the Formulas folder of the Report Manager Data tab.

**Guidelines on the syntax to use in formulas**

Whichever method you use to write formulas, always bear these guidelines in mind:

- A formula must begin with an "equal to" sign. If you remove this sign, the formula is considered as a constant or as text.
- Variables included in formulas must be surrounded by a less than sign (<) and a greater than sign (>). For example, <Revenue>.
- Text included in formulas must be surrounded by double-quotes (")

**Syntax errors**

If there is a syntax error in a formula, an error message is displayed and the part of the formula that contains the error is highlighted.
Local variables

A local variable is a named formula. Local variables appear in the list of variables in the Report Manager Data tab; you can use them to build tables, charts and crosstabs in the same way as you use variables returned by a data provider.

Why use local variables?

Variables have a number of advantages over formulas because there are some things you cannot do using formulas alone:

- You cannot apply alerters, filters, sorts and breaks on columns or rows containing formulas, but you can on those containing variables.
- You can include variables qualified as dimensions in drill hierarchies.

Local variables are also useful because:

- You can re-use them easily in the same document.
- Formulas can be complex. You can use (and reuse) variables in formulas to simplify them. Because you can re-use variables, you don’t need to type the same formulas over and over again. Variables make complex formulas easier to decipher because they break the formulas up into manageable components. See Determining the first and last days of the previous month on page 496 for an example.
How to recognize local variables

If you want to know whether a variable has been returned by a data provider or created locally in a report:

- Right-click on the variable in the list in the Data tab of the Report Manager. If the Edit Variable command is available in the shortcut menu, the variable is a local variable.

- If the Edit Variable is not available (grayed out) the variable was returned by the data provider and cannot be edited.

Creating a local variable

You can create local variables using the variable editor, by turning an existing formula into a variable, and by grouping values from an existing variable to create a new one.

➤ Using the Variable Editor

You can create a local variable using the Variable Editor. To do this:

1. Right-click on the Data tab of the Report Manager and click New Variable on the shortcut menu.
   The Variables dialog box opens.
2. Click the Formula tab.
3. Type the formula in the Formula box, or double-click the function(s),
variable(s) or operator(s) you need.

4. Click the **Definition** tab and type a name in the Name box.

5. In the Qualification box, choose whether you want the local variable to be defined as a Dimension, Measure or Detail object.

6. Click **OK**.

The new variable is displayed in the list of variables in the Report Manager data tab.

### Transforming a formula into a local variable

You can also transform an existing formula into a local variable. This allows you to apply alerters, filters, and sorts. After setting up a formula in a report, you might decide that you would like to highlight the top values. To do this, you transform the formula you have created into a variable.

#### EXAMPLE

**Highlighting above-average margin**

You have set up a formula to calculate average margin and now decide you want to highlight all those stores that have made above average margin. In BusinessObjects you can highlight data in this way using alerters but you cannot
apply an alerter on a column or row of data that contains a formula. However, you can use an alerter on a variable. By turning your formula into a variable you can highlight your above-average stores.

To transform a formula into a variable:

1. Select the column of data that contains the formula.
2. Click **Define As Variable** on the **Data** menu.
3. Type in a name for the variable in the Define the Variable dialog box.
4. Set how you want the formula to be defined.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Choose...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrict the definition of this variable to its context in the current block.</td>
<td>Evaluate the formula in its context. If you insert this variable in another block in the report, the result of that calculation will always be based on the original context</td>
</tr>
<tr>
<td>Allows you to define the variable so that it calculates the data dynamically, based on the context of the block where the variable is inserted</td>
<td>Keep the formula generic</td>
</tr>
</tbody>
</table>

5. Click **OK**.

The variable is now displayed in the variables folder in the Report Manager Data tab and you can set up an alerter using the variable.

For further information on calculation contexts, see "Calculation Contexts and Extended Syntax" on page 407.
Creating local variables by grouping values

You can also create new variables by grouping the values of existing variables which enables you to create dynamic groups for analysis purposes. Grouping values prior to analyzing your data in drill mode is useful, for example, if you have a variable that has many values. It allows you to create an intermediary level of detail in your analysis.

**EXAMPLE**

Group quarters to display revenue per semester

You have a report showing the sales revenue per quarter, as follows:

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>$9,729.86</td>
</tr>
<tr>
<td>Q2</td>
<td>$9,126.27</td>
</tr>
<tr>
<td>Q3</td>
<td>$8,200.63</td>
</tr>
<tr>
<td>Q4</td>
<td>$6,500.74</td>
</tr>
</tbody>
</table>

You want to display revenue per semester. You do this by grouping the quarters. The Quarter variable returns four values - Q1, Q2, Q3, Q4. By grouping Q1 and Q2 into one value (H1), and Q3 and Q4 into a second value (H2) you create a new variable, Semester, and then calculate revenue by semester. You then add Semester to a drill hierarchy and use it in your analysis in drill mode.

To do this:
1. Select Q1 in the table.
2. Holding down the Ctrl key, select Q2.
3. Click **Group Values** on the Report toolbar. The Rename grouped values box opens.
4. Type in the new name, H1, in the Rename Group box and click **OK**. BusinessObjects groups Q1 and Q2 together and displays the new name, H1,
Creating local variables by grouping values in the table.

<table>
<thead>
<tr>
<th>Quarter + In Table</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>$19,056,372</td>
</tr>
<tr>
<td>Q2</td>
<td>$10,500,628</td>
</tr>
<tr>
<td>Q4</td>
<td>$7,330,742</td>
</tr>
</tbody>
</table>

BusinessObjects creates a new variable and displays it in the Report Manager list. BusinessObjects updates the table column header with the name of the new variable, Quarter+ In Table 1.

5. Repeat Step 1 to Step 4 to group the values for Q3 and Q4 and name the new value H2.

The table now looks like this.

<table>
<thead>
<tr>
<th>Quarter + In Table 1</th>
<th>Sales revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>$6,106,023</td>
</tr>
<tr>
<td>H2</td>
<td>$7,066,420</td>
</tr>
</tbody>
</table>
You can rename the new variable to make its name more meaningful. To do this:

1. Right-click on the Quarter+ In Table 1 variable in the Report Manager window.
2. Click **Edit Variable** on the shortcut menu.
   The Edit box opens.

3. Type in a new name, Semester, in the Name of the Variable text box and click **OK**.
   The new name, Semester, is displayed in the Variables list and in the table.
   Your table now displays sales revenue per semester.
Editing grouped values
You can edit variables created by grouping values from another variable. You can rename the variable or the values, re-arrange the values in the groups or create a new group of values. To do this:
1. Right-click on the variable in the Report Manager window.
2. Click **Edit Variable** on the shortcut menu.
   The Edit box opens.
3. Make the required changes and click **OK**.

Deleting grouped values
You can delete a variable created by grouping values from another variable.
1. Click **Variables** on the **Data** menu.
2. In the Variables box, select the variable you want to delete.
3. Click **Remove** and then **OK**.

Ungrouping grouped values
You can ungroup a variable created by grouping values from another variable.
1. Select the grouped value in the table.
2. Click **Group Values** on the toolbar.
Adding grouped values to a drill hierarchy

You can add a local variable that has been qualified as a dimension to a drill hierarchy and then use it in your analysis in drill mode. You can now add Semester to the Time period hierarchy and then drill down from Year to Semester and then to Quarter.

To add Semester to the drill hierarchy:
1. Click Hierarchies on the Analysis menu.
   The Hierarchy Editor appears.
2. Add Semester to the Time period hierarchy between Year and Quarter.

**NOTE**
For more information on setting up and using drill mode, see Analyzing Data in Drill Mode on page 237.
Managing formulas and local variables

This section describes how to insert, edit and delete the local variables and formulas you have created.

Inserting local variables and formulas in a report

You can drag the variable or formula from the Data tab of the Report Manager window and drop it where you want it to appear on the report.

Editing formulas

You can edit a formula directly in the cell or in the Formula Bar or you can use the Formula Editor.

1. Click the cell that displays the result of the formula you want to edit.
2. Then:
   - Type your changes into the cell and press Enter to validate them
   - Click inside the Formula Bar, and edit the formula and click Validate Formula to validate the formula.
   - Click Edit Formula on the Data menu, make your changes in the Formula Editor and click OK to validate them.

BusinessObjects displays the result of the formula in the cell.

Editing local variables

You can edit a local variable directly from the Report Manager. To do this:

1. Right-click the variable in the Data tab of the Report Manager window.
2. Click Edit Variable on the shortcut menu.
   - The Variable Editor appears.
3. You can:
   - type in a new name in the Name box
   - change the qualification of the variable
   - edit the formula on the Formula tab
4. Click OK to validate your changes
Deleting formulas and local variables

To delete a formula or local variable from a report:

1. Click **Variables** on the **Data** menu.
   The Variables dialog box appears:

   ![Variables dialog box]

   The Edit and Remove buttons are not available if you choose a variable returned by the data provider.

2. Open the Variables or the Formulas folder.
3. Select the variable or formula to delete.
4. Click **Remove** and then click **Close**.

   **NOTE**

   You can delete local variables only, not variables returned by a data provider.
Functions

BusinessObjects contains many built-in functions which greatly extend its capabilities. Functions are pre-defined formulas. A function consists of the function name followed by a pair of parentheses. The parentheses can contain arguments and arguments supply functions with values on which to operate. Arguments can be objects, constants or other functions.

Some commonly used functions are available directly from the BusinessObjects menus and toolbars. When you choose one of these functions, the necessary arguments are added automatically.

The entire set of functions is organized in nine folders in the Functions box of the Formula Editor:

<table>
<thead>
<tr>
<th>The folder...</th>
<th>Lists...</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>All functions in alphabetical order.</td>
</tr>
<tr>
<td>Aggregates</td>
<td>Functions that return aggregate totals (for example sums or averages).</td>
</tr>
<tr>
<td>Numeric Functions</td>
<td>Functions that operate on numerical arguments.</td>
</tr>
<tr>
<td>Character Functions</td>
<td>Functions that operate on character objects and text strings.</td>
</tr>
<tr>
<td>Date Functions</td>
<td>Functions that operate on dates.</td>
</tr>
<tr>
<td>Logical Functions</td>
<td>Functions that return true or false.</td>
</tr>
<tr>
<td>Document Functions</td>
<td>Functions that return information about a document.</td>
</tr>
<tr>
<td>Data Provider Functions</td>
<td>Functions that return information about a data provider.</td>
</tr>
<tr>
<td>Miscellaneous Functions</td>
<td>Functions that cannot be categorized into one of the seven category folders.</td>
</tr>
</tbody>
</table>

Using Functions

When you select a function in the list in the Functions window, a description of the syntax the function requires is displayed at the bottom of the Formula Editor window.
When you add a function to the Formula window, any necessary commas are added. The following example shows you how to use a BusinessObjects function to rank data according to revenue generated.

**EXAMPLE**

**Ranking cities according to sales revenue**

In this example, you want to rank the cities in your table according to sales revenue generated this quarter.

<table>
<thead>
<tr>
<th>City</th>
<th>Sales revenue</th>
<th>Revenue Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>$2,699,973</td>
<td>7</td>
</tr>
<tr>
<td>Boston</td>
<td>$1,385,767</td>
<td>11</td>
</tr>
<tr>
<td>Chicago</td>
<td>$3,022,000</td>
<td>6</td>
</tr>
<tr>
<td>Colorado Springs</td>
<td>$2,660,775</td>
<td>8</td>
</tr>
<tr>
<td>Dallas</td>
<td>$1,970,034</td>
<td>9</td>
</tr>
<tr>
<td>Houston</td>
<td>$6,447,907</td>
<td>2</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>$4,220,000</td>
<td>3</td>
</tr>
<tr>
<td>Miami</td>
<td>$1,878,159</td>
<td>10</td>
</tr>
<tr>
<td>New York</td>
<td>$7,552,021</td>
<td>1</td>
</tr>
<tr>
<td>San Francisco</td>
<td>$3,258,641</td>
<td>4</td>
</tr>
<tr>
<td>Washington</td>
<td>$2,361,950</td>
<td>6</td>
</tr>
</tbody>
</table>

To do this:

1. Add a new column to the table and name it Revenue Rank. Click in the Revenue rank column.
2. Click **Edit Formula** on the Data menu. The Formula Editor appears.
3. Open the All functions and aggregates folder and scroll down to the Rank function.
4. Double-click on Rank.

![BusinessObjects adds parentheses and commas automatically.](image)

BusinessObjects adds the Rank function to the Formula window.
To use the Rank function, you need to add a dimension variable, a comma and then a measure variable inside the parentheses. The function then ranks the dimension according to the measure.

5. In the Variables list, double-click City and then Sales revenue.

BusinessObjects displays the completed syntax in the formulas window.

=Rank(<City>,<Sales revenue>)

6. Click **OK**.

The cities are ranked from 1 to 10 according to sales revenue.

**TIP**

To find a function quickly in the Functions list, open the All functions & aggregates folder and then type the first letter of the function you want to find. BusinessObjects highlights the first function beginning with that letter.
Using the function help

Before working with a function for the first time, use the Function Help to find out what the function does. To do this:

- Select the function in the Functions list and click the Function Help button.

The help page gives a description of the function, its syntax, and an example.
Function equivalents in Microsoft Excel

Many BusinessObjects functions have equivalents in Microsoft Excel, or equivalent formulas that use different Excel functions to produce the same result. The following table cross-references BusinessObjects functions to Excel functions or formulas.

**Aggregate function equivalents**

<table>
<thead>
<tr>
<th>BusinessObjects</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Sum</td>
</tr>
<tr>
<td>Count</td>
<td>Count</td>
</tr>
<tr>
<td>CountAll</td>
<td>CountA</td>
</tr>
<tr>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>Min</td>
<td>Min</td>
</tr>
<tr>
<td>Max</td>
<td>Max</td>
</tr>
<tr>
<td>StDev</td>
<td>StDev</td>
</tr>
<tr>
<td>StDevP</td>
<td>StDevP</td>
</tr>
<tr>
<td>Var</td>
<td>Var</td>
</tr>
<tr>
<td>VarP</td>
<td>VarP</td>
</tr>
</tbody>
</table>

**Numeric function equivalents**

<table>
<thead>
<tr>
<th>BusinessObjects</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs</td>
<td>Abs</td>
</tr>
<tr>
<td>Ceiling</td>
<td>Ceiling</td>
</tr>
<tr>
<td>Cos</td>
<td>Cos</td>
</tr>
<tr>
<td>EuroToRoundErr</td>
<td>Round(Round(Value / conversionRate, 2) - (Value / conversionRate), 3)</td>
</tr>
<tr>
<td>EuroFromRoundErr</td>
<td>Round(Round(Value * conversionRate, 2) - (Value * conversionRate), 3)</td>
</tr>
</tbody>
</table>
## BusinessObjects Excel equivalent

<table>
<thead>
<tr>
<th>BusinessObjects</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>EuroConvertTo</td>
<td>Round(Round(Value / conversionRate, 2), 2)</td>
</tr>
<tr>
<td>EuroConvertFrom</td>
<td>Round(Round(Value * conversionRate, 2), 2)</td>
</tr>
<tr>
<td>Exp</td>
<td>Round(Exp(Value), 2)</td>
</tr>
<tr>
<td>Fact</td>
<td>Fact</td>
</tr>
<tr>
<td>Floor</td>
<td>Floor</td>
</tr>
<tr>
<td>Ln</td>
<td>Round(Ln(Value), 2)</td>
</tr>
<tr>
<td>Log</td>
<td>Log</td>
</tr>
<tr>
<td>Log10</td>
<td>Log10</td>
</tr>
<tr>
<td>Median</td>
<td>Median</td>
</tr>
<tr>
<td>Mod</td>
<td>Mod</td>
</tr>
<tr>
<td>Power</td>
<td>Power</td>
</tr>
<tr>
<td>Round</td>
<td>Round</td>
</tr>
<tr>
<td>Sign</td>
<td>Sgn</td>
</tr>
<tr>
<td>Sin</td>
<td>Round(Sin(Value), 2)</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Sqr</td>
</tr>
<tr>
<td>Tan</td>
<td>Round(Tan(Value), 2)</td>
</tr>
<tr>
<td>Truncate</td>
<td>Round(Value, 0)</td>
</tr>
</tbody>
</table>

### Character function equivalents

<table>
<thead>
<tr>
<th>BusinessObjects</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordCap</td>
<td>StrConv(String, vbProperCase)</td>
</tr>
<tr>
<td>Upper</td>
<td>StrConv(String, vbUpperCase)</td>
</tr>
<tr>
<td>Trim</td>
<td>Trim</td>
</tr>
<tr>
<td>Substr</td>
<td>Mid</td>
</tr>
<tr>
<td>RightTrim</td>
<td>RTrim</td>
</tr>
<tr>
<td>Right</td>
<td>Right</td>
</tr>
<tr>
<td>Replace</td>
<td>Replace</td>
</tr>
</tbody>
</table>
### Date function equivalents

<table>
<thead>
<tr>
<th>BusinessObjects</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Year</td>
</tr>
<tr>
<td>Week</td>
<td>DatePart(ww, InputDate)</td>
</tr>
<tr>
<td>ToDate</td>
<td>Format(InputDate, date_format)</td>
</tr>
<tr>
<td>RelativeDate</td>
<td>DateAdd(d, numberOfDays, InputDate)</td>
</tr>
<tr>
<td>Quarter</td>
<td>DatePart(q, InputDate)</td>
</tr>
<tr>
<td>MonthNumberOfYear</td>
<td>Month(InputDate)</td>
</tr>
<tr>
<td>Month</td>
<td>MonthName(Month(InputDate))</td>
</tr>
</tbody>
</table>

### BusinessObjects

<table>
<thead>
<tr>
<th>BusinessObjects</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos</td>
<td>Instr</td>
</tr>
<tr>
<td>Match</td>
<td>If StrComp(Value, MatchPattern, vbTextCompare) Then True Else False End If</td>
</tr>
<tr>
<td>Lower</td>
<td>StrConv(String, vbLowerCase)</td>
</tr>
<tr>
<td>Length</td>
<td>Len</td>
</tr>
<tr>
<td>LeftTrim</td>
<td>LTrim</td>
</tr>
<tr>
<td>Left</td>
<td>Left</td>
</tr>
<tr>
<td>InitCap</td>
<td>InitCap</td>
</tr>
<tr>
<td>FormatNumber</td>
<td>FormatNumber</td>
</tr>
<tr>
<td>FormatDateF</td>
<td>Format</td>
</tr>
<tr>
<td>Fill</td>
<td></td>
</tr>
<tr>
<td>Concatenation</td>
<td></td>
</tr>
<tr>
<td>Char</td>
<td>Chr</td>
</tr>
<tr>
<td>Asc</td>
<td>Asc</td>
</tr>
</tbody>
</table>

Function equivalents in Microsoft Excel
More examples of using formulas

There are examples of using functions and writing formulas throughout this chapter and throughout this user’s guide. This section has several additional examples on using the BusinessObjects formula editor and BusinessObjects functions to set up personal calculations.

EXAMPLE
Calculating a 3-week rolling average

Using a rolling average smooths out the fluctuations of a measure variable that fluctuates over time, for example stock prices, which change daily. A rolling average is obtained by calculating the average of the current value and the specified number of previous values. In BusinessObjects, you use the Previous() function to set up a rolling average.

This example shows you how to create the variable to calculate a three-week rolling average for sales revenue.

1. Right-click on any variable in the Report Manager Data tab and click New Variable on the shortcut menu.
   The Variable Editor appears.
2. Click the Definition tab.
3. In the Name box, type “3 weeks rolling”.
4. Under Qualification, choose Measure.
5. Click the Formula tab.
6. In the Formulas box, type the formula:
   \[
   \frac{(<\text{Sales revenue}> + \text{Previous(<Sales revenue>)} + \text{Previous(Previous(<Sales revenue>))})}{3}.
   \]
   This formula adds the sales revenue for the current week to the sales revenue for the two previous weeks and then divides the total by three to obtain an average for those three weeks. To create this formula, we use the BusinessObjects Previous() function.
7. Click OK.
   The new variable is added to the list in the Report Manager Data tab and you can use it in the tables and charts in your report.
Combining data in a single cell

You frequently need to combine different pieces of data returned by data providers in a single cell of a report. For example, first and last names are typically stored as separate pieces of data in the database but you often need to display a person's whole name in a single cell of a report.

There are two ways of combining pieces of data or combining text and data in a single cell in BusinessObjects. You can either use the Concatenation() function or the & operator.

The examples below describe how to use both methods to combine character-type data, and how to combine numbers and dates with text or with other pieces of data.

**EXAMPLE**

Combining first and last names in a single cell

The BusinessObjects Concatenation() function allows you to combine two character strings. The character string may be a piece of text or a character-type variable. The syntax for this function is:

```
Concatenation(character string, character string)
```

To display a person's first and last name in a single cell, the following syntax:

```
=Concatenation(<First Name>,<Last Name>)
```

will give the following result: JohnGardner

You would typically wish to have a space between the first and last name. To do this, you need to use the following syntax:

```
=Concatenation(<First Name>,(Concatenation(" ",<Last Name>)))
```

You can also use the concatenation operator (&) to concatenate strings. If you are combining several character strings and want to add spaces, using the & operator is a simpler solution than using the Concatenation() function. The syntax to achieve the same result as shown above using the & operator is:

```
=<First Name>" ",&<Last Name>
```

Notice that the space you typed is surrounded by quote marks.
EXAMPLE

Combining text and data in a single cell

You can use the same syntax as in the example above to add a comment to data in a table cell or a master cell. The table below shows sales figures for Austin for Q1 2001, but quarter is not displayed in the table. By editing the master cell contents, you can add a more explanatory title for each section in the report as shown below:

<table>
<thead>
<tr>
<th>Line</th>
<th>City</th>
<th>Sales revenue</th>
<th>Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories Austin</td>
<td>$92,204</td>
<td>$37,503</td>
<td></td>
</tr>
<tr>
<td>City Shoes Austin</td>
<td>$162</td>
<td>$85</td>
<td></td>
</tr>
<tr>
<td>City Trousers Austin</td>
<td>$1,651</td>
<td>$183</td>
<td></td>
</tr>
<tr>
<td>Dresses Austin</td>
<td>$13,112</td>
<td>$6,036</td>
<td></td>
</tr>
<tr>
<td>Jackets Austin</td>
<td>$2,550</td>
<td>$967</td>
<td></td>
</tr>
<tr>
<td>Leather Austin</td>
<td>$966</td>
<td>$311</td>
<td></td>
</tr>
<tr>
<td>Outerwear Austin</td>
<td>$2,616</td>
<td>$1,512</td>
<td></td>
</tr>
<tr>
<td>Overcoats Austin</td>
<td>$596</td>
<td>$37</td>
<td></td>
</tr>
<tr>
<td>Shirt Ward Austin</td>
<td>$18,973</td>
<td>$9,225</td>
<td></td>
</tr>
<tr>
<td>Sweaters Austin</td>
<td>$16,595</td>
<td>$9,111</td>
<td></td>
</tr>
<tr>
<td>Sweat-T-Shirts Austin</td>
<td>$45,606</td>
<td>$22,477</td>
<td></td>
</tr>
<tr>
<td>Trousers Austin</td>
<td>$1,294</td>
<td>$469</td>
<td></td>
</tr>
</tbody>
</table>

To obtain the result illustrated above, the formula:

`='Sales for Q1 in '&<City>`

will give you the following result:

Sales for Q1 in Austin

Notice that text is surrounded by quote marks and that we typed a space at the end of the text and before the final quote marks to add a space between the text and the name of the city.
If you want to place a filter on the City section after combining text and data in the master cell, you’ll notice that the Insert Filter button on the Standard toolbar is unavailable. To insert a filter:
1. Select the master cell.
2. Click Filters on the Format menu.
   The Filters dialog box appears.
3. Click Add.
   The Variables to Filter dialog box appears.
4. Select City and click OK to return to the Filters dialog box.
In the Values box, select the cities you want to filter and click OK.

EXAMPLE
Combining text and numbers in a single cell
The Concatenation() function and & operator allow you to combine character-type data only. If you want to combine text or character-type data with numbers you must first convert the number into a character string. Otherwise, BusinessObjects displays the error message 'Incompatible data type'. You can convert a number to a character string using the FormatNumber() function.

<table>
<thead>
<tr>
<th>Lines</th>
<th>City</th>
<th>Sales revenue</th>
<th>Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>City Trouser</td>
<td>Austin</td>
<td>$832</td>
<td>$182</td>
</tr>
<tr>
<td>Dresses</td>
<td>Austin</td>
<td>$690</td>
<td>$357</td>
</tr>
<tr>
<td>Jackets</td>
<td>Austin</td>
<td>$611</td>
<td>$295</td>
</tr>
<tr>
<td>Leather</td>
<td>Austin</td>
<td>$456</td>
<td>$240</td>
</tr>
<tr>
<td>Outerwear</td>
<td>Austin</td>
<td>$246</td>
<td>$96</td>
</tr>
<tr>
<td>Shirt Wear</td>
<td>Austin</td>
<td>$2,712</td>
<td>$1,127</td>
</tr>
<tr>
<td>Sweaters</td>
<td>Austin</td>
<td>$182</td>
<td>$96</td>
</tr>
<tr>
<td>Sweat T-Shirts</td>
<td>Austin</td>
<td>$2,416</td>
<td>$1,249</td>
</tr>
</tbody>
</table>

To obtain the result illustrated above, the formula:
"Revenue for week: "&FormatNumber(<Week>,"0")
will give you the following result:
Revenue for week: 1
EXAMPLE

Combining text and dates in a single cell

In the same way, if you want to combine text with dates using the & operator or the Concatenation() function, you must first convert the date into a character string. Otherwise, BusinessObjects displays the error message 'Incompatible data type'. You can convert a date into a character string using the FormatDate() function.

Invoice Date: 01/01/1998

<table>
<thead>
<tr>
<th>Customer</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>McCarthy</td>
<td>27 Pasadena Drive</td>
</tr>
</tbody>
</table>

To obtain the result illustrated above, the formula:

`"Invoice date: "&FormatDate(<Date>, "dd/mm/yyyy")`

will give you the following result

Invoice date: 01/01/1998

EXAMPLE

Comparing yearly margin growth using the Where function

A common requirement in business is to compare data from different dates or periods so you can evaluate how key indicators such as revenue and margin have progressed. The BusinessObjects Where function allows you to identify data with the values of another variable so that you can compare related data.
In the example below, you want to compare yearly margin. Your BusinessObjects document contains the variables for year, margin and city. Using this data, you can write a BusinessObjects formula to assign margin to a specific year and then calculate margin growth from one year to the next.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>$283,804</td>
<td>$362,086</td>
<td>$425,790</td>
<td>50.7%</td>
<td>11.2%</td>
</tr>
<tr>
<td>Boston</td>
<td>$111,665</td>
<td>$93,857</td>
<td>$536,574</td>
<td>-42.9%</td>
<td>-42.8%</td>
</tr>
<tr>
<td>Chicago</td>
<td>$349,760</td>
<td>$465,478</td>
<td>$459,866</td>
<td>23.3%</td>
<td>-5.6%</td>
</tr>
<tr>
<td>Colorado Springs</td>
<td>$203,701</td>
<td>$254,483</td>
<td>$309,606</td>
<td>44.6%</td>
<td>5.3%</td>
</tr>
<tr>
<td>Dallas</td>
<td>$109,064</td>
<td>$279,052</td>
<td>$236,146</td>
<td>47.9%</td>
<td>2.3%</td>
</tr>
<tr>
<td>Houston</td>
<td>$960,513</td>
<td>$707,854</td>
<td>$895,542</td>
<td>40.3%</td>
<td>7.2%</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>$944,247</td>
<td>$934,710</td>
<td>$519,300</td>
<td>36.1%</td>
<td>2.4%</td>
</tr>
<tr>
<td>Miami</td>
<td>$120,679</td>
<td>$206,670</td>
<td>$130,122</td>
<td>38.6%</td>
<td>10.9%</td>
</tr>
<tr>
<td>New York</td>
<td>$179,317</td>
<td>$1,104,277</td>
<td>$1,151,106</td>
<td>41.7%</td>
<td>7.7%</td>
</tr>
<tr>
<td>San Francisco</td>
<td>$310,148</td>
<td>$471,148</td>
<td>$522,121</td>
<td>42.7%</td>
<td>8.4%</td>
</tr>
<tr>
<td>Washington</td>
<td>$310,355</td>
<td>$457,231</td>
<td>$305,416</td>
<td>47.3%</td>
<td>-15.7%</td>
</tr>
</tbody>
</table>

To do this:
1. Right-click on any variable in the list of variables in the Report Manager Data tab and click New Variable on the shortcut menu.

The Variable Editor appears.

2. Click the Definition tab.
3. In the Name box, type 2001 Margin.
4. In the Qualification section, choose Measure.
5. Click the Formula tab.
6. In the Formulas box, type the formula:
   =<Margin> Where (<Year>="2001")

This formula calculates the margin for the year 2001 only. Note that we use the BusinessObjects Where operator in the formula to specify the year.
7. Click OK.

BusinessObjects adds the new variable to the list in the Report Manager Data tab.
8. Repeat the procedure above to create one variable called 2002 Margin and one called 2003 Margin using the following syntax:
   =<Margin> Where (<Year>="2002")
   =<Margin> Where (<Year>="2003")

BusinessObjects adds the new variables to the list in the Report Manager Data tab.
You could have simply created a formula to calculate the margin for each year. The advantage of creating a variable is that you can then re-use it more easily in other formulas. For example, you can now easily calculate the percent increase in margin between the 2001 and 2002 using the variables you have just created and display the increase in a new column in the table. The formula for the 2001-2002 growth is as follows:

```excel
=FormatNumber((((<2002 Margin> - <2001 Margin>) / <2001 Margin>) * 100) ,"0.0") & 
```

and the formula for 2002-2003 growth is:

```excel
=FormatNumber((((<2003 Margin> - <2002 Margin>) / <2002 Margin>) * 100) ,"0.0") & 
```

**EXAMPLE**

Using function output as input to another function

You can use the output of a function as the input to another function. In this way you can combine functions to create complex formulas. For example, the UniverseName() function returns a string containing the name of a universe on which a data provider is based. You use this function by supplying the the name of the data provider as a string argument, for example:

```
UniverseName('Sales')
```

The problem with hard-coding a data provider name in this way is that, if the data provider name is changed, the function will no longer work.

You solve this by using the output of the DataProvider() function as input to the UniverseName() function. DataProvider() takes a variable as input, so

```
DataProvider(<Sale Date>)
```

returns the name of the data provider of the Sale Date variable. As a result, the formula

```
UniverseName(DataProvider(<Sale Date>))
```

always returns the universe name, even if the data provider name is changed.

**EXAMPLE**

Determining the first and last days of the previous month

It is common to run reports against the last complete month’s data. In order to do this you need to determine the first and last days of the previous month. You do this by creating variables that return these dates and then using these variables in your report. These variables combine numerous functions and use function output as the input to other functions.
First, create a variable that returns the date of the first day in the previous month as a string in the form “YYYYMMDD” (for example “20020601”).

To do this:
1. Click **Variables** on the **Data** menu.
   The Variables dialog box appears.
2. Click **Add**.
   The Variable Editor appears.
3. Type “FirstDayOfPrevMonthAsString” in the Name box.
4. Click the **Formula** tab.
5. Type the formula
   \[ \text{FormatNumber(Year(CurrentDate()),"0000")} \& \text{FormatNumber(MonthNumberOfYear(CurrentDate()) - 1 ,"00")} \& \text{"01"} \]
6. Click **OK** to close the Variable Editor.
7. Click **OK** to close the Variables dialog box.

This formula shows how you can combine multiple functions, using the output of functions as the input to other functions to create complex formulas. The table below breaks down the formula.

<table>
<thead>
<tr>
<th>Function</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CurrentDate()</code></td>
<td>The current date</td>
</tr>
<tr>
<td><code>Year(CurrentDate())</code></td>
<td>The current date’s year as an integer</td>
</tr>
<tr>
<td><code>MonthNumberOfYear(CurrentDate())</code></td>
<td>The current date’s month as an integer (Subtract 1 from this to give the previous month as an integer)</td>
</tr>
<tr>
<td><code>FormatNumber(Year(CurrentDate()),’0000’)</code></td>
<td>The current date’s year as a string</td>
</tr>
<tr>
<td><code>FormatNumber(MonthNumberOfYear(CurrentDate())-1,’00’)</code></td>
<td>The previous month number as a string</td>
</tr>
</tbody>
</table>

The formula you typed takes the last two functions in the table, concatenates them (using the ‘&’ operator) and concatenates “01” to the end (for the first day of the month) to give a string in the form “YYYYMMDD” (for example “20020601”).
Next, create a variable that expresses this string as a date:

1. Click **Variables** on the **Data** menu.
   The Variables dialog box appears.
2. Click **Add**.
   The Variable Editor appears.
3. Type “FirstDayOfPrevMonthAsString” in the Name box.
4. Click the **Formula** tab.
5. Type the formula
   \[ \text{ToDate}(<\text{FirstDayOfPrevMonthAsString}>, \text{’yyyymmdd’}) \]
in the Formula box.
6. Click **OK** to close the Variable Editor.
7. Click **OK** to close the Variables dialog box.

Note how you greatly simplified this formula by creating the `FirstDayOfPrevMonthAsString` variable beforehand. Without this variable you would have to type the formula it contains in full, making the formula of the `FirstDayOfPrevMonthAsDate` variable much more complex and difficult to follow.

Finally, create a variable that returns the last day of the previous month as a date. You use the `FirstDayOfPrevMonthAsDate` variable that you have already created in this formula.

To do this:

1. Click **Variables** on the **Data** menu.
   The Variables dialog box appears.
2. Click **Add**.
   The Variable Editor appears.
3. Type “LastDayOfPrevMonthAsDate” in the Name box.
4. Click the **Formula** tab.
5. Type the formula
   \[ \text{LastDayOfMonth}(<\text{FirstDayOfPrevMonthAsDate}) \]
in the Formula box.
6. Click **OK** to close the Variable Editor.
7. Click **OK** to close the Variables dialog box.

You can now use the `FirstDayOfPrevMonthAsDate` and `LastDateOfPrevMonthAsDate` variables in your report.
Calculating total revenue for all resorts when some are filtered out

In this example you have a report containing the top two resorts in a report showing resorts, their countries and their associated revenues. You restrict the report to the top two resorts by placing a rank on the Resort column.

<table>
<thead>
<tr>
<th>Top 2 Resorts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td>Sum for All Resorts:</td>
</tr>
</tbody>
</table>

If you insert a Sum calculation on the Revenue column, BusinessObjects uses the following formula by default:

$$ \text{Sum}(<\text{Revenue}>) $$

which gives the following result:

<table>
<thead>
<tr>
<th>Top 2 Resorts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td>Sum:</td>
</tr>
</tbody>
</table>

Why is this different from the result above? By default the Sum function takes into account only the revenues in the block; the sum shown is the total revenue for the Hawaiian Club and Bahamas Beach resorts. The French Riviera resort is filtered from this report by the rank on the Resort column. However, you need to include its revenue in the calculation of total overall revenue. The NoFilter() function makes this possible. This function tells BusinessObjects to ignore filters when calculating, so the formula

$$ \text{NoFilter(Sum(<Revenue>))} $$

returns the total revenue for all resorts.
Accessing Data and Data Analysis

Formulas, Local Variables and Functions
Launching BusinessObjects with the Run Command
Overview

This appendix explains how to run BusinessObjects by using the Run command on Windows. You can use the Run command as an alternative way of double-clicking the BusinessObjects icon. You can also specify command line options such as your user name, password and other options.
Using the Run command

The following procedure describes how to launch BusinessObjects by using the Run command. The options you can include in the Run command are listed and explained in “Run command options” below.

1. Click the **Start** button, then click **Run** on the **Start** menu.
   The Run dialog box appears.

2. In the Open text box, enter the path to the BusinessObjects executable file (Busobj.exe). By default, this file is located in the BusinessObjects folder. You can click Browse to specify the path, rather than type it.

3. Click **OK**.
   The User Identification dialog box appears.

4. Enter the user name and password that your BusinessObjects supervisor provided, then click **OK**.
   The BusinessObjects application is now launched.
### Run command options

Run command options allow you to log in to BusinessObjects with your user name and password. The table below describes the options that you can use.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-user [user name]</td>
<td>The user name assigned to you by your supervisor. User names that include spaces must be written in double quotes, for example “user name”.</td>
</tr>
<tr>
<td>-pass [password]</td>
<td>The password assigned to you by your supervisor. This option is mandatory if you enter the -user option. Passwords that include spaces must be written in double quotes, for example “my password”.</td>
</tr>
<tr>
<td>-online or -offline</td>
<td>By default, the last connection mode of the specified user, or “online” the first time you launch BusinessObjects. Offline mode disconnects you from the repository and therefore disables remote connections during your work session.</td>
</tr>
<tr>
<td>repname.rep</td>
<td>The name of the document that you wish to work with on launching BusinessObjects. You must include the path to this file, for example: c:\BusinessObjects\userdocs\sales.rep</td>
</tr>
<tr>
<td>-keyfile [keyfile name]</td>
<td>If you are working with multiple repositories, specifies the repository you want to work with.</td>
</tr>
<tr>
<td>-nologo</td>
<td>Runs BusinessObjects without showing the logo screen.</td>
</tr>
<tr>
<td>-vars myfile.txt</td>
<td>Name of a text file in which variables are specified. You can specify BOUSER and BOPASS, which manage your access to BusinessObjects. You can also declare your own variables in the file. For more information on these variables, refer to “Specifying BOUSER, BOPASS and Other Variables” below.</td>
</tr>
</tbody>
</table>
**TIP**
In the file you declare after the -vars option, you can also specify the variables such as DBUSER, DBPASSWORD and DBDSN. (The names of such variables depend on the database at your site). These variables can be used to define a restriction on an object, for example. For further information on these variables, refer to “BusinessObjects Variables” in the Database Guide included in your BusinessObjects package.

### Specifying BOUSER, BOPASS and Other Variables

You can use the BOUSER and BOPASS variables to manage your access to BusinessObjects. You can specify the values of these variables in the Run command, or in a file that you call from the Run command. Other variables can be declared in this file.

#### BOUSER and BOPASS

When the BusinessObjects supervisor creates users, they assign each one a user name and password. The user’s name and password are stored on the repository. When you log in to BusinessObjects in online mode, which is the default working mode, BusinessObjects connects to the repository and reads your security information. Your user name and password are then written to either the objectslsi file or the objectsssi file, located in either the ShData folder or the LocData folder.

Once you have launched BusinessObjects in online mode, you can use the BOUSER and BOPASS variables in the Run command. You can:

- Declare the value of the variables after -user and -pass.
  
  For example, if your supervisor assigned you the user name JOHN and the password SMITH, you can write the following command:
  
  c:\BusinessObjects\Busobj.exe -user JOHN -pass SMITH

- Declare the variables and their values in a text file in the BusinessObjects folder. Then, in the Run command, you specify the file name after the -vars option.
  
  For example, if your supervisor assigned you the user name JOHN and the password SMITH, you create a .txt file (myfile.txt) in which you specify:
  
  BOUSER=JOHN
  BOPASS=SMITH

  You can now use the following Run command:
  
  c:\BusinessObjects\Busobj.exe -vars myfile.txt
NOTE

User names and passwords that contain spaces must be written in double quotes, e.g., "user name". You must use upper-case characters when specifying the variables that manage security, as in the example above.

► Other variables you can specify in a file

In the .txt file that you declare after the -vars option, you can specify other variables that you work with in BusinessObjects. For example, if you have created a variable that displays a prompt when a query is run, you can specify this variable’s value in the .txt file. The syntax is as follows:

VARIABLENAME=VALUE
BusinessObjects and Visual Basic for Applications
Overview

You can customize BusinessObjects 6.x using the Microsoft Visual Basic for Applications programming language. BusinessObjects 6.x has a Visual Basic Editor that you can use to develop macros, add-ins and VBA data providers. The Visual Basic Editor is the standard Microsoft VBA editor that you may already be familiar with if you use Microsoft Office products.

This chapter describes how to use macros and add-ins in BusinessObjects. For information on building VBA data providers, see Using Visual Basic for Applications procedures on page 98.

What is a macro?

A macro is a series of commands and functions that are stored in a Visual Basic for Applications module and can be run whenever you need to perform the task. If you perform a task repeatedly, you can automate the task with a macro. You create macros using the Visual Basic Editor.

What is an add-in?

Add-ins are programs that add optional commands and features to BusinessObjects. Add-ins are usually created by those responsible in your company for adding customized features to BusinessObjects. All you probably need to do is install and uninstall add-ins that are sent to you.

Before you can use an add-in, you must install it on your computer and then load it in BusinessObjects. Add-ins (*.rea files) are installed by default in the UserDocs folder in the BusinessObjects folder. Loading an add-in makes the feature available in BusinessObjects and adds any associated commands to the appropriate menus.

Unloading an add-in removes its features and commands from BusinessObjects, but the add-in program remains on your computer so you can easily load it again.

You can use your own Visual Basic for Applications programs as custom add-ins. For information about making a Visual Basic for Applications program an add-in, see the BusinessObjects SDK Reference Guide.

What about the scripts I used in BusinessObjects 4.1?

The scripts that you created using the ReportScript programming language can be automatically converted into Visual Basic macros by BusinessObjects 6.1. See Converting scripts to macros on page 516 for information on how to do this.
Using macros

Macros are created and stored inside BusinessObjects documents (.rep files) or BusinessObjects add-ins (.rea files). You can run macros either from the Macros dialog box or from the Visual Basic toolbar if macros have been assigned to the macro buttons.

Running a macro

1. Click Macro, then Macros on the Tools menu or click Macros on the Visual Basic toolbar.
   The Macros dialog box opens.

2. From the Macros in: list box, choose the documents where the macros are stored. You can display the macros available in the active document, all macros in all open documents, macros in a selected open document or macros in add-ins.

   The macros stored in the selected document(s) are displayed in the Macro Name list.

3. Select the name of the macro you want to use and click Run.
NOTE

If VBA is not installed and you open a document that contains macros, you receive an error message warning you that macros will not be executed. In this situation, your document might return incorrect or incomplete data. The Olap Connect and Web Connect data providers also require VBA in order to function correctly.

Using the Visual Basic toolbar

To open the Visual Basic toolbar:
• Right-click on any other open toolbar and click Visual Basic on the shortcut menu.

<table>
<thead>
<tr>
<th>Visual Basic</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>E</td>
<td>C</td>
</tr>
</tbody>
</table>

Opens the macros dialog box
Opens the Visual Basic Editor
Buttons 1-5 run the macros that have been assigned to them
**Associating a macro to a toolbar button**

1. Click **Options** on the **Tools** menu.
   The Options dialog box opens.
2. Click the **Macro** tab.

3. Click check box 1 to activate the first button on the Visual Basic toolbar.
4. Click the button to the right of the Macro Name box.
5. The Macros dialog box opens.
6. Click on the macro you want to use from the list and click **Select**.
   The name the macro is displayed in the Macro Name box.
7. In the Tooltip box, type the tooltip that you want to use for the macro.
   The tooltip appears when you rest the cursor over the button on the Visual Basic toolbar.
Using add-ins

BusinessObjects add-ins are Visual Basic for Applications programs that add optional commands and features to BusinessObjects. You can distribute add-ins you have created to other users and retrieve and use add-ins that others have created.

Installing an add-in

Click **Add-Ins** on the **Tools** menu.

The Add-Ins dialog box opens.

8. Click **Browse** to locate and open the add-ins on your computer.

The Add-Ins Available box displays the list of available add-ins. There are two types of add-ins that you may see in this dialog box: those that are available and those that have been installed. You cannot use an add-in until it has been installed.

9. Click the check box next to the name of the add-in and click **OK**.

The add-in is installed and can now be used.

**NOTE**

When a user installs an add-in, it is only installed for that user. If that user logs on under a different name, the add-in will not be available.
Using an add-in

You can run an installed add-in from the Macros dialog box, or you can associate it with a button on the Visual Basic toolbar.

Uninstalling an add-in

1. Click the check box next to the add-in name in the Add-Ins dialog box to remove the check mark
2. Click OK.
   The add-in features and commands are removed from BusinessObjects, but the add-in program remains on your computer so you can easily load it again if you want to use it.

Exchanging add-ins with other users

You can send and retrieve add-ins (.rea files) in the same way you can send and retrieve BusinessObjects documents.
Converting scripts to macros

In BusinessObjects 4.1, you could create scripts to automate tasks using the ReportScript programming language. This programming language has been replaced in BusinessObjects 5.x and upwards by the Visual Basic programming language.

BusinessObjects 6.1 can convert your SBL scripts into Visual Basic (VBA) macros which you can then run from the Macros dialog box.

The script is converted in the following way:
• Dialogs are converted to a VBA form.
• The code logic is converted to a VBA module.
• SBL specific functions and instructions are declared in an extra module.

To convert a script
1. Click Macro then Convert from ReportScript on the Tools menu.
   The Open dialog box is displayed. By default, the Scripts folder is open and a list of available scripts is displayed.
   Image not available for Beta
2. Select the script you want to convert.
3. In the Convert in: list box, choose where you want the converted macro to be saved.
   You can convert the macro in the active document or in a new document.
4. Click Import.
   The script is converted.

NOTE
The macro may sometimes need some slight tweaking in the Visual Basic Editor after conversion to get it to work correctly. For information on how to do this, see the Customizing BusinessObjects guide.
Using the Visual Basic editor

You can open the Visual Basic Editor directly from BusinessObjects to create macros and programs to use in BusinessObjects. This development environment has its own set of online Help files.

To open the Visual Basic Editor:

- Click **Visual Basic Editor** on the Visual Basic toolbar.
  The Visual Basic development environment opens up.

Programming in Visual Basic requires knowledge of the programming environment. This is covered in the *Customizing BusinessObjects* guide.
Index

### error 450
#ALERTER error 451
#COMPUTATION error 439
#DICT.ERROR error 452
#DIV/0 error 454
#ERROR error 455
#ERR error 456
#MULTIVALUE error 443
#OVERFLOW error 459
#SYNTAX error 460
#UNKNOWN error 461

A
Abs function
Excel equivalent 487
accessing data sources 36
alerters
  copying 327
creating 323
deleting 328
displaying/hiding 327
editing 327
naming 324
selecting data 324
setting conditions 325
setting formatting 325
switching off 326
switching on 326
analyzing data
  on-report analysis 233
  overview 232
  using an OLAP server 235
  using drill mode 234
  using slice-and-dice mode 236
Asc function
Excel equivalent 489
Average function
Excel equivalent 487

B
Basic Authentication
  querying a page using 175
blocks
  using separate data providers for 205
BOPASS variable 507
BOUSER variable 507
breaks
  and ranking 319
  applying in slice-and-dice mode 292
BusinessObjects
  defined 25
demo materials and samples 30
  launching with the Run command 505
new features 31
source of data 25
  upgrading from earlier versions 31
BusinessObjects administrator
  and free-hand SQL 87

C
calculations
  adding to reports 389
  examples of 392
  re-using 395
Ceil function
Excel equivalent 487
cells
  autoqualifying 171
Char function
Excel equivalent 489
charts
  displaying data in 208
drilling on 247
combining data 40
complex condition calculations
  examining the SQL of 352
Accessing Data and Data Analysis

complex conditions
  applying on queries 337
  applying with a calculation 344
  applying with calculations 344
  deleting 344
  editing 343
  tips for applying 341
  using the Different From operator 342
  using the Except operator 342
  using the In List operator 342
  using the Not In List operator 342
  using wildcard characters 341

Concatenation function
  Excel equivalent 489

conditions
  applying to queries 65
  defined 65

connections
  defining 39
  setting up 39, 40

 corporate queries
  running 131

Cos function
  Excel equivalent 487

Count function
  Excel equivalent 487

crosstab
  building from table or 2-D chart 286
  moving data between columns and rows 288
  reducing to table or 2-D chart 286

crosstabs
  displaying data in 208
  custom hierarchies 262

data
  presenting and analyzing 27
  workflows for accessing 41

data providers
  adding to existing document 47
  basing on existing data providers 210
  cancelling 48
  combinations of 204
  creating a base data provider 227
  deleting links between 217
  discarding the data from 49
  editing 47, 48
  free-hand SQL 75
  getting statistics on 224
  linking 207, 212
  linking dimensions 212
  linking existing 214
  linking when inserting new block 213
  OLAP cubes 116
  packaged applications 36
  personal data files 94
  prompted linking 212
  purging and deleting 225
  reasons for renaming 221
  renaming 221, 222
  restrictions on access 39
  restrictions on building 38
  situations requiring linking 212
  spreadsheets 36
  stored procedures 88
  text files 36
  using efficiently 227
  using for separate blocks 205
  VBA 36
  VBA procedures 98
  viewing data from previous execution 49
  viewing partial results 49
  Web Connect 117
  web pages 36
  workflows for creating 41
  XML 36, 105

data sources 203
  accessing 35, 36
  combining 40
  combining different in one report 204
  list of 36
Accessing Data and Data Analysis

database connections
  creating 78, 79, 80
  editing 79, 81
  managing using free-hand SQL editor 79
  selecting 78
demonstration materials 53
eFashion 55
generic SQL scripts 53
Island Resorts Marketing 53
detail objects 55
dimension objects 55
dimensions
  adding to hierarchies 263
  restrictions on combining 263, 265
documents
  creating new 56
  creating new data providers in 56
  editing data providers in 56
  refreshing 29, 168
  refreshing in Broadcast Agent 169
  refreshing in InfoView 169
drill filters
  bringing in new data using 259

drill mode
  analyzing measures 254
  and hierarchies 239
  bringing in new data using filters 259
  changing data in tables 250
  continuing drill-down 242
  default BusinessObjects actions 241
  defined 234, 238
  displaying values in the Drill toolbar 243
  drilling across 244
  drilling down 241
  drilling down to another hierarchy 245
  drilling on charts 247
  drilling on multiple hierarchies 248
  drilling through to the database 258
  drilling up 243
  drilling using custom hierarchies 262
  printing a report from 269
  printing from 269
  re-organizing the drill toolbar 253
  setting options 270
  setting up a report for 240
  switching to 241
  taking report snapshots 256
  undoing drill actions 244
  using 241
  using the drill toolbar 252
Drill toolbar
  moving variables from 251
  moving variables to 250
  using 252
drill toolbar
  displaying values in 243
  inserting contents as a report title 269
  moving objects to block from 253
  removing objects from 252
  re-organizing 253
  drilling down 241, 245
  drilling through 258
  drilling up 243
error messages

#ALERTER 451
#COMPUTATION 439
#DICT ERROR 452
#DIV/0 454
#ERROR 455
#IERR 456
#MULTIVALUE 443
#OVERFLOW 459
#SYNTAX 460
#UNKNOWN 461

tips and tricks regarding 462

EuroConvertFrom function
Excel equivalent 488

EuroConvertTo function
Excel equivalent 488

EuroFromRoundErr function
Excel equivalent 487

Euros
converting from 400
converting to 399
converting to and from 398
displaying rounding errors in calculations 400
displaying the euro symbol 398

EuroToRoundErr function
Excel equivalent 487

Exp function
Excel equivalent 488

extended calculation syntax 409
and Rank function 425
context operators 435
ForAll operator 422
ForEach operator 422
keywords 429
quick reference 433

filters
adding 302
applying in slice-and-dice mode 292
creating complex filters 304
displaying names of in special fields 303
editing complex filters 305
ignoring 305
inserting 299
inserting complex filters 304
managing 300
removing 303
selecting different values for 301

Floor function
Excel equivalent 488

FormatDate
Excel equivalent 489

FormatNumber
Excel equivalent 489

formulas
creating using Formula Editor 469
defined 467
guidelines on syntax 471
qualifying 266

free-hand SQL
and sorts 69
creating a connection 80
creating interactive reports with 83
creating reports using 75
creating/editing a connection for 79
editing a connection for 81
editing an SQL script 79
example of 83
example of query with prompt 86
opening an existing script 77
restrictions on scripts 87
viewing raw data 78
writing a new script 77

functions
Excel equivalents of 487

grouped values
adding to drill hierarchy 480
Index

H
hidden data
  displaying 322
  hiding report data 321
  hiding data 321
hierarchies 239
  adding dimensions to 263
  changing the order of dimensions in 264
  creating 264
  creating custom hierarchies 264
  custom hierarchies defined 262
  default 239
  deleting 264
  drilling on multiple hierarchies 248
  drilling up on multiple hierarchies 249
  drilling using custom hierarchies 262
  editing 262
  in Island Resorts Marketing 240
  qualifying data for 266
  removing dimensions from 264
  renaming 264
  restrictions on editing 262
  viewing 63
highlighting data see Alerters 323

I
InitCap function
  Excel equivalent 489
Internet queries 121
  conversion errors 174
  creating 138
  editing 166
  saving 167
  saving preferences 168
  setting individual options 172
Internet Query Panel
  description of 134
  setting size 168
  toolbar buttons 135

L
Left function
  Excel equivalent 489
LeftTrim function
  Excel equivalent 489
Length function
  Excel equivalent 489
lists of values
  assigning data from a dBase file 383
  assigning personal data from a text file 380
  assigning personal data from an Excel file 382
  assigning personal data to 380
  customizing 375
  defined 373
  displaying, purging and refreshing 385
  editing 376
Ln function
  Excel equivalent 488
local variables
  creating 473
  creating by grouping values 476
  defined 472
  qualifying 266
  transforming formulas into 474
Log function
  Excel equivalent 488
Log10 function
  Excel equivalent 488
Lower function
  Excel equivalent 489

M
master/detail reports
  activating/deactivating multiple sections 282
  building 277
  deactivating sections of 279, 280
  ranking in 318
  reactivating sections of 282
  reorganizing 276
  structuring existing reports as 275
  undoing 278
  using different master 277
  working in slice-and-dice mode with 273
Match function
  Excel equivalent 489
matrix chart, 3-D
  building from table or 2-D chart 286
Max function
Excel equivalent 487

measure objects 55
measures
analyzing in drill mode 254
 collapsing 255
defined 254
expanding 254
Median function
Excel equivalent 488
Min function
Excel equivalent 487
Mod function
Excel equivalent 488
monovalued prompts
defined 153
Month function
Excel equivalent 489
MonthNumberOfYear function
Excel equivalent 489
multivalued prompts
defined 153

P
personal data files 37
benefits of 94
creating reports using 94
setting options 97
Pos function
Excel equivalent 489
Power function
Excel equivalent 488
predefined conditions
applying to queries 66
in Island Resorts Marketing universe 66
removing 67
printing
from drill mode 269
prompts
adding to Web Connect queries 124
creating in Web Connect queries 154
defined 153
displaying values in a separate column 157
in free-hand SQL 83
monovalued 153
multivalued 153
syntax in free-hand SQL scripts 84
using with Web Connect queries 153

Q
Quarter function
Excel equivalent 489

O
objects
changing order of in queries 60
detail 55
displaying in Query Panel 59
including in queries 60
measure 55
moving from Drill toolbar to blocks 253
removing from queries 60
searching for in Query Panel 60
OLAP cubes
using to build reports 116
OLAP servers 38, 235
operators
Different From 342
Except 342
In List 342
Not In List 342
options
setting for drill mode 270
queries
- adding content to in Web Connect 136
- applying complex conditions on 337, 338
- building in Query Panel 59
- building on universes 52, 56
- building powerful queries 63
- Business Objects Web Connect queries 123
- corporate Web Connect queries 123
- creating new Web Connect queries 133
- eliminating duplicate rows 70
- generating SQL without retrieving data 70
- Internet queries 123
- running 61
- running on different universes 72
- saving definitions of 62
- setting options 70
- specifying number of rows to return 70
- using output in conditions 356
- query catalogs 121
- query panel
  - building a query using 59
  - changing object order 60
  - described 58
  - displaying 56
  - including objects in queries 60
  - removing objects from queries 60
  - running queries from 61
  - searching for objects in 60

reports
- adding columns to 212
- combining data sources in 204
- including data from different data sources 204
- making copies of 256
- sharing 28
- Right function
  - Excel equivalent 488
- RightTrim function
  - Excel equivalent 488
- Round function
  - Excel equivalent 488
- SAP 38
  - saving documents in Excel format 31
- scope of analysis
  - and hierarchies 63
  - defined 63
  - defining 63
  - defining default 64
  - defining manually 65
  - drilling through to the database 258
  - extending 257
- secured web sites
  - accessing using Web Connect 175
- security 29
- separators
  - specifying in Internet queries 173
- Sign function
  - Excel equivalent 488
- simple conditions
  - applying 67
  - deleting 69
  - selecting different values for 68
- Sin function
  - Excel equivalent 488

R
- ranking
  - and breaks 319
  - applying in slice-and-dice mode 291
- ranking data 315
  - displaying percentages 318
  - displaying subtotals 317
  - editing an existing ranking 319
  - ranking in master/detail reports 318
  - removing a ranking 320
- RelativeDate function
  - Excel equivalent 489
- Replace function
  - Excel equivalent 489
- Report Manager
  - moving variables from 250
slice-and-dice mode
  adding data in 209
  and Web Connect queries 124
  applying breaks in 292
defined 272
displaying and removing data 288
making calculations in 291
moving data in crosstabs 288
positioning data horizontally in 282
ranking data in 291
repositioning data vertically 287
sorting data in 291
transforming blocks in 289
using filters in 292
working in 273
working with blocks in 289
working with crosstabs/3-D matrix charts in 284
working with master-detail reports in 273
sorting data 307
sorts
  adding from the Sorts dialog box 314
  and free-hand SQL 69
  applying custom 309
  applying in slice-and-dice mode 291
  applying on objects 69
  applying on report data 307
defined 69
defining priority 70
defining sort priority 313
inverting 69, 307
managing 311
removing 69, 308, 314
SQL
  defined 75
  parsing 77
Sqrt function
  Excel equivalent 488
StDev function
  Excel equivalent 487
StDevP function
  Excel equivalent 487
stored procedures
  defined 37, 88
  example of 89, 90
  restrictions on 88
  supplying parameters to 93
  using to retrieve data 88
Substr function
  Excel equivalent 488
Sum function
  Excel equivalent 487
T
  Tan function
    Excel equivalent 488
ToDate function
  Excel equivalent 489
ToNumber function
  Excel equivalent 488
transparent sorts
  defined 70
Trim function
  Excel equivalent 488
Truncate function
  Excel equivalent 488
U
  universe queries
    defined 52
universes
  building basic queries on 56
  creating 52
  creating queries with 52
defined 37, 52
  running queries on different 72
  selecting 43
Upper function
  Excel equivalent 488
user objects
creating 331
creating a time hierarchy for 336
defined 331
deleting 336
ing 335
qualifying 268
reasons for creating 331
restrictions on 332
sharing with other users 332

V
Var function
Excel equivalent 487
variables
compatibility rules 207
VarP function
Excel equivalent 487
VBA procedures 37
accessing Outlook with 100
creating a report using 99
example of 102
using 98
<table>
<thead>
<tr>
<th>W</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Connect</td>
<td>accessing form pages 179</td>
</tr>
<tr>
<td></td>
<td>accessing secured Web sites 175</td>
</tr>
<tr>
<td></td>
<td>adding content to a query 136</td>
</tr>
<tr>
<td></td>
<td>adding multiple cells in one cell 136</td>
</tr>
<tr>
<td></td>
<td>adding prompts to queries 124</td>
</tr>
<tr>
<td></td>
<td>assembling query data 122</td>
</tr>
<tr>
<td></td>
<td>autoqualifying cells 171</td>
</tr>
<tr>
<td></td>
<td>clearing data from rows 151</td>
</tr>
<tr>
<td></td>
<td>creating a new query 133</td>
</tr>
<tr>
<td></td>
<td>creating a query with prompts 192</td>
</tr>
<tr>
<td></td>
<td>creating Internet queries 123</td>
</tr>
<tr>
<td></td>
<td>creating prompts 154</td>
</tr>
<tr>
<td></td>
<td>creating reports in HTML 144</td>
</tr>
<tr>
<td></td>
<td>deleting columns 150</td>
</tr>
<tr>
<td></td>
<td>deleting rows 151</td>
</tr>
<tr>
<td></td>
<td>editing an Internet query 166</td>
</tr>
<tr>
<td></td>
<td>editing cells 151</td>
</tr>
<tr>
<td></td>
<td>generating reports 124</td>
</tr>
<tr>
<td></td>
<td>interpretation of separators 172</td>
</tr>
<tr>
<td></td>
<td>linking data to data in browser 151</td>
</tr>
<tr>
<td></td>
<td>making data static 151</td>
</tr>
<tr>
<td></td>
<td>managing queries 164</td>
</tr>
<tr>
<td></td>
<td>menu options 150</td>
</tr>
<tr>
<td></td>
<td>opening existing queries 164</td>
</tr>
<tr>
<td></td>
<td>pasting data into rows 151</td>
</tr>
<tr>
<td></td>
<td>placing data on the clipboard 151</td>
</tr>
<tr>
<td></td>
<td>qualifying data as details 150</td>
</tr>
<tr>
<td></td>
<td>qualifying data as dimensions 150</td>
</tr>
<tr>
<td></td>
<td>qualifying data as measures 150</td>
</tr>
<tr>
<td></td>
<td>query examples 183</td>
</tr>
<tr>
<td></td>
<td>querying multi-frame pages 169</td>
</tr>
<tr>
<td></td>
<td>querying pages using Basic Authentication 175</td>
</tr>
<tr>
<td></td>
<td>refreshing reports 124</td>
</tr>
<tr>
<td></td>
<td>renaming headers 150</td>
</tr>
<tr>
<td></td>
<td>retrieving Business Objects queries 128</td>
</tr>
<tr>
<td></td>
<td>retrieving corporate queries 128</td>
</tr>
<tr>
<td></td>
<td>retrieving HTML cell source 151</td>
</tr>
<tr>
<td></td>
<td>retrieving hyperlinks 151</td>
</tr>
<tr>
<td></td>
<td>running a query with a prompt 162</td>
</tr>
<tr>
<td></td>
<td>running Business Objects queries 129</td>
</tr>
<tr>
<td></td>
<td>running corporate queries 131</td>
</tr>
<tr>
<td></td>
<td>saving Internet queries 167</td>
</tr>
<tr>
<td></td>
<td>saving Internet query preferences 168</td>
</tr>
<tr>
<td></td>
<td>saving reports 149</td>
</tr>
<tr>
<td></td>
<td>selecting corporate queries 123</td>
</tr>
<tr>
<td></td>
<td>selecting the Web Connect data provider 126</td>
</tr>
<tr>
<td></td>
<td>setting data types as characters 150</td>
</tr>
<tr>
<td></td>
<td>setting data types as dates 150</td>
</tr>
<tr>
<td></td>
<td>setting data types as numeric 150</td>
</tr>
<tr>
<td></td>
<td>setting error management options 173</td>
</tr>
<tr>
<td></td>
<td>setting query timeout 168</td>
</tr>
<tr>
<td></td>
<td>using prompts with 153</td>
</tr>
<tr>
<td></td>
<td>Web Connect data provider 31</td>
</tr>
<tr>
<td></td>
<td>selecting 126</td>
</tr>
<tr>
<td></td>
<td>Week function</td>
</tr>
<tr>
<td></td>
<td>Excel equivalent 489</td>
</tr>
<tr>
<td></td>
<td>WordCap function</td>
</tr>
<tr>
<td></td>
<td>Excel equivalent 488</td>
</tr>
<tr>
<td></td>
<td>World Wide Web 38</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XML</td>
</tr>
<tr>
<td></td>
<td>building an XML filter 106</td>
</tr>
<tr>
<td></td>
<td>creating a report using 106</td>
</tr>
<tr>
<td></td>
<td>defined 105</td>
</tr>
<tr>
<td></td>
<td>example of 105</td>
</tr>
<tr>
<td></td>
<td>setting file locations 115</td>
</tr>
<tr>
<td></td>
<td>XML data provider 31</td>
</tr>
<tr>
<td></td>
<td>XML files 37</td>
</tr>
<tr>
<td></td>
<td>XML query panel 113</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Year function</td>
</tr>
<tr>
<td></td>
<td>Excel equivalent 489</td>
</tr>
</tbody>
</table>