SAP BusinessObjects Desktop Intelligence Access and Analysis Guide
## Contents

<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>Introduction to Desktop Intelligence</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is Desktop Intelligence?</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Where does the data come from?</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Presenting and analyzing data</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Sharing information</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Keeping a document's data up-to-date</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Demo materials and samples</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Upgrading from earlier versions of Desktop Intelligence</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>The Repository</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Folders and Categories</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Folders</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Categories</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>There are 2 types of folders:</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>There are two types of Categories:</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Introduction to Accessing Data with Desktop Intelligence</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>What data sources are available?</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>How do you access data sources?</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Can all Desktop Intelligence users build data providers?</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Who sets up database connections?</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>Restrictive connections</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Can you combine data from different sources in one report?</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Workflows for accessing data</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Building a data provider when you create a new document</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Building a query in an existing document</td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3  Building Queries on Universes  45

Overview....................................................................................................46
   What is a universe?..............................................................................46
   Who is responsible for creating universes?.........................................46
   What are universe queries?.................................................................47
   Demonstration materials......................................................................47

Building a basic query on a universe.........................................................49
   Displaying the query panel.................................................................49
   Building a query in the Query Panel and running the query.................51
   Saving the definition of a query..........................................................54
Chapter 4  Building Queries with Other Types of Data Provider 65

Overview...........................................................................................................66
Using free-hand SQL.........................................................................................66
  Creating a report using free-hand SQL..........................................................66
  Editing a free-hand SQL script.................................................................68
Creating or editing a connection for free-hand SQL.........................................68
Creating a report showing sales by store and category.....................................70
Creating interactive reports using free-hand SQL..........................................71
Restrictions on free-hand SQL scripts..............................................................75
Using stored procedures....................................................................................76
  What are stored procedures?.........................................................................76
How do you use stored procedures in Desktop Intelligence?.........................76
Restrictions on stored procedures......................................................................76
Using a stored procedure to retrieve data.........................................................76
Using personal data files..................................................................................78
  What are the benefits of using personal data files?......................................78
Creating a report using a personal data file....................................................79
Using Visual Basic for Applications procedures............................................81
  To write a VBA data provider......................................................................81
To create a report using a VBA data provider.................................................82
Accessing an Outlook inbox using VBA..........................................................82
Using XML files.................................................................................................85
  What is XML?..............................................................................................86
## Chapter 7: Introduction to Data Analysis

- Overview ................................................................. 110
- On-report analysis .................................................. 110
- Desktop Intelligence drill mode .............................. 111
- Slice and dice mode .................................................. 111

## Chapter 8: Analyzing Data in Drill Mode

- Overview ................................................................. 114
  - What is drill mode? ................................................ 114
  - How does drill mode work? ................................. 114
  - Hierarchies .......................................................... 115
- Using drill mode ...................................................... 116
  - To switch to drill mode ........................................ 116
  - Drilling down ....................................................... 117
  - Displaying different values in the Drill toolbar .... 118
  - Drilling up ............................................................ 118
  - Undoing drill actions ........................................... 118
  - Drilling across ..................................................... 119
- Drilling on charts .................................................... 120
  - To drill on charts ................................................ 120
- Drilling on multiple hierarchies ............................ 120
  - To drill on multiple hierarchies .......................... 121
  - To drill up on multiple hierarchies ...................... 121
- Getting a different view of your data .................... 121
  - Changing the data in tables as you drill .............. 122
  - Using the Drill toolbar ........................................ 124
- Analyzing measures in drill mode ....................... 126
  - To expand a measure ........................................... 127
  - To collapse a measure ........................................ 127
Chapter 9  Slice and Dice Mode  137

Overview.................................................................................................. 138
Working in slice-and-dice mode................................................................. 138
  To display the Slice and Dice Panel:.................................................. 138
Working with master/detail reports in slice and dice mode.................. 139
  To undo a master/detail report.......................................................... 141
Deactivating sections of master/detail reports..................................... 142
Positioning data horizontally in slice-and-dice mode......................... 144
Working with crosstabs and 3-D matrix charts................................. 144
Displaying and removing data in Slice-and-Dice Mode....................... 148
Deleting, renaming and resetting blocks in Slice and Dice mode........ 148
To transform blocks in Slice-and-Dice mode...................................... 149
Chapter 10  Filtering and Ranking Data

Overview..................................................................................................154
Limiting the data displayed......................................................................154
   To insert a filter...................................................................................154
   Managing filters..................................................................................155
   Creating more complex filters.............................................................158
   Ignoring filters.....................................................................................159
Ordering data...........................................................................................160
   Sorting data........................................................................................160
   To sort months correctly.....................................................................162
   Managing multiple sorts.....................................................................163
Using ranking to view the top and bottom values....................................164
   To apply a ranking on report data.......................................................165
   Managing ranking with filters and sorts..............................................168
Hiding columns and rows of data............................................................168
   To hide columns and rows of data......................................................168
Highlighting data......................................................................................169
   To create an alert.............................................................................169
   Switching alerters off and on............................................................171
   Working with existing alerters..........................................................172

Chapter 11  Customizing Queries on Universes

Overview.................................................................................................176
Creating user objects..............................................................................176
   Why create a user object?.................................................................176
   What does a user object consist of?................................................177
   What are the restrictions on user objects?........................................177
   How can an end-user share user objects with other users?..............178
Creating, editing and deleting user objects ................................................................. 178
To create a time hierarchy for a user object .................................................................. 181
Applying complex conditions on queries ....................................................................... 181
To benefit from complex conditions: Which customers made reservations for 2001 and 2002? .......................... 182
To apply a complex condition on a query ........................................................................ 183
Tips for applying complex conditions ............................................................................. 187
To edit complex conditions ............................................................................................ 190
To delete complex conditions .......................................................................................... 190
Applying a condition with a calculation .......................................................................... 190
Examining the SQL .......................................................................................................... 194
Applying a condition with a subquery ............................................................................ 194
Subqueries and calculations ............................................................................................. 197
Using an existing query in a condition .............................................................................. 198
   To return list of resorts/revenues where resort country revenue > $1000000 ........................................ 198
Applying groups of conditions ......................................................................................... 199
   Organizing groups of conditions .................................................................................. 199
   AND and OR ................................................................................................................. 200
   Order of precedence ...................................................................................................... 201
   To apply groups of conditions ....................................................................................... 202
   To delete groups of conditions ..................................................................................... 203
Building combined queries .............................................................................................. 203
   To build a combined query ............................................................................................ 203
   Restrictions on combined queries ................................................................................ 205
   Using SQL from Desktop Intelligence queries .......................................................... 207

Chapter 12 Using and Customizing Lists of Values .......................................................... 209
Overview .......................................................................................................................... 210
What is a list of values? ..................................................................................................... 210
How are lists of values created? ....................................................................................... 210
Customizing lists of values in Desktop Intelligence.................................210
Editing lists of values..................................................................................211
Example: To show cities and regions in a list of cities...............................211
Assigning personal data to a list of values..................................................212
To assign personal data from a text file......................................................212
To assign personal data from an Excel file...............................................213
To assign personal data from a dBase file....................................................214
To display, refresh and purge lists of values.................................................214

Chapter 13 Creating Calculations 217
Overview......................................................................................................218
Calculations.................................................................................................218
To add simple calculations to reports.........................................................218
Count and Count All....................................................................................220
Making calculations on dimension and detail objects.....................................220
Calculation examples....................................................................................221
Converting to and from Euros ........................................................................223
What is the euro?..........................................................................................223
Displaying the euro symbol..........................................................................224
How does the conversion work?....................................................................224
Conversion errors........................................................................................224
Displaying currency formats in Desktop Intelligence....................................225
To convert to euros........................................................................................225
To convert from euros....................................................................................226
Displaying rounding errors..........................................................................226
Conversion rates............................................................................................227
Triangulation.................................................................................................230

Chapter 14 Overview 231
Who should read this chapter........................................................................232
# Contents

What's in this chapter........................................................................................................232

## Chapter 15  Introduction to contexts and extended syntax  233

- Semantically-dynamic calculations.................................................................................234
- Understanding input and output contexts.................................................................234
  - How Desktop Intelligence defines input and output contexts..................................235
- Using your understanding of input and output contexts...........................................237
  - Viewing the extended syntax of a formula...............................................................239
  - Viewing extended syntax by using the Formula Bar.................................................239
  - Viewing extended syntax by using the Define As Variable command.......................239

## Chapter 16  Using extended syntax for advanced calculations  241

- Defining calculation contexts with extended syntax....................................................243
- How to define input and output contexts.......................................................................244
  - Syntax for input and output contexts......................................................................244
  - To add an input and output context to a formula......................................................245
- Reset contexts................................................................................................................245
  - How to define reset contexts..................................................................................246
  - To define a reset context: ......................................................................................246
  - Using reset contexts in crosstabs............................................................................247
  - Syntax for combining reset, input and output contexts..........................................248
- Modifying contexts with the operators ForEach and ForAll.........................................248
  - Getting the same result: ForAll City vs. In Region..................................................250
  - Using the Rank function and extended syntax.........................................................251
  - Defining contexts with keywords............................................................................254
- Quick reference...............................................................................................................258
  - Frequently used terms.............................................................................................258
  - Calculation contexts..............................................................................................261
  - Context operators..................................................................................................263
  - Keywords..............................................................................................................263
## Chapter 17 Calculation Troubleshooting

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>266</td>
</tr>
<tr>
<td>#COMPUTATION</td>
<td>266</td>
</tr>
<tr>
<td>#COMPUTATION in cumulative aggregations</td>
<td>266</td>
</tr>
<tr>
<td>#COMPUTATION in non-aggregate formulas</td>
<td>270</td>
</tr>
<tr>
<td>#MULTIVALUE</td>
<td>271</td>
</tr>
<tr>
<td>#MULTIVALUE in aggregations</td>
<td>272</td>
</tr>
<tr>
<td>#MULTIVALUE in break headers and footers</td>
<td>274</td>
</tr>
<tr>
<td>#ALERTER</td>
<td>280</td>
</tr>
<tr>
<td>#DICT.ERROR</td>
<td>281</td>
</tr>
<tr>
<td>To fix this problem:</td>
<td>281</td>
</tr>
<tr>
<td>To avoid #DICT.ERROR</td>
<td>281</td>
</tr>
<tr>
<td>#DIV/0</td>
<td>282</td>
</tr>
<tr>
<td>To fix this problem:</td>
<td>282</td>
</tr>
<tr>
<td>#ERROR</td>
<td>283</td>
</tr>
<tr>
<td>To fix this problem:</td>
<td>283</td>
</tr>
<tr>
<td>#IERR</td>
<td>283</td>
</tr>
<tr>
<td>#IERR in a formula combining measures and dimensions</td>
<td>284</td>
</tr>
<tr>
<td>#IERR in an aggregation containing a complex formula</td>
<td>284</td>
</tr>
<tr>
<td>#IERR in a formula using WHERE</td>
<td>285</td>
</tr>
<tr>
<td>#OVERFLOW</td>
<td>286</td>
</tr>
<tr>
<td>#SYNTAX</td>
<td>286</td>
</tr>
<tr>
<td>#UNKNOWN</td>
<td>287</td>
</tr>
<tr>
<td>To fix this problem:</td>
<td>288</td>
</tr>
<tr>
<td>Tips and tricks</td>
<td>288</td>
</tr>
</tbody>
</table>

## Chapter 18 Formulas, Local Variables and Functions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>293</td>
</tr>
</tbody>
</table>

SAP BusinessObjects Desktop Intelligence Access and Analysis Guide
To create the variable to calculate a three-week rolling average for sales revenue..................................................................................................................315

Chapter 19  Launching Desktop Intelligence with the Run Command 327
Overview..................................................................................................................328
To use the Run command..................................................................................328
Run command options.......................................................................................328
Specifying BOUSER, BOPASS and Other Variables.........................................330

Chapter 20  Desktop Intelligence and Visual Basic for Applications 333
Overview..................................................................................................................334
What is a macro?..................................................................................................334
What is an add-in?...............................................................................................334
Using macros.......................................................................................................335
To run a macro.....................................................................................................335
Using add-ins......................................................................................................336
To install an add-in.............................................................................................337
Using an add-in..................................................................................................337
To uninstall an add-in........................................................................................337
Exchanging add-ins with other users.................................................................338
Converting scripts to macros.............................................................................338
To convert a script..............................................................................................338
Using the Visual Basic editor...........................................................................339
To open the Visual Basic Editor......................................................................339
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 21</td>
<td>Overview</td>
<td>341</td>
</tr>
<tr>
<td>Chapter 22</td>
<td>Syntax</td>
<td>343</td>
</tr>
<tr>
<td>Chapter 23</td>
<td>Options</td>
<td>345</td>
</tr>
<tr>
<td>Chapter 24</td>
<td>Help Message</td>
<td>349</td>
</tr>
<tr>
<td>Chapter 25</td>
<td>Error messages</td>
<td>351</td>
</tr>
<tr>
<td>Appendix A</td>
<td>More Information</td>
<td>353</td>
</tr>
<tr>
<td>Index</td>
<td></td>
<td>357</td>
</tr>
</tbody>
</table>
Introduction to Desktop Intelligence
What is Desktop Intelligence?

Desktop Intelligence 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 Desktop Intelligence document.

Desktop Intelligence makes it easy to access this data, because you work in familiar business terms and not technical database terms like SQL.

Once you've used Desktop Intelligence to access data, you can present the information in reports as tables, or as sophisticated dynamic documents with drillable charts.

Where does the data come from?

Desktop Intelligence 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 Business Objects Designer. The designer then makes universes available to you and other users, to access data from the database through an intuitive, user-friendly interface.

Universes are made up of classes and objects.

Objects are elements that map to a set of data from a relational database using business terms. These objects allow you to retrieve data for your documents.

Classes are logical groupings of objects.

Using this interface, you build a Desktop Intelligence 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, Desktop
Intelligence 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.

Desktop Intelligence lets you access data from a wide range of sources: from relational and multidimensional databases, from packaged applications, from personal data documents, 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.

Alternatively, you can create sophisticated reports containing large amounts of data, organized and formatted to make it easy to go directly to pertinent information.

You can add images and embedded objects and format your documents to high presentation standards.

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 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 Exporting them to the repository as Folders or Categories. When you distribute documents in these different ways, you use the Desktop Intelligence 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 also which users in the company have the right to retrieve and view a document.
You can import documents that other users have sent, using Web Intelligence documents which you can open and view in Desktop Intelligence. You can also use InfoView to send documents for scheduled processing.

**Note:**
For information on sending, retrieving, printing, publishing and scheduling documents, see the InfoView User’s Guide. You can open an electronic version of this guide directly from the Desktop Intelligence Help menu.

## Security

The repository is set up and administered by the Business Objects administrator who grants all user rights.

The Business Objects administrator does the following:

- defines the parts of the Desktop Intelligence interface you can access
- restricts the availability of Desktop Intelligence functionality, such as access to certain menu commands
- defines your database connection
- defines 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 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. Desktop Intelligence 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, Desktop Intelligence reconnects to the database, and retrieves the updated data. This is called a document.
Demo materials and samples

To help you get up and running with Desktop Intelligence, demonstration databases, universes and sample reports are included in the Desktop Intelligence demo kit. There are two demonstration universes, Island Resorts Marketing and 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 on Demonstration Materials.

Upgrading from earlier versions of Desktop Intelligence

For users who are upgrading from an earlier version of Desktop Intelligence, previously known as BusinessObjects.

Documents created in BusinessObjects from 5.1 to 6.5 are fully compatible with Desktop Intelligence.

The Repository

Desktop Intelligence uses the repository to secure access to your data warehouse and to provide an infrastructure for distributing documents to be shared with others.

You select the documents you want to import from or export to Desktop Intelligence.

Folders and Categories

The Repository organizes documents into Folders and Categories in an orderly system that permits easy access for you and others working with documents.
Folders

- Folders are the physical place where documents are stored.
- Only one document with a given name may be placed in a folder or category.
- It is possible to place documents in several categories.
- If necessary, change the name of the document or give it a number to place it in the same folder or category.
- Shortcuts and copies may be placed in other folders or categories.
- Your Repository is organized into Folders and Categories to help you organize your documents. It is possible to create or delete sub-folders.
- Make sure that your document is saved before you export it to the repository.
- You are able to browse the Folders structure or the Categories structure.

Categories

- Categories are used for classifying information regardless of its storage location.

There are 2 types of folders:

- My Folders with 2 sub-folders
  - Favorites (Generally reserved for often used documents)
  - Inbox (Generally reserved for documents received from other users)
- Public Folders (For shared documents.)

There are two types of Categories:

- Corporate Categories
• Personal Categories
Introduction to Desktop Intelligence

Folders and Categories
Introduction to Accessing Data with Desktop Intelligence
What data sources are available?

Desktop Intelligence lets you access data from a wide range of sources. You can access data from a number of sources:

- Universes
- Personal Data Files
- Stored Procedures
- Freehand SQL Server
- XML Data Provider
- VBA Data Provider

How do you access data sources?

Desktop Intelligence lets you access 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 Desktop Intelligence, you build a data provider.

The types of data provider that Desktop Intelligence supports are described in the table below:
### What data sources are available?

<table>
<thead>
<tr>
<th>Data provider</th>
<th>Description</th>
<th>CD Install</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>Yes</td>
</tr>
<tr>
<td>Personal data files</td>
<td>You can retrieve data from Excel, dBASE and text files.</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 an SQL (Structured Query Language) script, saved and executable on your database.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Data Provider Table

<table>
<thead>
<tr>
<th>Data Provider</th>
<th>Description</th>
<th>CD Install</th>
</tr>
</thead>
<tbody>
<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>
</tr>
<tr>
<td>XML Data provider</td>
<td>You can retrieve data from XML files</td>
<td>Yes</td>
</tr>
<tr>
<td>VBA Data provider</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>
</tr>
</tbody>
</table>

### Can all Desktop Intelligence users build data providers?

Your Desktop Intelligence 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 Desktop Intelligence features depends entirely upon the query and reporting needs of your organization.
By default, all Desktop Intelligence users can refresh data providers to get the latest information from their database.

**Who sets up database connections?**

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 Desktop Intelligence "talk" to this data source.

In most cases, you, the Desktop Intelligence end user, do not have to concern yourself with setting up database connections. Thus, Desktop Intelligence lets you get the information you need, without technical knowledge of what's going on behind the scenes.

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

The following table describes who sets up database connections for the various Desktop Intelligence data providers.

<table>
<thead>
<tr>
<th>Data provider</th>
<th>Who sets it up?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries on universes</td>
<td>The universe designer sets up the connection in the universe, so the connection is hidden when you build or edit queries.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The supervisor may modify the existing connection or assign a new connection to the universe</td>
</tr>
<tr>
<td>Stored procedures</td>
<td>The supervisor creates the connection to access a stored procedure.</td>
</tr>
</tbody>
</table>
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 Desktop Intelligence; it is the username/password of the underlying database (for example an SQL Server database) that the universe accesses. This database normally remains hidden, but the universe designer can set up a restrictive 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").
When you refresh a query (for more information on refreshing a query, see "Refreshing Desktop Intelligence Documents".

When you parse a query to test its validity (for more information on parsing a query, see "Using SQL from Desktop Intelligence queries".

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

Can you combine data from different sources in one report?

Yes. With Desktop Intelligence, 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.

Workflows for accessing data

There are two basic workflows for building data providers to access your data in Desktop Intelligence. You can build a data provider for two reasons:

• to create a new document
• to work with an existing document.

Also with 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 Desktop Intelligence. 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, Desktop Intelligence launches the New Report Wizard when you start the application for the first time.

**To build a new data provider using the wizard**

2. Select an option for the report layout.
3. Click **Begin**.
   
   The Specify Data Access dialog box appears.
4. Make a selection depending on how you want to build your query (use the Choices element to wrap the following list):
   
   • To build a query on a universe, click **Universe**, then **Next**.
   
   • To build a query based on a stored procedure, free-hand SQL, personal data file, XML file, or VBA procedure, click **Others**, then select a data source from the list, then click **Finish**.
   
   • To build a query on a universe using the Query Panel, click **Universe**, then **Finish**.

   If you selected **Others** in the previous step, a dialog box appears to let you build your data provider and retrieve the data for your report.

   If you selected **Universe** and clicked **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." (ts_note: Make this a related-link.)

**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.
2. Click the **New Document** tab.
3. Click **Invoke the New Report Wizard with the following settings**.
4. In the Data Access group box, select the type of data provider you want to use.
   - Use a Default Universe lets you select the universe you want.
   - Use a Different Data Provider lets you select a data provider type from the drop-down list.
5. 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 Desktop Intelligence. 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.

Your company’s sales information is stored in your corporate database, which you access by running a query on a universe in Desktop Intelligence. You already have a Desktop Intelligence 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 compare the corporate figures with your personal data

1. Open the document containing the corporate data.
2. Click **New Data Provider**
3. Click **Access new data in a different way**.
4. Click **Personal data files**.
5. Click **Finish**.
6. In the dialog box that appears, browse to the Excel file that contains your personal data.

7. Click **Run**.

   Desktop Intelligence 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 the data provider.

4. Click **Run**.

   Desktop Intelligence retrieves the data, making it available in the document.

Tip: If you want to see the new data as soon as Desktop Intelligence 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 add result objects to the existing data provider:**

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**.

Desktop Intelligence 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**

Other reasons for editing a data provider include the following:

- 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>Click <strong>OK</strong>.</td>
</tr>
<tr>
<td>More than one data provider</td>
<td>Select the data provider you want to edit, then click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

3. Edit then run the data provider.

Desktop Intelligence returns the new data set to your report.
To cancel a data provider

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:

<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 <strong>Continue the execution</strong>.</td>
</tr>
<tr>
<td>Want to view the partial results created by the data provider when you interrupted the execution,</td>
<td>Click <strong>Stop the execution and keep the partial results</strong>. When you have partial results in a report, the Partial Results 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 <strong>Discard the results</strong>.</td>
</tr>
<tr>
<td>Want to view the results of the previous execution,</td>
<td>Click <strong>Keep the results of the previous execution</strong>.</td>
</tr>
</tbody>
</table>

Using the repository

Documents are placed in Folders and Categories in the repository

See "The Repository".
Exporting to the repository

When you create a document, before you can Export it to the Repository for the first time save the document and Export it to an existing folder or create a new folder.

Folders contain actual copies of your files, while Categories simply point to documents.

To Export a document to the Repository

1. With a saved document open in your Desktop Intelligence Administrator.
2. Click Export to Repository in the File menu.
   - Browse to the folder where you want to export your document, or create a new folder.
3. Highlight the folder where you want to export the document.
4. Click Add.
   - If your document has the correct name...
5. Click OK.
6. Click OK again.
7. Enter the summary information.
8. Click OK.
9. Click Replace.
   - If you do not click Replace, the export is aborted.
10. Click OK.

Creating a New Folder

When you export a document to a folder you must place it in an existing folder or create a new folder.
To create a new Folder

1. Click Export to Repository.
2. Highlight the file where you want to create your folder.
3. Click New.
4. Type the name of the folder.
5. Click OK.

Exporting to a Category

Save files to your local disk before exporting them to the repository for the first time. It is best to export the document to a folder before exporting it to a category.

To place a file in a Category

1. Open your file in the Desktop Intelligence Administrator.
2. Click Export to Repository in the File menu.
3. Click Categories at the bottom of the Dialog Box.
4. Activate the Category where you want to send your document.
   You can check more than one Categories.
5. Click OK.
6. Click Add.

To schedule export of a document

1. Click File > Export to Repository.
2. In the "Export" box, click Browse, and then locate and add the documents you want to send.
   To remove documents from the Document(s) to Send list, select the documents to remove and click Remove.
3. Select a repository folder to which to send the document.
• Click **New** to create a new repository folder.
• Click **Delete** to delete an empty repository folder (you cannot delete a folder that is not empty).

4. If desired, click **Categories** to associate categories with the document or **Clear** to clear the list.

5. Click **Schedule**.

6. In the "Send Document to Broadcast Agent" box, click the "General" tab and select your general options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formats</strong></td>
<td>Choose from the available formats in which to publish the document.</td>
</tr>
<tr>
<td><strong>Caching Option</strong></td>
<td>Choose whether or not to publish the document in the repository cache for efficient access, and the format. You can choose one or more of the available formats.</td>
</tr>
<tr>
<td><strong>Specify the Printer</strong></td>
<td>Click <strong>Default Printer</strong> and either <strong>Enable</strong> or <strong>Printing Options</strong> to specify printing options.</td>
</tr>
</tbody>
</table>

7. Click the "Change Schedule" tab to select scheduling options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Run</strong></td>
<td>Choose the publication frequency.</td>
</tr>
<tr>
<td><strong>Object will run now</strong></td>
<td>Choose further options depending on the frequency you selected under <strong>Run</strong>.</td>
</tr>
</tbody>
</table>

8. Click the "Distribution" tab to select distribution options.
   All choices except Default Enterprise Location give you the following options:
Managing Categories

When exporting a document to the repository, it is also possible to create a new category, delete an existing category, or rename one.

To manage your Categories

1. Open your file in the Desktop Intelligence Administrator.
2. Click **Export to Repository** in the File menu.
3. Click **Categories** at the bottom of the dialog box.
4. Click **Manage**.
5. Browse to the category you want to manage.
6. Activate the category.
   - Add a Category
   - Delete a category
   - Edit a category

To Add a category to the list of categories

With the Send to dialog box open:

1. Click **Categories** at the bottom of the dialog box.
2. Click **Manage**.
3. Activate the category you want to manage.
4. Click Add.

To Delete a category from the list of categories

1. Open your file in the Desktop Intelligence Administrator.
2. Click Export to Repository in the File menu.
3. Click Categories at the bottom of the dialog box.
4. Click Manage.
5. Browse to the category you want to delete.
6. Activate the category.
7. Click Delete.

To Edit a category

1. Open your file in the Desktop Intelligence Administrator.
2. Click Export to Repository in the File menu.
3. Click Categories at the bottom of the dialog box.
4. Click Manage.
5. Browse to the file you want to edit or rename.
6. Click Edit.

The right to create or delete a folder is controlled by the server. If you receive an error message, see your server administrator.

Importing from the repository

You can import from folders or categories.

To import a document from a folder in the repository

1. Choose folders or categories at the bottom left of the Browse Categories box.
2. Click Import from Repository in the File Menu.
3. Browse to the document you want to import.
4. Select the document.
5. Select **Open on retrieval**.
6. Click **Retrieve**.

**Retrieving different instances of a given document**

If you have more than one instance of a file, the **Retrieve Inst** button is activated.

Select the file you want to Import to the Repository.

**Note:**
- Instances are placed in folders using InfoView.
- Scheduling options are accessed from InfoView.

If the **Retrieve Inst** button is available, there is more than one instance of the document.

**To choose an Instance of a document**

1. Click the **Retrieve Instance** button.
2. Choose an instance of the document.
3. Click **Retrieve**.

**Sending documents from Desktop Intelligence**

You can send Desktop Intelligence documents either to other users of the Business Objects deployment or to other individuals via email.

Other users of the Business Objects deployment must have appropriate rights in order to access the document. If they do not, or if they are not a part of the Business Objects deployment and receive the document via email, the document must have been saved with the **Save for all users** option selected in order for them to have access to the document.
To send documents to users and groups from Desktop Intelligence

1. Click File > Send To > Users.
2. In the Send documents to users box, click Browse, and then locate and add the document you want to send. The document appears in the Document(s) to Send list. If a document was already open in Desktop Intelligence, it appears by default.
3. In the "Send documents to users" box, click To.
4. In the "Select Users and Groups" box, click Groups to see groups of users or Users to see individual users. You can refresh the list by clicking Refresh.
5. Select one or more in the list. Use Ctrl+click for multiple selection.
6. Click Add to add your selection to the Document Recipients list. You can choose an option, Groups or Users, add users to the Document Recipients, and then do the same with the other option.
7. To remove users or groups from the Document Recipients list, select them and click Remove.
8. Click OK.

The selected documents are sent to the Business Objects inbox of the document recipients.

To send documents by email with Desktop Intelligence

You must have an email client on your machine that is configured to open new emails with a valid email account. If the recipient is not a user of the Business Objects deployment with the appropriate rights to read the document, you must first save the document with the Save for all users option selected.

1. Open a Desktop Intelligence document.
2. Click File > Send To > Send to Mail.
An empty email is opened by your email client. The Desktop Intelligence document is attached to the email.

3. Specify the desired recipient and subject, add a message if desired, and then send the email.

The email with the attached document is sent to the email address you specified.

Sending documents from Desktop Intelligence

You can send Desktop Intelligence documents either to other users of the Business Objects deployment or to other individuals via email.

Other users of the Business Objects deployment must have appropriate rights in order to access the document. If they do not, or if they are not a part of the Business Objects deployment and receive the document via email, the document must have been saved with the Save for all users option selected in order for them to have access to the document.
Building Queries on Universes
Overview

This chapter is about accessing data using Desktop Intelligence native technology: building queries on universes.

What is a universe?

Desktop Intelligence 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 an 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 Desktop Intelligence. 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. Desktop Intelligence connects to the database, and retrieves the data mapped by the objects you selected. Desktop Intelligence retrieves this data by executing an SQL query against the database; Desktop Intelligence generates this SQL according to the objects you select. SQL stands for Structured Query Language; it is the query language understood (in various dialects) by all relational databases.

Note:
SQL queries generated by Desktop Intelligence cannot exceed 65,536 characters in length.

Demonstration materials

Several demonstration databases, and their accompanying universes and reports are included in the Desktop Intelligence package. They are installed with Desktop Intelligence, and used in the examples in this guide. The databases are compatible with Microsoft Access 2000. The Desktop Intelligence 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 below shows the universe’s classes and objects as they appear in Desktop Intelligence.

Because universes provide a business-intelligent semantic layer between you and the database, the names of the classes and objects in the
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 Desktop Intelligence, refer to *Classes and sub-classes* and *Dimension objects, measure objects and detail objects*.

**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:

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 in Displaying the query panel.

There are three steps in building a basic query on a universe.

- Display the query panel
- Build the query in the Query Panel and run the query
- 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 New Report Wizard button</td>
</tr>
<tr>
<td></td>
<td>(Standard toolbar).</td>
</tr>
<tr>
<td>to edit a query or other type of data provider</td>
<td>click Edit Data Provider on the</td>
</tr>
<tr>
<td>in the current document,</td>
<td>Data menu.</td>
</tr>
</tbody>
</table>
If you want to create a new query or other type of data provider in the current document, click **New Data Provider** on the Data menu.

If you need more information, refer to "Workflows for accessing data."

The Query Panel displays the contents of your Desktop Intelligence universe and lets you select data with simple mouse clicks.

In the **Classes and Objects tree** on the left side of the screen, display is as follows:

- **Classes** appear as folders.
- **Objects** appear as cubes (for dimensions), spheres (for measures), or pyramids (for details).

The button on the bottom left, under the Classes and Objects tree, with an icon showing the different kinds of objects, is selected by default and controls display of the universe's classes and objects.

The button to its right displays the universe's predefined conditions.

The search box next to these buttons lets you type a search string to search for objects in the universe.

The **Options** button enables you to set options before running the query, for example to specify a maximum number of rows.

The **Result Objects** box displays the objects that are included in the query.

The **Conditions** box displays the conditions limiting the data returned by the query.

The **Save and Close** button lets you save the query you have defined without running it. You can run it later on by using the Refresh command.

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.
When you click Run, the query connects to the database and the data appears in the report.

To display the query panel

You've launched Desktop Intelligence 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.

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 is displayed.

Building a query in the Query Panel and running the query

Building and running a query includes the following steps:

• Display all the objects that you can include in a query
• Include objects in a query
• Remove objects from a query
• Change the order of objects in a query
• 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. Desktop Intelligence 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 Desktop Intelligence 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.
To remove objects from a query

1. In the Result Objects box, click the icon of the object you want to remove.
2. Drag the icon to the Classes and Objects list.
3. Press the Delete key, or right-click the icon and click Delete.

To change 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.

1. Click an object in the Result Objects box.
2. To change the object's position, drag the icon to the left or right.
3. To swap the icon's position with that of another object in the Result Objects box, press Shift while dragging the object icon until it is above the icon of the other object, then release the 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: to build a query in the Query Panel and run 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.
3. Click Run.
Desktop Intelligence retrieves the data for Resort, Year and Revenue and displays this in a new document.

Note: If the universe designer has set the up the universe with a restrictive connection, Desktop Intelligence prompts you to enter your database username and password before retrieving the data. For more information on restrictive connections, see “Restrictive connections.”

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.

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," 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. In it:

• A check appears next to objects that are included in the scope of analysis.

• Hierarchies are represented by folders.

• You can click the + sign to the left of the hierarchy's folder to view the objects it contains.

Note:
If a universe contains no hierarchies, Desktop Intelligence uses its classes as hierarchies by default.
Defining 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.

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 of limiting query results by using a condition.

The Resort object retrieves five values: Australian Reef, Bahamas Beach, French Riveria, 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 Desktop Intelligence, you can set three types of conditions on a query:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined conditions</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>
</tbody>
</table>
### Table: Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple conditions</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>Complex conditions</td>
<td>Enable you to limit the query results by any object in the universe. For more information on complex conditions, refer to &quot;Applying complex conditions on queries.&quot;</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.

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.

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

### To remove a predefined condition

1. Click the condition’s icon in the Conditions box.

2. 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.

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 the List of Values dialog box.

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.

1. In the Conditions box of the Query Panel, click the value(s) that appear(s) on the right-hand side of the condition.
2. The Classes and Objects box becomes the Operands box.
3. Double-click the Show list of values operand.
4. The object’s list of values appears in the List of Values dialog box.
5. 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.
6. Click any selected values that you do not want the object to retrieve, and click **OK**.

To delete a simple condition

1. Click the condition in the Conditions box
2. 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.
3. A sort icon appears below the object icon in the Result Objects box.

To remove a sort

There are two ways of doing this:
1. Click the sort icon, then press the Delete key.
2. Alternatively, 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.

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, Desktop Intelligence 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 automatically switch off, meaning that the query will connect to the database and the data will appear in the report.

5. 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

Desktop Intelligence 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.

To run a query on a different universe

1. Open the report containing the query.
2. Click **View Data** on the Data menu.
   The Data Manager dialog box opens.

3. Choose the query you want to use in the Data Providers list.

4. 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.

7. Click **OK**.

8. Click the **Results** tab.

9. Click **Refresh**.

10. Click **OK** to close the Data Manager.
Building Queries on Universes

Running a query on a different universe
Building Queries with Other Types of Data Provider
Overview

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

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 Desktop Intelligence 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

Click the New Report Wizard button on the Standard toolbar.


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

2. Under Others, select Free-hand SQL from the list box, then click Finish.
   
   The Free-hand SQL editor appears.

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

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

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

5. 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).

6. 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.

7. 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.

8. 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 edit a free-hand SQL script

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.
   
   • Click **Run**.

   Desktop Intelligence 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 Desktop Intelligence. 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.

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/data source 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 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.

To create an eFashion report that shows sales by store and category in Florida

1. Start Desktop Intelligence
3. Select an option for the report layout, then click Next.
4. Select Free-hand SQL. 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.

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.

7. Type the following SQL in the Free-Hand SQL dialog box:

```
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 = 'Florida'
GROUP BY ol.shop_name, al.category
```

8. Click Run.
Creating interactive reports using free-hand SQL

This section gives an example of a free-hand SQL script that includes a Desktop Intelligence prompt. When you run a report containing a prompt Desktop Intelligence 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 Desktop Intelligence rather than an SQL feature, but the Desktop Intelligence 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:

\[
@\text{prompt} \left('\text{prompt}', \text{data type}', \{',\text{value1}',',\text{value2}',
\text{etc.}\}\right),\text{mono/multi,free/constrained}
\]

The following table describes each function component:
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>@prompt</td>
<td>The <code>@prompt</code> function, which can take up to five arguments. The only mandatory argument is 'prompt'. If you omit an argument, Desktop Intelligence 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:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>@prompt('Which year?'',',',')</code></td>
<td></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>
<td></td>
</tr>
<tr>
<td></td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</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>
<td>'A'</td>
</tr>
<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>Argument</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</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>

To create a prompted eFashion report on sales by state, store and category

1. Click **New Report Wizard**.
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 for an explanation of how to do this).

5. Type the following SQL into the Free-Hand SQL dialog box:

```sql
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.

   Desktop Intelligence 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 Desktop Intelligence 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 Desktop Intelligence 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 Desktop Intelligence 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 Desktop Intelligence?

In Desktop Intelligence, 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 Desktop Intelligence which Desktop Intelligence presents as a report.

Restrictions on stored procedures

- The Desktop Intelligence 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 Desktop Intelligence 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)
   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
```

To create a report showing article sales by state

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. 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:
Then... If you want...

<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 Use this value in the Next Execution dropdown</td>
</tr>
<tr>
<td>Desktop Intelligence to prompt you for a value the next time you run the report</td>
<td>Select Prompt me for a value in the Next Execution list</td>
</tr>
</tbody>
</table>

7. Click **Run**.

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

---

**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 Desktop Intelligence 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 To select the personal data file for the report.
- 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.

To select 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:
   The Access Personal Data dialog box appears.
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.
   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 column headers in the report</td>
<td>Select <strong>First row contains column names</strong></td>
<td>All files</td>
</tr>
<tr>
<td>Create drill hierarchies and open the report in drill mode. (Desktop Intelligence can do this only if the dimensions in your report have a hierarchical structure, for example Year, Quarter, Month)</td>
<td>Select <strong>Build hierarchies and start in drill mode</strong></td>
<td>All files</td>
</tr>
<tr>
<td>Specify the delimiter in a text file</td>
<td>Select <strong>Tabulation, Space or Character.</strong> (If you select Character, you need to enter the character that delimits the data.)</td>
<td>Text Files (.asc; .prn; .txt; .csv)</td>
</tr>
<tr>
<td>Select the worksheet containing the data you want</td>
<td>Select the worksheet from the Sheet Name list</td>
<td>Microsoft Excel (.xls)</td>
</tr>
<tr>
<td>Select data from all fields in a worksheet</td>
<td>Select <strong>All Fields</strong></td>
<td>Microsoft Excel (.xls)</td>
</tr>
<tr>
<td>Select data from a range of cells in a worksheet</td>
<td>Type the range (for example A3:R25) in the Range Definition box</td>
<td>Microsoft Excel (.xls)</td>
</tr>
</tbody>
</table>
5. 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 Desktop Intelligence. The DpVBAInterface is the interface to the VBA data provider Automation object which is described in detail in the *SAP BusinessObjects Desktop Intelligence Developer Guide*.

**To write a VBA data provider**

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 Desktop Intelligence.
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.
4. Select the subroutine and click **Run**.  
Desktop Intelligence generates the report.

Accessing an Outlook inbox using VBA

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

To reference the Outlook object library

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

To enter the code of the VBA data provider.

2. Select an option for the report layout.
3. Click **Next**.
   The Specify Data Access dialog box appears.
4. Click **Visual Basic for Applications procedures** under Others.
5. Click **Finish**.
   The Access Data from VBA dialog box appears.
6. Type Outlook for the subroutine name.
7. Click **Create**.
   The Visual Basic editor opens with a skeleton subroutine.
8. Type the following code:
   ```vba
   Public Sub Outlook(dpInterface As DpVBAInterface)
   Dim olkApp As Outlook.Application
   Dim nspNameSpace 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 nspNameSpace = olkApp.GetNamespace("MAPI")
   Set objInboxFolder = nspNameSpace.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(7) 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
End If
oCol.Qualification = boDimension
oCol.Item(row) = oColData(col)
Next col
Next row
dpInterface.CheckDataIntegrity (boCheckAll)
End Sub

9. Click **Compile** on the Debug menu to compile the project.
10. Click **Close and Return to** Desktop Intelligence on the File menu to return to Desktop Intelligence.
11. Click **Run**.
    Desktop Intelligence generates the report.

**Using XML files**

This section describes how to use XML as a Desktop Intelligence 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>
```

Building Queries with Other Types of Data Provider
Using XML files
<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>
<Resort>
<Country>US</Country>
<ResortName>Bahamas Beach</ResortName>
<ServiceLine>Recreation</ServiceLine>
<Revenue>128100</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 Desktop Intelligence.
Creating a report using an XML file

Creating an XML-based report involves two steps:

• building an XML filter
• building the report

To build 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.

2. Select a layout option.
3. Click Begin.
4. Click Others and select XML Data Provider from the Others box.
5. Click Next.

   The Create XML Filter dialog box appears.

7. Click Load XML (the first button at the top, with an open folder icon), then use the Open dialog to navigate to and select the XML 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.

8. 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 Desktop Intelligence report, it appears as Resorts_00001, Resorts_000002.

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

10. Click Save.
The Save XML Filter As dialog box appears.

11. Type a filter name in the New Filter Name text box
12. Click OK.

The filter appears in the list of XML filters.

To build an XML report

You build reports based on XML using the XML filters that you have defined.

2. Select a layout option.
3. Click Begin.
4. Click Others.
5. Select XML data provider from the Others box.
6. Click Begin.
7. Select the filter in the list of filters.
8. Click Finish.

The XML Query Panel appears, showing the elements you selected when you built the filter available for inclusion in the report.

9. Double-click the elements that you want to include in the report.
10. Click Run.

Desktop Intelligence generates the report.

To set the location of XML files

When you create an XML filter, Desktop Intelligence creates a file with the filter definition. You can tell Desktop Intelligence where to store XML filter files and XML files.

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.
4. Select **XML Filters** to change the location of XML filter files.
5. Click **Change**.
   
   The Browse for Folder dialog box appears.

6. Use the Browse for Folder dialog box to select the folder where you want the files to be stored.
7. Click **OK** to close the Browse for Folder dialog box.
8. Click **OK** to close the Options dialog box.
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. Desktop Intelligence 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 Desktop Intelligence automatically links data from different sources, and when you have to make the link yourself

Which data sources are available?

Desktop Intelligence 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
- text files and spreadsheets
- packaged applications such as SAP.
- almost any data source using Microsoft Visual Basic for Applications (VBA) procedures
- XML files

Including data from different data sources in the same report

You access data sources in Desktop Intelligence 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. Desktop Intelligence supports the following types of data providers:

- queries on universes
- stored procedures
- free-hand SQL
- personal data files
- VBA procedures
- SAP
- XML

**Which data providers can you combine in one report?**

You can combine data from any Desktop Intelligence-supported data provider with data from any other Desktop Intelligence-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, or a personal data file.

**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 Desktop Intelligence 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**

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.

   Desktop Intelligence 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. Desktop Intelligence will prompt you to link them.

   and

   • 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**.
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 Desktop Intelligence 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.

You can use the common dimension from either data provider. Most often, you can use measures from both data providers in the same block.

To display 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**.

### To display 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 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.

### To add 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**.
Basing a data provider on an existing data provider

You can base new data providers on existing data providers that use universes, or personal data providers.

To base a data provider on an existing data provider

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.
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, or a personal data provider.

A list of data providers currently in the document appears.

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

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

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 names 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* 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?

Desktop Intelligence 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. Desktop Intelligence 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; Desktop Intelligence automatically inserts the new columns of data in the report or creates a new report.

**Example:** Desktop Intelligence prompts you to link data providers

Here's an example scenario where Desktop Intelligence 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, Desktop Intelligence 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. Desktop Intelligence displays a dialog box which prompts you to link the new data provider with the data in the report.
7. Select the linking dimension from the new data provider by clicking an icon in the Dimensions box.
8. Select the report's section master by clicking an icon in the Master(s) in the Report box.
9. The dimension you clicked in the Dimensions box appears below the dimension in the Master(s) in the Report box:
10. 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.

**To link existing data providers**

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.
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.

**To delete the link between data providers**

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.
Combining Data from Different Sources

Linking data providers
Managing Data Providers
Overview

In what ways can you manage queries and other data providers in Desktop Intelligence 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 ), and delete data providers you no longer need

Note:
Data provider is the Desktop Intelligence generic term for all types of queries: queries on universes, free-hand SQL, stored procedures, VBA procedures or XML files.

Renaming data providers

Desktop Intelligence 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>
</tbody>
</table>
### Why rename data providers?

Renaming data providers is by no means required in Desktop Intelligence. You can work with the software without ever thinking about these names. However, data provider names appear in several places in Desktop Intelligence, 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, Desktop Intelligence 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.
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
4. Click the Definition tab.
5. Type the new name in the Name box.
6. Click OK.

Getting statistics on data providers

Desktop Intelligence 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 statistics on data providers

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.
4. 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 your hard drive, 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.
4. Click Purge/Delete.
5. Click Yes in the confirmation box that appears.
6. Click OK to close the Data Manager.

Tip: 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 Desktop Intelligence 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.

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.

Example: To create a data provider using certain objects

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 for an explanation of how to build a report on a universe.)
3. Drag the Country and Revenue objects to the new report tab to create the revenue by country report.

When you run this report, Desktop Intelligence 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).
Introduction to Data Analysis
Overview

With Desktop Intelligence, 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 Desktop Intelligence interface to address all your multidimensional analysis needs:

- Desktop Intelligence 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 Desktop Intelligence, enables you to carry out multidimensional analysis in Drill mode.
- Desktop Intelligence 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 Desktop Intelligence 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. Desktop Intelligence 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.
Desktop Intelligence drill mode

Desktop Intelligence 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" for more information on drill mode.

Slice and dice mode

Chapter 9, "Slice and Dice Mode," 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.
Slice and dice mode
Analyzing Data in Drill Mode
Overview

This chapter describes how to use Desktop Intelligence drill mode.

What is drill mode?

Drill mode is a Desktop Intelligence 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.

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 Desktop Intelligence universe, the objects you can include are grouped in folders and organized in a specific order.

Classes are represented as folders in the Classes and Objects tree. The objects in a class named Sales, for example, are all connected to sales. They may be arranged in hierarchical order with Sales Person as the highest level object, then down through Year to Invoice Date, the finest level of detail. Measure objects are stored in a separate folder.
The person who creates the Desktop Intelligence 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 used 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 Desktop Intelligence 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. Desktop Intelligence 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."

Using drill mode

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

To switch to drill mode

You open drill mode from a Desktop Intelligence report.

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 (a magnifying glass icon) 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. If this happens, click inside the table, chart or crosstab you want to analyze.

What happens when you enter Drill mode

- Desktop Intelligence 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.
- Desktop Intelligence adds a sum on measure objects.
- If you are working on a master/detail report, Desktop Intelligence displays the Drill toolbar.

Note:
You can change these default behaviors. See "Setting options for working in drill mode" 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. For example: 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. 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 Desktop Intelligence 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.

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.

To change values in the toolbar

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

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

To drill 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.

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.

   Desktop Intelligence displays the data for the next level up.

Undoing drill actions

You can undo up to ten drill actions, which can be useful if you lose track of your analysis.

- 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
- Create new hierarchies inside the report.

---

**To drill across and 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.

2. Choose a dimension from the list.
   
   The list in the Drill By sub-menu displays five dimensions only by default. To display the full list of dimensions:
   
   3. Right-click a value and click **Drill By More** on the shortcut menu.
      
      The Drill By - All Available Dimensions dialog box appears.

4. Choose a dimension from the list

5. 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 for more information.

Drilling on charts

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

To drill on charts

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.
To drill on multiple hierarchies

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

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.
   Desktop Intelligence 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 Desktop Intelligence updates the Revenue column.

To drill up on multiple hierarchies

1. Right-click on a measure object value
2. 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.
## Analyzing Data in Drill Mode

### Getting a different view of your data

<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 to a table or crosstab in the report.

• You can move variables from the Drill toolbar to a table or crosstab in the report.

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>Insert 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</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 Drill toolbar shows the Replace highlighting the status bar displays</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.
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.
To remove 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.

To move 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 (arrow cursor) the status bar displays Drop to insert</td>
</tr>
</tbody>
</table>
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."

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**

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 (camera icon).

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).

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.

To expand the 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.

• To include all the dimensions from a hierarchy, click the hierarchy's check box.
• Click on a filter box to limit the data retrieved.
• Check the dimension box to include just the individual dimensions you need.
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. Desktop Intelligence 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.

**To drill through the database to bring in new data**

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. The dimensions that are currently available in the report are grayed out. All dimensions that belong to drill hierarchies but are not currently available in the report are displayed in black.

3. Choose a dimension and click **OK**.

   The dimension is retrieved from the database and displayed in the block you are analyzing. You can choose only one dimension 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: To 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.

1. Click **Options** on the Tools menu.

   The Options dialog box appears.

2. Click the **Drill** tab.

3. Click the Apply Drill Filters option in the Drill Through section.

   You can now retrieve the new data from the database. You drilled down on Country and the table displays revenue for US resorts only.

4. 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.

5. Choose Country of Origin from the Customer hierarchy and click **OK**.

   Desktop Intelligence retrieves the data, displays Country of Origin in the table and moves Resort to the Drill toolbar.

   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.
5. click OK to close the Hierarchy Editor.
To add a dimension to an existing hierarchy

1. Click the dimension you want to add to the hierarchy,
2. 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."

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."

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,
6. Click Add.
7. The dimension you clicked appears in the new hierarchy's folder in the Available Hierarchies box.
8. 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." Note that you cannot combine dimensions from different data providers in a single hierarchy.

Qualifying data for hierarchies

Desktop Intelligence 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.

**To requalify 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.

**To requalify 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**.

**To requalify user objects**

1. Click **Universes** on the **Tools** menu.
   
   The Universes dialog box opens.
2. Select the universe that contains the user object,
3. 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.

**To insert 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.
To print 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
4. Click **OK**.
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

When working in slice-and-dice mode, you use the Slice and Dice Panel.

To display the Slice and Dice Panel:

1. Click Slice and Dice on the Analysis menu.
2. Alternatively, click Slice and Dice on the Standard toolbar.

   The Slice and Dice Panel appears.
   • The first button on the top left lets you show/hide the Available Variables box.
   • The next set of buttons lets you apply breaks, filters, sorts, rankings and calculations.
   • The report variables are shown in the Available Variables tree.
   • The options on the bottom left let you view all variables, dimensions only, measures only, or all variables by data provider.
The Section pane shows masters in master/detail reports.

The Block Structures pane shows the variables in the active table, chart or crosstab.

The Crosstab 1 tab shows the name and type (table, chart or crosstab) of each block.

You can rename, transform, reset or delete a block by right-clicking its tab, then clicking a command on the shortcut menu.

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.

- The Available Variables box shows the variables and formulas you can display in the report.
- The Section box shows the master.
- The Block Structure box shows the data that appears in the table.

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 or master/detail report
- undo a master/detail report by removing the master
- deactivate sections of master/detail reports

To structure an existing report as a master/detail report in slice and dice mode

1. Drop the icon of the master in the Section box.
2. To use available data, drag an icon from the Block Structure box, drop it in the Section box, then click Apply.

3. To use data that is not yet displayed in the report, drag an icon from the Available Variables box, drop it in the Section box, then click Apply.

To structure an existing report as a master/detail report in drill mode

1. Switch to drill mode.
2. Select the data you want to use as master.
3. Right-click.
4. Click Set as Master on the shortcut menu.

Reorganizing a master/detail report

You can reorganize a master/detail report by:

• using a different master
• building a master/master/detail report

   This structure enables you to view data on two levels of detail.

To reorganize a master/detail report using a different master

1. Open a master/detail report.
2. Select the icon of the master in the Section box, then press the Delete key.
   The icon disappears from the Section box.
3. To replace the master with data that is already displayed in the report, drag an icon from the Block Structure box, and drop it in the Section box.
4. To swap the master with data that is displayed in the report, hold down the Shift key, then drag the master until it is located above the icon with which you want to swap, in the Block Structure box, release the mouse button and click Apply.
5. To replace the master with data that is not yet displayed in the report, 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.

**To reorganize a master/detail report by building 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.

1. Open a master/detail report.
2. Open the Slice and Dice panel.
3. Drag an icon to the Section box.
   - You can drag an icon from the Available Variables box, or from the Block Structure box.
4. Drag the icon until it is located just below the existing icon, then release your mouse button.
5. 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.

1. Drag the master from the Section box to the Block Structure Box
2. Click **Apply**.
   - This moves the master to a block.
3. Select the master in the Section box.
4. Press the Delete key.
5. Click **Apply**.
   - This removes the master from the report.
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.

One section of the report 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.

To obtain the number of guests per quarter for all resorts, rather than the number of guests per resort, you deactivate 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, Desktop Intelligence 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,
2. Open the Slice and Dice Panel.
3. Click the tab of the block (table, chart or crosstab) that you want to recalculate.
4. In the Section box, right-click the master of the section that you want to deactivate.
5. Click Deactivate this section on the shortcut menu.
6. 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.
To reactivate a section of a master/detail report

Once you have deactivated a section of a master/detail report, you can reactivate it.

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, right-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, Desktop Intelligence 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, right-click the master of the uppermost section that you want to deactivate.
4. Click **Deactivate this section and all sections below it** on the shortcut menu.
5. Click **Apply**.

Desktop Intelligence recalculates the report.

6. Click the master of the lowest section with your right-mouse button (Quarter, in the example above).
7. 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

1. Select an icon.
2. Drag the icon horizontally, left or right, to its new position.
   As you move the mouse, the cursor changes to an arrow shape.
3. Release the mouse button to drop the icon at its new position.
4. Click Apply to display the data in its new position in the report.

To swap two variables horizontally

1. Hold down the Shift key.
2. Select one of the icons you want to swap.
3. The cursor changes to show that you are carrying out a swap operation, into an arrow shape with two smaller arrows pointing in opposite directions.
4. Drag the icon horizontally until it is above the other icon you want to swap, and release the mouse button.
5. 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.
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".

**To build a crosstab or a 3-D matrix chart from a table or 2-D chart**

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 in Working in slice-and-dice mode.
5. Click **Apply** to make the crosstab or matrix chart appear.
6. To use data already displayed in the report, take an icon from the Block Structure or Section box.
7. To use data not yet displayed in the report, take an icon from the Available Variables box.

**Reducing 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
To reduce a cross-tab or 3-D matrix chart to a table or 2-D chart

1. Click inside a crosstab or 3-D matrix chart.
2. Open the Slice and Dice Panel.
3. 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.
4. If you want to remove the data from the report, press the Delete key.
5. 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.
6. Release your mouse button.
7. Click Apply to make the table or 2-D chart appear.

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:

To move a variable up or down

1. Select an icon.
2. Drag the icon vertically, up or down, to its new position.
3. As you move the mouse, the cursor changes to an arrow shape.
4. Release your mouse button to drop the icon at its new position.
5. Click Apply.

To swap two variables vertically

1. Hold down the Shift key
2. Select one of the icons you want to swap.
   The cursor changes to show that you are carrying out a "swap" operation, to an arrow shape with two smaller arrows pointing in opposite directions.
3. Drag the icon vertically until it is above the other icon you want to swap.
4. Release your mouse button.
5. Click Apply.

Moving 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

1. Select an icon in the upper right-hand corner of the Block Structure box.
2. Hold down your mouse button and drag it to the bottom left-hand corner of the box.
3. As you move the mouse, the cursor changes to an arrow shape.
4. Release your mouse button.
5. Click Apply.

To swap two variables between columns and rows

1. Hold down the Shift key.
2. Select one of the icons you want to swap.
3. 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, to an arrow shape with two smaller arrows pointing in opposite directions.
4. Release your mouse button.
5. Click Apply.

Moving data between the X- and Z-axis in 3-D matrix charts

Moving data between the X- and Z-axis in 3-D matrix charts is equivalent to moving data between columns and rows in crosstabs. Follow the procedures described for moving data between columns and rows. 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 in Slice-and-Dice mode

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 in Slice-and-Dice mode

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.
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. Right-click.
   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.

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.

To transform blocks in Slice-and-Dice mode

In slice-and-dice mode, you can turn tables and crosstabs into charts, and vice versa.

1. Open the Slice and Dice Panel, then click the tab of the block you want to transform.

2. Right-click, then select a command.
   • To turn the selected table or crosstab to a chart, click Turn to Chart.
   • To turn the selected chart to a table, click Turn to Table.
   • To turn the selected matrix chart to a crosstab, click Turn to Crosstab.

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: in order, they are Sorts, Ranking, Calculations, Breaks, and 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.

**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.
3. To invert the sort, double-click the sort icon.
4. To remove the sort, select the sort icon, then press the Delete key.

**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.

5. To redefine the ranking, double-click its icon in the Slice and Dice Panel, then modify its attributes in the Select Top/Bottom Variable Name dialog box.

6. To remove the ranking, select its icon, then press the Delete key.

For further information on ranking and how to apply them, see *Filtering and Ranking Data*.

### 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 "Calculations" dialog box appears.

2. Select the function(s) you wish to apply, then click **OK**.

3. Click **Apply**.

   The calculations appear in the report.

### To apply a break

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.

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.

2. Click **Apply**.

   Desktop Intelligence applies the break to the report.
3. To redefine a break, double-its click icon. In the dialog box that appears, select different attributes for the break.
4. To remove a break, select its icon, then press the Delete key.

**To use 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.
5. To redefine the filter, double-click its icon in the Slice and Dice Panel, then modify its attributes in the Filter on dialog box.
6. To remove the filter, select its icon, then press the Delete key.
Filtering and Ranking Data
Overview

A Desktop Intelligence 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**

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.

To insert a filter

You can insert a filter on data in tables, crosstabs or charts.

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. The title of this dialog box depends on the data you select in the report.

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.

### To select different values for an 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.

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. Filters in the Global folder affect the whole report. Filters in other folders affect the table/crosstab/chart to which the folder refers.

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.

To add a filter

The Filters dialog box enables you to add filters to the report.

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 Block name 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 variables you clicked in the Variable(s) to Filter dialog box appear 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.
   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.

To remove a filter

When you remove a filter, Desktop Intelligence 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, Desktop Intelligence displays revenue for all regions.

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.

To display global filters you have applied to a whole report

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.

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, Desktop Intelligence inserts the names.

4. To insert the data in an existing cell, select the cell.

5. Click **Special Field**, then click **Global Filters** on the **Insert** menu.

   Desktop Intelligence 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."

**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 $200,000. The syntax is as follows:

\[
=\text{(<Revenue>)}=200000
\]

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 functions, variables or operators 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**.

**To edit a complex filter**

Desktop Intelligence identifies a complex filter with an asterisk next to the filter symbol.

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, Desktop Intelligence displays an error message. Click OK to return to the formula. Desktop Intelligence 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 Desktop Intelligence 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:

```plaintext
=NoFilter(formula)
```

The example below shows how this works.
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:

\[=\text{Sum}(<\text{SalesRevenue}>)\]

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:

\[=\text{NoFilter} \left( \text{Sum}(<\text{SalesRevenue}>)) \right)\]

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.

To apply 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.

To invert 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.

To remove 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.

To apply 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. Drag and drop the values in the list into the desired order.
4. Alternatively, use the Move Up and Move Down buttons to place the values in the desired order.

5. Alternatively, choose an option from the Sort Option list box:
   - Default: 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.
   - Alphanumeric: Displays values in alphabetical or numerical order.
   - Month: Displays the months of the year in chronological, not alphabetical order. Only applicable for a variable that returns the months of the year.
   - Day: Displays the days of the week in chronological, not alphabetical order. Only applicable for a variable that returns the days of the week.

6. Click OK to close the Custom Sort dialog box.

Note:
You cannot use custom sorts and alerters. For example, if you set up an alert to highlight months greater than May, Desktop Intelligence will use the alphabetical sort order instead of the chronological sort order.

To sort months correctly

This example shows a table that has two custom sorts. The months have been sorted to display in chronological order and the product lines have been manually sorted.

By default, Desktop Intelligence 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.
   Desktop Intelligence sorts the months correctly.
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

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.

---

**To 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. The primary sort is first in the list, followed by the secondary sort.
   
   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.

To add 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. Desktop Intelligence 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.

To apply a ranking on report data

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.
3. The Select Top/Bottom dialog box appears. It displays the options you can select for the ranking.
   • To select the largest n values, click Top and select the value of n. (Desktop Intelligence selects 3 by default when you click Top.)
   • To select the smallest n values, click Bottom and select the value of n. (Desktop Intelligence selects 3 by default when you click Bottom.)
   • To select the top n% of values, click Top, click In percentage of total number of values, then select the value of n.
   • To select the bottom n% of values, click Bottom, click In percentage of total number of values, then select the value of n.
   • To display subtotals relating to the values that appear in the report and the values that are omitted from the report, click Display subtotals.
   • To display percentages relating to the values that appear in the report and the values that are omitted from the report, click Display percentages.
   • To select the measure on which the ranking is based, choose the measure in the based on combo box.
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 Desktop Intelligence 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 Desktop Intelligence adds the following calculations to the report:

- each included row expressed as a percentage of all included rows (a)
- the total number of included rows expressed as a percentage of all rows (included and excluded) (b)
- the total number of excluded rows expressed as a percentage of all rows (c)
- the total number of included and excluded rows expressed as a percentage (which is always 100%) (d).

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.

Show sales revenue for top three months, compare with overall revenue

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.

**To edit 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.

1. Click inside the data where the ranking was applied.
2. Click **Ranking** on the **Format** menu.
   - The Select Top/Bottom dialog box appears, with the current ranking options selected.
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.

**To remove 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.

1. Click inside the data on which you applied the ranking.
2. Click the dimmed **Apply Ranking** button on the Report toolbar
   - Alternatively, click Ranking on the Format menu.
3. 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, Desktop Intelligence displays the following message: **The selection already has a sort and/or a filter. The current action will overwrite the sort and/or filter. Do you want to continue?**

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 hide columns and rows of data

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.

#### To hide data

1. In the Used Variables box, click the variable you want to hide.
   - to select more than one variable at the same time, hold down the Ctrl key while clicking the variables.
2. Click **Hide**.
   - Hidden variables are grayed in the Used Variables box.

Note that when you hide a dimension, Desktop Intelligence does not recalculate measures. Hiding a dimension is not the same as removing it from a report.
To show data

1. Click a previously hidden variable in the Used Variables list.
2. Click Show.

Highlighting data

You can highlight data in a Desktop Intelligence 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.

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.

To create 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.

**To select the data when creating an alerter**

1. In a report, click a cell, row or column of data.
2. Click **Alerters** on the **Format** menu.
3. In the Alerters dialog box that appears, click **Add**.

**To name and describe the alerter when it is created**

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.

**To set the conditions when creating an alerter**

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.

**To set the formatting when creating an alerter**

You can 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 Alerters 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.

**To 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**.

**To 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**.

**To 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 Desktop Intelligence 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**.

### To 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**.
To delete an alerter

1. Click **Alerters** on the **Format** menu.
2. Click the alerter you want to delete.
3. Click **Remove**.
Filtering and Ranking Data

Highlighting data
Customizing Queries on Universes
Overview

This chapter describes how to benefit from the most powerful query features in Desktop Intelligence. 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 Desktop Intelligence 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, Desktop Intelligence 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, Desktop Intelligence 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, Desktop Intelligence 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 Desktop Intelligence end-users.

**Creating, editing and deleting user objects**

Desktop Intelligence 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>

**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*.

9. Click **OK**.
   
   The user object you have created appears in the User Objects dialog box. The next time you build or edit a query on the universe, the user object you have created will appear in the User Objects class.

   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. Click the **Formula** tab.
2. 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 appears.

3. In the Function[FunctionName] dialog box, type an argument in each string 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.

4. 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.

5. Click OK, then click Test to check the syntax of the formula.

To edit a user object using the Universes command

Editing a user object enables you to change the object's name, type, and definition (formula).

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 using the Universes command

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**.

### To create 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:
To benefit from complex conditions: 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.

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 &lt;select an operator&gt; 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 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.

To apply 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>Double-click...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is equal to one given value, for example a particular year</td>
<td>Equal to</td>
</tr>
<tr>
<td>Is different from one given value, for example &quot;countries, not including France&quot;</td>
<td>Different from</td>
</tr>
<tr>
<td>Is greater than a given value, for example revenue over $100,000</td>
<td>Greater than</td>
</tr>
<tr>
<td>Is greater than or equal to a given value, for example customers who are 60 or older</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>Is lower than a given value, for example revenue under $100,000</td>
<td>Less than</td>
</tr>
<tr>
<td>To obtain data that...</td>
<td>Double-click...</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Is lower than or equal to a given value, for example customers who are 60 or under</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>Falls between two given values, for example weeks between 25 and 36</td>
<td>Between</td>
</tr>
<tr>
<td>Falls outside two given values, for example 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, for example revenue from only two resorts</td>
<td>In list</td>
</tr>
<tr>
<td>Is different from given values, for example non-European customers</td>
<td>Not in list</td>
</tr>
<tr>
<td>Contains empty rows, for example customers who have not paid (i.e., without invoice dates)</td>
<td>Is null</td>
</tr>
<tr>
<td>Does not contain empty rows, for example 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, for example customers whose names begin with the letter S</td>
<td>Matches pattern</td>
</tr>
</tbody>
</table>
To obtain data that...

<table>
<thead>
<tr>
<th>Double-click...</th>
<th>To obtain data that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different from pattern</td>
<td>Does not contain a given letter or pattern of letters, for example customers whose names do not begin with the letter S</td>
</tr>
<tr>
<td>Both</td>
<td>Satisfies two conditions on one object, for example customers who settled an invoice in June and in July</td>
</tr>
<tr>
<td>Except</td>
<td>Excludes a given value, for example customers who stayed at resorts other than French Riviera</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 and tells you what to do next.
<table>
<thead>
<tr>
<th>If you want to compare the condition object with...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Values that you type</strong></td>
<td>Double-click <strong>Type a new constant</strong>, then type the values with a separator (comma, semi-colon, etc.) between each one. The separator to use is defined in the Windows Control Panel (Regional Settings).</td>
</tr>
<tr>
<td><strong>Values that you select from the object's list of values</strong></td>
<td>Double-click <strong>Show list of values</strong>, then hold down the Ctrl key. Click the values you want, then click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>
| **Values that you will select when you run the query** | Double-click **Type a new prompt**, then type your question and press Enter.  
OR  
Double-click **Show list of prompts**, then select a prompt from the dialog box that appears. |
| **Another object (which can be a user object)** | Double-click **Select an object**, then double-click the object in the Classes and Objects box. |
| **Any value returned by another query**        | Double-click **Create a subquery (ANY)**, then build a query in the new query tab that appears. |
If you want to compare the condition object with...

<table>
<thead>
<tr>
<th></th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All values returned by another query</strong></td>
<td>Double-click <strong>Create a subquery (ALL)</strong>, then for more information, refer to <em>Example: To determine when each customer last paid for a product.</em></td>
</tr>
<tr>
<td><strong>The result of a calculation (sum, minimum, maximum, average or count)</strong></td>
<td>Double-click <strong>Calculation</strong>, then follow the screens of the wizard that appears. For more information, refer to <em>Applying a condition with a calculation</em></td>
</tr>
<tr>
<td><strong>The values returned by an existing query</strong></td>
<td>Double-click <strong>Select Query Results</strong>, then follow the procedure described in <em>Using an existing query in a condition</em></td>
</tr>
</tbody>
</table>

### Tips for applying complex conditions

This section provides some tips on getting the most out of Desktop Intelligence 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.

Wildcards are special characters that can denote any single character, or any number of characters. Desktop Intelligence supports the standard wildcard characters, which are:
### Wildcard

<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 Desktop Intelligence 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 depends on restrictions in the target DBMS or on the length of the SQL statement generated by Desktop Intelligence. (Desktop Intelligence can generate SQL of up to 65,536 characters.)

### 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 the condition "Resort Different from 'Bahamas Beach'" 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. Desktop Intelligence checks these customers' records against reservations and determines that no reservations satisfy the condition 'Resort differs from 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, Desktop Intelligence builds two queries:

- All customers.
- Customers who have reservations for Bahamas Beach.

Desktop Intelligence 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*.

**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
A 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.

**To edit complex conditions**

You can edit a complex condition by changing its object, operator and/or operand.

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.

**To delete 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**

Desktop Intelligence 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. Desktop Intelligence 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.
   • To obtain a single result row, click **Globally**, then click **Next**.
   • To obtain several result rows, click **By one or more objects**, select the objects from the list, then click **Next**.

   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.
   • To make a calculation independently of your objects, click **independently of your objects**, then click **Next**.
   • To make a calculation for each value of one or more of your objects, which allows you to limit the calculation to particular objects, click **For each value of one or more objects**, select the objects from the list, then click **Next**.

   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.
   • To compare the object with at least one value, which allows you to limit the values compared with the object, click **At least one value**, then click **Finish**.
   • To compare the object with all values, click **All values**, then click **Finish**.
The Query Panel reappears with the query defined with a condition on a calculation.

10. Click Run.

**Example: To determine when each customer last paid 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.

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.

**Examining the calculation in detail**

Let's look in detail at the calculation you have just created to make what is happening clear.
When you use a calculation, Desktop Intelligence builds SQL that contains a subquery. A subquery is an inner query. The database that receives the SQL generated by Desktop Intelligence 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 that you can create subqueries explicitly. See *Applying a condition with a subquery*.

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

Here is the SQL generated by Desktop Intelligence from the calculation in the example:

```sql
SELECT
    Customer.last_name,
    Sales.invoice_date
FROM
    Customer,
    Sales
WHERE
    ( Customer.cust_id=Sales.cust_id)
AND ( Sales.invoice_date = ALL
    ( SELECT
        max( Sales.invoice_date )
    FROM
        Sales,
        Customer SubAlias__4
    WHERE
        SubAlias__4.cust_id=Sales.cust_id
        AND SubAlias__4.last_name = Customer.last_name
    )
)
```

- invoice_date is the object used in the calculation, so the SQL joins the two queries using the Sales.invoice_date field.
- The calculation level is global, so the sub-query has no grouping.
- The queries are synchronized via the Customer object so the SQL creates a correlated subquery on Customer.last_name.
- 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**

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.
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: 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). Desktop Intelligence 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.

6. Click Run.

   The name and address of the customer who made the earliest reservation appears in the report:

**To delete a subquery**

A subquery is generated by a condition containing the Create a subquery (ALL) operand or the 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 Delete on the shortcut menu.

Subqueries and calculations

The example *Example: To determine when each customer last paid for a product* 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.

**Example: To discover 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 **Equal to** operator.
4. Double-click the **Calculation** operand.
   
   The Complex Condition wizard appears.
5. Select the Reservation Date object, then click **Begin**.
6. Select the Minimum function, then click **Next**.
   
   The Minimum returns the earliest (smallest) reservation date.
7. Select **Globally**, then click **Next**.
8. Select **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 **All Values**, then click **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 *Example: To determine when each customer last paid for a product* 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 is that Desktop Intelligence 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.

To 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.

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.

**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.

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 and vice versa</td>
<td>Double-click the AND or OR operator.</td>
</tr>
<tr>
<td>Indent or outdent conditions within a group of conditions</td>
<td>Click the AND or OR operator with your right mouse button.</td>
</tr>
<tr>
<td></td>
<td>Click Shift right or Shift left 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**

Desktop Intelligence automatically links multiple conditions using operators in the Conditions box in the Query Panel:

- AND specifies a result that is true for both conditions.
- OR specifies a result that is true for either the first or the second condition.

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*.

**Order of precedence**

When you have a group of conditions, Desktop Intelligence objects evaluates them in order. For example, in the following list of conditions, Desktop Intelligence 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 is $\geq 1000000$.

Desktop Intelligence shows this order of precedence by indenting conditions that are evaluated first. and returns the following list of stores that satisfy these conditions:

- Austin
- Chicago 33rd
- Houston Leighton
- Los Angeles
- New York Magnolia
- New York Sundance
- San Francisco
- Washington Tolbooth

However, the following group of conditions:

Sales revenue Greater than or equal to 2000000

OR

Quantity sold Greater than or equal to 10000 / Margin Greater than or equal to 100000

returns the following list of stores:

- Austin
- Chicago 33rd
- Colorado Springs
- Houston 5th
- Houston Leighton
- Los Angeles
New York Magnolia
New York Sundance
San Francisco
Washington Tolbooth

because Desktop Intelligence 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: To determine 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: Resort=Bahamas Beach and Year = 2001 or 2002.

1. Include the Customer object in a query on the Island Resorts Marketing universe.
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.
   The conditions are linked by an AND operator.
6. Double-click the AND that links Year 2001 with Year 2002.
   Desktop Intelligence changes AND to OR, and indents the group of conditions.
7. Click Run.

To delete groups of conditions

1. Click the operator (AND or OR) that links the group of conditions that you want to delete.
2. Right-click.
3. Click Delete on the shortcut menu.

Building combined queries

Desktop Intelligence 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.

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, right-click the Query 2 tab, then click the operator you want on the shortcut menu:
<table>
<thead>
<tr>
<th>Use...</th>
<th>To...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UNION</strong></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, Desktop Intelligence 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.</td>
</tr>
<tr>
<td><strong>INTERSECT</strong></td>
<td>Obtain data common to two sets of results, such as customers from a given region and a given age group. The example, <em>Example: To determine which customers bought a given product in a given time period</em>, illustrates the use of INTERSECT.</td>
</tr>
</tbody>
</table>
To...Use...

To...

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.

<table>
<thead>
<tr>
<th>Use...</th>
<th>To...</th>
</tr>
</thead>
<tbody>
<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.</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 right-clicking its tab, then by clicking the **Delete** command in the shortcut menu.

**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: Determining 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 operator; in this case, you would obtain customers from either 2001 or 2002, or possibly both.

- You cannot obtain the list of paying customers from two years by linking two conditions by AND (and) because this will return no data.

- You cannot obtain the list of paying customers from two years by linking two conditions by OR (or) 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.

**Building a combined query that returns customers who match two criteria**

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 right-click the Query 2 tab.
   - The operators you can use to combine the queries appear on a shortcut menu.
8. Click Intersect.
The INTERSECT symbol appears on the Query 2 tab.

9. Click Run.

A list of customers who bought products in years appears in the report.

Using SQL from Desktop Intelligence queries

When you build a query in the Query Panel, Desktop Intelligence writes the query's corresponding SQL. If you know SQL and want to find out how Desktop Intelligence resolves a query, you can view the SQL script.

You can also reuse the SQL that Desktop Intelligence generates by saving it to a file. And because Desktop Intelligence 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 View SQL on the Query Panel toolbar.

The query's SQL script appears in the SQL Viewer dialog box.

Note: It is possible that Desktop Intelligence 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 Desktop Intelligence 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 if you want to keep the changes you have made.

This option automatically parses the script when you click OK.

Note that 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.
6. Click Save to save the edited script to a file.
7. Click OK to return to the Query Panel.
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, Desktop Intelligence 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.

There is no direct Desktop Intelligence restriction on the number of values in a list of values. Nevertheless the list length might be restricted indirectly by limitations in the database or the maximum length of an SQL query that Desktop Intelligence can generate. (Desktop Intelligence cannot generate queries of more than 65,536 characters.)

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 Desktop Intelligence.

The first time you view an object's list of values, Desktop Intelligence 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 Desktop Intelligence

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.

---

**Editing lists of values**

The universe designer decides whether a list of values is editable in Desktop Intelligence. 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: To show 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.

1. Click **Universes** on the **Tools** menu.
   
   The Universes dialog box appears.

2. Select the Island Resorts Marketing universe and click **List of Values**.
3. Navigate to the City object beneath the Customer class.
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.

### To assign personal data from a text file

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 Desktop Intelligence, 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**.

9. Click **OK**.

**To assign personal data from an Excel file**

1. Create an Excel file containing the list of values.

2. In Desktop Intelligence, 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 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 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 **Range Name** list.
9. Select **First Row Contains Column Names** if the range contains the column name.
10. Click **Run**.

**To assign personal data from a dBase file**

1. Click **Universes** on the **Tools** menu, select the Island Resorts Marketing 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 list.
5. Click **Browse** and select the dBase (.dbf) file that contains the list of values.
6. Click **Run**.

**To display, refresh and purge lists of values**

Desktop Intelligence lets you display, refresh and purge lists of values at any time.

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.
To display, refresh and purge lists of values

<table>
<thead>
<tr>
<th>Button</th>
<th>Action</th>
</tr>
</thead>
</table>
| Display | Desktop Intelligence displays the values in the List of Values of Object Name dialog box.  
To view the list in table or hierarchy format, click Tabular View or Hierarchical View 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:  
• In **Tabular View**, the columns appear next to each other.  
• In **Hierarchical View**, 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. |
| Refresh | Desktop Intelligence runs the query for the list of values, and a refreshed list appears. |
| Purge | Desktop Intelligence empties the .lov file corresponding to the list of values. |

5. Click **OK** to close the dialog box.  
You can populate a purged list of values by clicking Refresh.
To display, refresh and purge lists of values
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 Desktop Intelligence.

The following chapters give further information on using calculations in Desktop Intelligence:

• "Formulas, Local Variables and Functions" explains how you can use the power of the Desktop Intelligence formula editor to write your own calculations.

• "Calculation Contexts and Extended Syntax" explains the more sophisticated aspects of writing formulas and gives some background about how the Desktop Intelligence calculation engine works.

• "Calculation Troubleshooting" 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.

Calculations

Desktop Intelligence 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."

To add simple calculations to reports

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.
The result of the calculation is inserted in a new cell in the table or crosstab. 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.

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. They are, in order: Sum, Percentage, Count, Variance, and Variance percentage.

To delete calculations from a table or a 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 that 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

Calculation examples

The following section contains several examples of making calculations using the Calculations menu.

To display total revenue and subtotals

You want to calculate and display the total revenue in a table.
1. Click in the Revenue column.
2. Click Sum on the Calculations toolbar.
   • Desktop Intelligence displays the total revenue in the footer at the end of the table.
   • The sum button on the Calculations toolbar is dimmed.

To insert a break to display subtotals for each resort

You now decide you want to insert a break on this table to display subtotals for each resort.
1. 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.
2. Click in the Revenue column.
3. Click Sum once to remove the existing calculation.
4. Click Sum again to insert the subtotals.
   Desktop Intelligence displays a subtotal for each resort and a total at the end of the table.
Example: 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, Desktop Intelligence adds an extra column that shows each row as a percentage of the total.

Example: To calculate 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.

1. Select Q2 with the mouse, then, holding down the Ctrl key, select Q1.
2. Click Variance Percentage on the Calculations toolbar.
   Desktop Intelligence 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.
   Desktop Intelligence displays the difference in revenue between Q1 and Q2 for each resort as a percentage in a new cell called Q4-Q3.

Note: You can also use the Shift key to select columns and rows. If you use the Shift key, Desktop Intelligence 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, Desktop Intelligence will calculate Q1-Q2.

To reuse 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. Desktop Intelligence keeps the formula with the cell.
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 that 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."

**Example: To display 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.

1. Select the Sum cell at the bottom of the report and drag it to the top of the report.
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.

**Converting to and from Euros**

The following section describes how Desktop Intelligence 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.
Desktop Intelligence 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 Desktop Intelligence, 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. Desktop Intelligence 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 Desktop Intelligence

Desktop Intelligence 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 Desktop Intelligence 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 Desktop Intelligence.

To convert to euros

You can convert an amount from one of the eleven EMU currencies into
euros using one easy mouse click.

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 Desktop Intelligence recognizes the currency, it carries out the
     conversion and displays the result.
   • If Desktop Intelligence 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 Desktop Intelligence recognize the currency of the selected data?

Desktop Intelligence 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 Desktop Intelligence as German marks.

If the data is not formatted as a currency Desktop Intelligence 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.

To convert from euros

You can convert a figure from euros into one of the eleven EMU currencies using one easy mouse click.

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

Desktop Intelligence 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.

To view conversion rates

- 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:

<table>
<thead>
<tr>
<th>Currency</th>
<th>Shows the official ISO (International Standards Organization) label for the currency.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate</td>
<td>Shows the official 6-digit conversion rate between the currency and the euro.</td>
</tr>
<tr>
<td>Decimal</td>
<td>Shows the number of decimal digits used by each currency.</td>
</tr>
</tbody>
</table>
To edit 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. It displays the information that Desktop Intelligence 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.

To add 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: To add 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.

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 Desktop Intelligence 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 Desktop Intelligence formula.

The following Desktop Intelligence formula converts 100 DEM to 2063 BEF.

```
EuroConvertFrom(EuroConvertTo(100, "DEM", 3), "BEF", 0)
```
This chapter is designed to help you use the powerful calculation capabilities delivered with Desktop Intelligence.

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 Desktop Intelligence reports.

What's in this chapter

The chapter provides information on how Desktop Intelligence performs calculations in reports. It describes the concepts behind the Desktop Intelligence 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 "#COMPUTATION in cumulative aggregations."
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 Desktop Intelligence, 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 Desktop Intelligence demo universe, Revenue is the calculation of number of items sold multiplied by item price.

When you run the query, Desktop Intelligence retrieves the data corresponding to the dimensions and measures you selected. Desktop Intelligence 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.

Desktop Intelligence dynamically calculates the Revenue measure according to the dimensions in the table. If you remove the Year column, Desktop Intelligence returns revenue per region.

**Understanding input and output contexts**

Desktop Intelligence 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:
To understand how Desktop Intelligence 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>
</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>

(Desktop Intelligence also supports reset contexts, which are used in cumulative aggregations such as running totals. To find out about reset contexts, refer to Reset contexts).

This section explains how Desktop Intelligence defines input and output contexts in different parts of a report. This information is important if you

- want to understand the different results Desktop Intelligence 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 Desktop Intelligence defines input and output contexts

To understand how Desktop Intelligence 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:
Understanding input and output contexts

<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 Desktop Intelligence uses to obtain the result of each calculation.
<table>
<thead>
<tr>
<th></th>
<th>The calculation returns...</th>
<th>Because the body (input) context is...</th>
<th>And the local (output) context is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>Total revenue for the report</td>
<td>All dimensions - including any that are not displayed in the report</td>
<td>The same as the body.</td>
</tr>
<tr>
<td>b.</td>
<td>Total revenue for 2000</td>
<td>Year, the section master, (the calculation is placed at the section level)</td>
<td>The same as the body.</td>
</tr>
<tr>
<td>c.</td>
<td>Revenue per city per quarter per year</td>
<td>Year, Quarter, City - the dimensions in the section and table</td>
<td>The same as the body.</td>
</tr>
<tr>
<td>d.</td>
<td>Average revenue per city per quarter per year</td>
<td>Year, Quarter, City - the dimensions in the section and table</td>
<td>Quarter (the calculation is placed in the footer of the break on Quarter.)</td>
</tr>
</tbody>
</table>

### 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:

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

\[=\text{RunningSum}(\text{Revenue} \text{ In Body}) \text{ In <Country>}\]

where Body is the input context, and Country, the output context.

You can use your knowledge of extended syntax to:

- understand the results Desktop Intelligence 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.*
**Viewing the extended syntax of a formula**

Desktop Intelligence 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

**Viewing extended syntax by 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.
   
   The extended syntax of the formula appears in a tooltip.

**Viewing extended syntax by 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.
4. Click **Cancel** to close the dialog box.
Introduction to contexts and extended syntax

Using your understanding of input and output contexts
Using extended syntax for advanced calculations
When you insert a simple aggregation in a report, for example to calculate average revenue, Desktop Intelligence writes a simple formula, in this case

\[ \text{Average}(<\text{Revenue}>) \]

By default, Desktop Intelligence evaluates the formula in its default context, as explained in the previous section, *Understanding input and output contexts*.

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, Desktop Intelligence 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?

Desktop Intelligence 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, Desktop Intelligence counts the cities one by one, using the following formula:

\[ \text{Count}(<\text{City}>) \]

By default, Desktop Intelligence makes the calculation based on the dimensions in the table (Region, City). There's only one city per city, so Desktop Intelligence returns 1 every time.

Extended syntax enables you to specify that you want Desktop Intelligence 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.
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 Desktop Intelligence.

Desktop Intelligence 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 earlier example (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, Desktop Intelligence 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. Desktop Intelligence allows you to do this, because even though you have removed City from the report display, it is still available in the document.
The formula is as follows:

=Min(<Revenue> In (<Region>,<City>))

---

**How to define input and output contexts**

When you insert an aggregation such as Sum, Desktop Intelligence writes a simple formula, for example

=Sum(<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:

=AggregateFunction(<measure> In <input context>) In <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:

=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 this example: Defining calculation contexts with extended syntax.

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

=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.

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 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,

```
<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:

`;<Year>,<Region>`

7. Type a closing parenthesis at the end of the formula, then press Enter.

### 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:

<table>
<thead>
<tr>
<th>Region</th>
<th>FY1998</th>
<th>FY1999</th>
<th>FY2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid West</td>
<td>$128,362.00</td>
<td>$150,666.00</td>
<td>$162,566.00</td>
</tr>
<tr>
<td>South</td>
<td>$128,330.00</td>
<td>$135,580.00</td>
<td>$136,989.00</td>
</tr>
<tr>
<td>West</td>
<td>$96,360.00</td>
<td>$106,210.00</td>
<td>$120,050.00</td>
</tr>
</tbody>
</table>

Now, to calculate running total revenue, you insert a break on Year, then change Revenue to

`=RunningSum(<Revenue>)`

Here’s the result:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid West</td>
<td>$128,362</td>
<td>$128,362</td>
<td>$150,666</td>
<td>$279,028</td>
</tr>
<tr>
<td>South</td>
<td>$128,330</td>
<td>$407,358</td>
<td>$136,580</td>
<td>$542,938</td>
</tr>
<tr>
<td>West</td>
<td>$96,380</td>
<td>$639,318</td>
<td>$108,210</td>
<td>$747,528</td>
</tr>
</tbody>
</table>

Notice that Desktop Intelligence 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

`=RunningSum(<Revenue>;<Year>)`

and the result is as shown:
Users of previous versions of Desktop Intelligence 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

\[=\text{RunningSum}(\text{<Revenue}>; \text{Col } \text{<Year>})\]

and

\[=\text{RunningSum}(\text{<Revenue}>; \text{<Year>})\]

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:

\[=\text{RunningAggregateFunction}(\text{<measure> In <input context>;<reset context>}) \text{ In <output context>}\]

**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:

\[=\text{Min}(\text{<Revenue>} \text{ In } (\text{<Region>,<City>}) \text{ In <Region>})\]

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:
### Operator and Definition

<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:

```plaintext
=Max(<Revenue> ForEach <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:

```plaintext
=Max(<Revenue>) ForAll <City>
```
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, Desktop Intelligence 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,

\[ \text{Max}(<\text{Revenue}>) \text{ ForAll } <\text{City}> \]
\[ \text{Max}(<\text{Revenue}>) \text{ In } <\text{Region}> \]

where the local context is Region and City, obtain the same result.

Now try ForEach <City> vs. In (<Region>,<City>)

In the example entitled Defining calculation contexts with extended syntax, you used

\[ \text{Min}(<\text{Revenue}>) \text{ In } (<\text{Region}>,<\text{City}>) \]

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

\[ =\text{Min}(\text{Revenue} \text{ ForEach City}) \]

### 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:

\[ =\text{Rank}(\text{dimension} ,\text{measure}) \]

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:

\[ =\text{Rank}(\text{Quarter} ,\text{Revenue}) \]

But what if you put a break or a section on country? The result is that the quarters are ranked for each country separately.

This is because, by default, Desktop Intelligence includes country in the calculation context. Since there are two values for Country, Desktop Intelligence makes a calculation for each value.

You can modify the calculation context by using the ForAll context operator. This tells Desktop Intelligence to ignore the break (or section) on Country. The syntax is:

\[ =\text{Rank}(\text{Quarter} ,\text{Revenue}) \text{ ForAll Country} \]

The result is:
Desktop Intelligence 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.

### To remove a break from one context and keep it in another

1. Create a new measure variable using the following syntax:

   ```plaintext
   =<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:

   ```plaintext
   =Rank(<Quarter> ,<Revenue ForEach Country>) ForAll <Country>
   ```

   The result is:
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>Country</th>
<th>Quarter</th>
<th>Revenue</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Q1</td>
<td>$208,565.00</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$242,165.00</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$226,125.00</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$158,565.00</td>
<td>8</td>
</tr>
<tr>
<td>US</td>
<td>Q1</td>
<td>$632,051.00</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$579,652.00</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$658,572.00</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$630,829.00</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>Country</th>
<th>Rank</th>
<th>Quarter</th>
<th>Rank</th>
<th>French Riviera</th>
<th>Rank</th>
<th>Hawaiian Club</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahamas Beach</td>
<td>3</td>
<td>Q1</td>
<td>3</td>
<td>$224,881</td>
<td>3</td>
<td>$357,170</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Q2</td>
<td>2</td>
<td>$237,672</td>
<td>2</td>
<td>$341,780</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Q3</td>
<td>1</td>
<td>$263,422</td>
<td>1</td>
<td>$395,150</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Q4</td>
<td>4</td>
<td>$245,239</td>
<td>4</td>
<td>$335,560</td>
<td>4</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.

Desktop Intelligence 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 Desktop Intelligence assigns keywords to formulas, and
- an explanation of how report filters behave when you use keywords.

<table>
<thead>
<tr>
<th></th>
<th>Bahamas Beach</th>
<th>Rank</th>
<th>French Riviera</th>
<th>Rank</th>
<th>Hawaiian Club</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>$224,981</td>
<td>4</td>
<td>$208,565</td>
<td>3</td>
<td>$367,170</td>
<td>3</td>
</tr>
<tr>
<td>Q2</td>
<td>$237,872</td>
<td>3</td>
<td>$242,165</td>
<td>1</td>
<td>$341,780</td>
<td>4</td>
</tr>
<tr>
<td>Q3</td>
<td>$263,422</td>
<td>1</td>
<td>$226,125</td>
<td>2</td>
<td>$395,150</td>
<td>1</td>
</tr>
<tr>
<td>Q4</td>
<td>$245,269</td>
<td>2</td>
<td>$158,565</td>
<td>4</td>
<td>$365,660</td>
<td>2</td>
</tr>
</tbody>
</table>
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></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:

\[ \text{Sum(<Revenue>) In Report} \]

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.

---

**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,

\[ \text{=Sum(<Revenue>)} \]

in the Formula Bar.

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 keyword (Block, Body, Report, or CurrentPage), then press **Enter**.

**How Desktop Intelligence uses keywords**

You use keywords in extended syntax when you need to define contexts in aggregate formulas. Desktop Intelligence defines contexts in all aggregate
formulas, however simple, and often uses keywords to do so. The following example illustrates this behavior.

**Example: The formula Desktop Intelligence 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.

Desktop Intelligence, however, writes a full formula in which it specifies input and output contexts. The formula is

\[ \text{=Sum(<Revenue> In Body) In (<Year>)} \]

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 Desktop Intelligence uses. But what if you don't understand the numbers Desktop Intelligence 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 Desktop Intelligence 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**

Desktop Intelligence still applies filters when you use keywords in aggregations, but you can force Desktop Intelligence to ignore them.
How to force Desktop Intelligence 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

\[ =\text{NoFilter(formula)} \]

an example being

\[ =\text{NoFilter(Sum(<Revenue>) In Report)} \]

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>Document</td>
<td>A Desktop Intelligence file (extension .rep.).</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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.</td>
</tr>
<tr>
<td></td>
<td>You can use extended syntax to define your own input and output contexts.</td>
</tr>
<tr>
<td></td>
<td>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>
<tr>
<td>Measure</td>
<td>Qualification of an object, variable or formula that returns numeric data, such as revenue.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Because documents can contain data from different sources in Desktop Intelligence, any given document can contain multiple microcycles.</td>
</tr>
</tbody>
</table>
**Term** | **Definition**
---|---
Report | 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.
Variable | A named formula. Desktop Intelligence always stores query results as variables.

### Calculation contexts

A calculation context consists of one or more dimensions that determine the input and output of aggregate calculations.

<table>
<thead>
<tr>
<th>Context</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>Dimension or dimensions that govern the part of the report where the calculation is inserted (e.g., a master variable in a section).</td>
</tr>
<tr>
<td>Context</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Body</td>
<td>Dimension or dimensions that are present in the part of the report (e.g., a block) where the calculation is inserted.</td>
</tr>
<tr>
<td>Input</td>
<td>One or more dimensions that go into an aggregate calculation. By default, the input context for a calculation is defined by the dimension(s) in the body. You can define your own input context using extended syntax.</td>
</tr>
<tr>
<td>Output</td>
<td>One or more dimensions that determine the result of an aggregate calculation. By default, the output context is defined by the dimension(s) in the local context. You can define your own output context using extended syntax.</td>
</tr>
<tr>
<td>Reset</td>
<td>One or more dimensions that reset a cumulative aggregation such as running total to 0 when the value of the dimension(s) changes. You specify reset contexts using extended syntax.</td>
</tr>
</tbody>
</table>

**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>Block</td>
<td>Corresponds to the dimensions in the current section. Can only be used in a formula placed inside a block.</td>
</tr>
<tr>
<td>Body</td>
<td>When the formula is placed...</td>
</tr>
<tr>
<td></td>
<td>corresponds to...</td>
</tr>
<tr>
<td></td>
<td>Outside a block</td>
</tr>
<tr>
<td></td>
<td>Inside a block</td>
</tr>
<tr>
<td></td>
<td>Anywhere in the report</td>
</tr>
<tr>
<td>CurrentPage</td>
<td>Inside the current page</td>
</tr>
</tbody>
</table>
Calculation Troubleshooting
Overview

This chapter describes solutions for computation errors that might occur in Desktop Intelligence. A computation error always starts with # and appears 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 #MULTIVALE errors in your reports, you are strongly advised to first read "Calculation Contexts and Extended Syntax," because these errors often result from the use of extended syntax.

#COMPUTATION

This section provides descriptions, examples and solutions for the following errors in Desktop Intelligence:

- #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 Desktop Intelligence extended syntax, which is described in "Using extended syntax for advanced calculations."
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."

#COMPUTATION occurs in cumulative aggregations for the following reason:

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:

```plaintext
=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:
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:

\[=\text{RunningSum}(\text{<Revenue>};\text{<Year>})\]

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 (In, ForEach or ForAll) on the right of the formula.
If you do not define an output context in the formula, Desktop Intelligence makes the calculation using the local context. For further information, refer to "Understanding input and output contexts."

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;;Region) In (Region,&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 (Region,&lt;City&gt;);Region)</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;;Region)</td>
<td>Region must be present in the local context.</td>
</tr>
</tbody>
</table>
Then... When the reset context is Region and...

<table>
<thead>
<tr>
<th>When the reset context is Region and...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both the input and the output contexts are defined, e.g.,</td>
<td>Region must also be specified in the output context.</td>
</tr>
<tr>
<td>( =\text{RunningSum(&lt;Revenue&gt; In (&lt;Region&gt;,&lt;City&gt;);&lt;Region&gt;) In (&lt;City&gt;;&lt;Region&gt;)} )</td>
<td></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

\[
\begin{align*}
\text{= If (}<\text{Year}>="\text{FY1998}\text{") Then <Revenue}>\end{align*}
\]

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:
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:

\[=\text{(If (Year="FY1998") Then Revenue)} \text{ ForEach Year}\]

Finally, you add Max (or Min) to return the single value you need:

\[=\text{Max((If (Year="FY95") Then Revenue) ForEach Year)}\]

Desktop Intelligence now displays Revenue for 1998 in the break footer.

---

**Solution for #COMPUTATION in a non-aggregate formula**

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.

---

**#MULTIVALUE**

This section provides descriptions, examples and solutions for #MULTIVALUE in Desktop Intelligence:

- **#MULTIVALUE in aggregations** such as maximum revenue per year
- **#MULTIVALUE in break headers and footers**
#MULTIVALE 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 #MULTIVALE occurs in aggregations, with an example, and offers a solution.

Note:
The information presented here requires that you understand Desktop Intelligence extended syntax, which is described in "Calculation Contexts and Extended Syntax."

Description of #MULTIVALE in an aggregation

#MULTIVALE occurs in aggregations because

Example: #MULTIVALE 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 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>

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

=\langle \text{Revenue} \rangle \text{ In } \langle \text{Year} \rangle

returns #MULTIVALE 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, Desktop Intelligence 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>

Desktop Intelligence 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 #MULTIVALE in an aggregation

To avoid #MULTIVALE 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 #MULTIVALE:

   =Min(<Revenue>) In(<Year>)

   • Does the output context contain more dimensions than the local context?
• For example

\[ \text{Min}(<\text{Revenue}>) \text{ In}(<\text{Region}>,<\text{City}>) \]

• 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:

<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: #MULTIVALE in a break footer

The table below shows the running total revenue for two customers. When you insert Age in the break footer, Desktop Intelligence returns #MULTIVALE.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Resort</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arai</td>
<td>Bahamas Beach</td>
<td>$8,036.00</td>
</tr>
<tr>
<td>#MULTIVALE</td>
<td></td>
<td>$8,036.00</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>$141,584.00</td>
</tr>
<tr>
<td></td>
<td>French Riviera</td>
<td>$141,300.00</td>
</tr>
<tr>
<td></td>
<td>Hawaiian Club</td>
<td>$158,710.00</td>
</tr>
<tr>
<td>#MULTIVALE</td>
<td></td>
<td>$441,594.00</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 #MULTIVALE in a break header or footer

You fix this problem by including the variable from the header or footer in the break definition.

1. Click inside the table or crosstab containing the break, then select click **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 previous example), then click **OK**.

5. Click **OK** in the Breaks dialog box.

   Desktop Intelligence displays the variable in the break footer.
A report section displays data in a master cell and in a block or blocks. If you use two variables at the section level, #MULTIVALUE can occur. Here's an example.

### Example: #MULTIVALUE 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:

```
Baker
```

<table>
<thead>
<tr>
<th>Invoice Date</th>
<th>Revenue</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/20/1998</td>
<td>9,536.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>3/21/1998</td>
<td>23,540.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>6/19/1998</td>
<td>25,160.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>9/19/1998</td>
<td>35,412.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>12/19/1998</td>
<td>34,714.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>3/20/1999</td>
<td>36,654.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>6/19/1999</td>
<td>38,614.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>9/19/1999</td>
<td>36,654.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>12/19/1999</td>
<td>38,744.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>3/20/2000</td>
<td>40,068.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>6/19/2000</td>
<td>41,556.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>9/19/2000</td>
<td>38,916.00</td>
<td>2690 Grant Avenue</td>
</tr>
<tr>
<td>12/19/2000</td>
<td>42,026.00</td>
<td>2690 Grant Avenue</td>
</tr>
</tbody>
</table>

Why? Because by default, there is only one master variable per section.

---

**Solution #1 for #MULTIVALUE at the section level**

The way around this problem is to turn the variable that returns the error into a measure.

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>
If the Edit button is grayed

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**.

If the Edit button is available

Click **Edit**.

The Variable Editor appears.

In the Qualification box, click **Measure**, then 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 #MULTIVALE at the section level**

You can also fix #MULTIVALE at the section level by applying the Min or Max function, which forces Desktop Intelligence 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 #MULTIVALE 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 #MULTIVALE.

2. In the Formula Bar, type Min or Max after (=), then add parentheses, like this =Min(<Address>)
3. Press Enter.

The cell is not wide enough to display the data it contains in full.

Solution: double-click the cell's right border. Desktop Intelligence widens the cell to autofit the data.

#ALERTER

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.
In Desktop Intelligence, 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, Desktop Intelligence automatically replaces the variables in the template with the variables in the report. In some cases, Desktop Intelligence cannot match the variable dictionaries and returns #DICT.ERROR.

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.

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.

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 is checked, then click **OK**.

8. Click **OK** in the Apply Template dialog box.

9. Desktop Intelligence applies the template to the report.

---

**#DIV/0**

Occurs when a formula performs a division by 0. For example, the formula

\[
=\frac{\text{Revenue}}{\text{Quantity Sold}}
\]

returns 20 if Revenue is 100 and Quantity Sold is 5. But if Quantity Sold is 0, then the result is #DIV/0.

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 fix this problem:**

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

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.

To fix this problem

You need to break down the formula into its component parts in order to find which part contains the error.

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.

#IERR in a formula combining measures and dimensions

Desktop Intelligence 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>

Desktop Intelligence 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 "Calculations."

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:

=Sum((If(<Customer>="Prince") Then <Revenue>) ForEach <Customer>)

#IERR in an aggregation containing a complex formula

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**

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 #IERR in an aggregation containing a complex formula for more information.

#OVERFLOW

The calculation returns a number that is too big for Desktop Intelligence to compute. The maximum is 17 +/- 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 Desktop Intelligence online help on functions includes maximum values where appropriate.

#SYNTAX

#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:
If the variable you need is... | Then...
---|---
Available as an object in the universe you are using, | Edit the query (Edit Data Provider command, Data menu), add the object you need, then click Run.
A local variable that you have defined in the report, | Create it in the Formula Editor (Variables command, Data menu, Add button).

#UNKNOWN

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 Desktop Intelligence does not always remove it from the report, More specifically:

<table>
<thead>
<tr>
<th>If the variable is displayed in...</th>
<th>Then Desktop Intelligence...</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, Desktop Intelligence returns #SYNTAX.
To fix this problem

You have to add the missing object to the query.

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.

3. The object appears in the Result Objects box.

4. Click **Run**.

   Desktop Intelligence replaces #UNKNOWN with the variable corresponding to the object you added.

Note that 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, #MULTIVALUE and other errors.

<table>
<thead>
<tr>
<th>Tip</th>
<th>What to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>All formulas must begin with =</td>
<td>Type = before your formula.</td>
</tr>
<tr>
<td>If you forget =, Desktop Intelligence displays the formula as text!</td>
<td></td>
</tr>
<tr>
<td>Tip</td>
<td>What to do</td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Click the cell containing the formula. Rest your cursor over the Formula Bar. The extended formula appears in a tooltip.</td>
</tr>
<tr>
<td>- or -</td>
<td>Click the cell containing the formula. Click <strong>Define as Variable</strong> on the <strong>Data</strong> menu. The &quot;Define As Variable&quot; dialog box appears. Select <strong>Evaluate the formula in its context</strong>. The extended formula appears in the dialog box.</td>
</tr>
<tr>
<td>Click the table containing the dimension you want to hide.</td>
<td>Click <strong>Table</strong> on the <strong>Format</strong> menu. In the Pivot tab, click the dimension. Click <strong>Hide</strong>, then click <strong>OK</strong>.</td>
</tr>
<tr>
<td>The best way to fix #MULTIVALUE is to include the variable returning the error in the current break.</td>
<td>Click <strong>Break</strong> on the <strong>Format</strong> menu. For more information, refer to <strong>Solution for #MULTIVALUE in a break header or footer</strong>.</td>
</tr>
<tr>
<td>Tip</td>
<td>What to do</td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
</tr>
<tr>
<td><strong>If you get #MULTIVALUE when you insert a variable in a column header of a table, try fixing it by applying a sort.</strong></td>
<td>Click the cell containing #MULTIVALUE. Click <strong>Sort</strong> on the <strong>Insert</strong> menu.</td>
</tr>
<tr>
<td><strong>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, Desktop Intelligence tells you so and highlights the anomaly in the formula itself. This helps you fix the specific problem.</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>You can edit formulas directly in the report, without using the Formula Bar or the Formula Editor.</strong></td>
<td>Double-click the cell containing the formula you want to edit. Edit the formula in the cell, then press Enter when you're done.</td>
</tr>
<tr>
<td><strong>You can display the Formula Editor from the Formula Bar.</strong></td>
<td>Click the Formula Editor button: ![Formula Editor Button]</td>
</tr>
<tr>
<td><strong>You can view and edit all the formulas displayed in a report by switching to Structure view.</strong></td>
<td>Click <strong>Structure</strong> on the <strong>View</strong> menu.</td>
</tr>
<tr>
<td><strong>Desktop Intelligence lists all the formulas you create in a document in the Data tab of the Report Manager.</strong></td>
<td>Click <strong>Report Manager</strong> on the <strong>View</strong> menu. Click the <strong>Data</strong> tab and expand the Formulas folder.</td>
</tr>
<tr>
<td>Tip</td>
<td>What to do</td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
</tr>
<tr>
<td>Desktop Intelligence also lists formulas in the Variables dialog box. In this dialog box, you can view, edit or insert any formula you have created.</td>
<td>Click <strong>Variables</strong> on the <strong>Data</strong> menu. Double-click the Formulas folder. To edit a formula, select it then click <strong>Edit</strong>. To insert a formula, select it then click <strong>Insert</strong> or <strong>Replace</strong>.</td>
</tr>
</tbody>
</table>
Formulas, Local Variables and Functions
Overview

This chapter gives you an introduction to using Desktop Intelligence formulas, local variables and functions. It explains how to set up your own formulas and variables in Desktop Intelligence reports using Desktop Intelligence syntax, how to use Desktop Intelligence functions, and also includes several examples of business calculations.

"Calculation Contexts and Extended Syntax" describes how to build more powerful calculations by using extended syntax to define calculation contexts. This chapter also has a calculations "Quick reference" that recaps some of the key Desktop Intelligence terms.

"Quick reference" gives solutions to computation errors that may occur in Desktop Intelligence when you are writing or using formulas.

Formulas

When you run a query, Desktop Intelligence 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 Desktop Intelligence formulas.

Why use formulas?

You use formulas to carry out calculations locally in the report to set conditions on filters and data display.

A Desktop Intelligence 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:

=Sum<Sales Revenue>
=Margin/<Sales revenue>*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 Desktop Intelligence 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.

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 Desktop Intelligence 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.

The Formula Bar has five buttons and a text box where the formula is displayed. If you click inside any cell in a report, the contents are displayed in this text box. Depending on the type of cell you clicked in, this could be text, the name of a variable or a formula.

### Displaying the Formula Editor

If you are not familiar with Desktop Intelligence syntax or are writing more complex formulas, the best method is to use the Formula Editor.

### To open the Formula Editor

1. Click inside the cell where you want the formula to appear.
2. If the Formula Bar is open, click **Formula Editor** on the Formula Bar, or else 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 Desktop Intelligence 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.

Calculating a running total

You want to display a running total for the monthly sales revenue.

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.
Desktop Intelligence displays RunningSum in the Formulas box. Notice that the cursor is inside the brackets. This is where Desktop Intelligence 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.

Desktop Intelligence 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 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.
To create a local variable using the Variable Editor

You can create a local variable using the Variable Editor.

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 Desktop Intelligence 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>Allow the variable to be defined in the context where it is inserted.</td>
<td>Keep the formula generic 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>
</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 alert using the variable.

For further information on calculation contexts, see "Calculation Contexts and Extended Syntax."

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

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 display revenue per semester**

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**.
   
   Desktop Intelligence groups Q1 and Q2 together and displays the new name, H1, in the table.

   Desktop Intelligence creates a a new variable and displays it in the Report Manager list. Desktop Intelligence 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.

**To rename a variable**

You can rename the new variable to make its name more meaningful.

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.

**To edit 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.

1. Right-click on the variable in the Report Manager window.
2. Click **Edit Variable** on the shortcut menu.

   The Edit box opens.
   
   • You can rename the variable in the Name of the Variable box.
   
   • You can rename the values directly in the Groups of Values box. Click on the name and type in the new one.
   
   • You can add a new group to the Groups of Values box by clicking **New**.
   
   • You can drag and drop values from one group folder to another.

3. Make the required changes and click **OK**.

**To delete 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**.

**To ungroup 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."

### 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.

### To edit 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.

Desktop Intelligence displays the result of the formula in the cell.

**To edit local variables**

You can edit a local variable directly from the Report Manager.

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

**To delete 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. 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

Desktop Intelligence 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 Desktop Intelligence 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>
</tbody>
</table>
### The folder... | Lists...
---|---
Logical Functions | Functions that return true or false.
Document Functions | Functions that return information about a document.
Data Provider Functions | Functions that return information about a data provider.
Miscellaneous Functions | Functions that cannot be categorized into one of the seven category folders.

## 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 Desktop Intelligence 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.

**To rank cities by quarterly revenue**

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.
   
   Desktop Intelligence adds the Rank function to the Formula window.
   
   In the Formula Editor box, Desktop Intelligence adds parentheses and commas automatically, and at the bottom left, shows the syntax for the selected function. Click Function Help to get more information on the function.
   
   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.
   
   Desktop Intelligence 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. Desktop Intelligence 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.

- 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 Desktop Intelligence functions have equivalents in Microsoft Excel, or equivalent formulas that use different Excel functions to produce the same result. The following table cross-references Desktop Intelligence functions to Excel functions or formulas.

### Aggregate function equivalents

<table>
<thead>
<tr>
<th>Desktop Intelligence</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>Desktop Intelligence</td>
<td>Excel equivalent</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>VarP</td>
<td>VarP</td>
</tr>
</tbody>
</table>

### Numeric function equivalents

<table>
<thead>
<tr>
<th>Desktop Intelligence</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs</td>
<td>Abs</td>
</tr>
<tr>
<td>Ceil</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>
<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>Desktop Intelligence</td>
<td>Excel equivalent</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------</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>
</tbody>
</table>
### Function equivalents in Microsoft Excel

<table>
<thead>
<tr>
<th>Desktop Intelligence</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tan</td>
<td><code>Round(Tan(Value), 2)</code></td>
</tr>
<tr>
<td>Truncate</td>
<td><code>Round(Value, 0)</code></td>
</tr>
</tbody>
</table>

#### Character function equivalents

<table>
<thead>
<tr>
<th>Desktop Intelligence</th>
<th>Excel equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordCap</td>
<td><code>StrConv(String, vbProperCase)</code></td>
</tr>
<tr>
<td>Upper</td>
<td><code>StrConv(String, vbUpperCase)</code></td>
</tr>
<tr>
<td>Trim</td>
<td><code>Trim</code></td>
</tr>
<tr>
<td>Substr</td>
<td><code>Mid</code></td>
</tr>
<tr>
<td>RightTrim</td>
<td><code>RTrim</code></td>
</tr>
<tr>
<td>Right</td>
<td><code>Right</code></td>
</tr>
<tr>
<td>Replace</td>
<td><code>Replace</code></td>
</tr>
<tr>
<td>Pos</td>
<td><code>Instr</code></td>
</tr>
<tr>
<td>Desktop Intelligence</td>
<td>Excel equivalent</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</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>
Date function equivalents

<table>
<thead>
<tr>
<th>Desktop Intelligence</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>

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 Desktop Intelligence formula editor and Desktop Intelligence 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 Desktop Intelligence, you use the Previous() function to set up a rolling average.

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:

   \[ =(<\text{Sales revenue}>+\text{Previous(<Sales revenue>)}+\text{Previous(Previous(<Sales revenue>)))}/3. \]

   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 Desktop Intelligence 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 Desktop Intelligence. 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 Desktop Intelligence 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:

\[
\text{Concatenation(character string, character string)}
\]

To display a person's first and last name in a single cell, the following syntax:

\[
=\text{Concatenation(<First Name>,<Last Name>)}
\]

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:

\[
=\text{Concatenation(<First Name>, (Concatenation(" ",<Last Name>)))}
\]

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:

\[
=<\text{First Name}>&" ",&<\text{Last Name}>
\]

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.
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**.

### 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, Desktop Intelligence displays the error message 'Incompatible data type'. You can convert a number to a character string using the FormatNumber() function.

**Revenue for week: 1**

<table>
<thead>
<tr>
<th>Lines</th>
<th>City</th>
<th>Sales revenue</th>
<th>Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>City Trousers</td>
<td>Austin</td>
<td>$303</td>
<td>$182</td>
</tr>
<tr>
<td>Dresses</td>
<td>Austin</td>
<td>$660</td>
<td>$257</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>$91</td>
</tr>
<tr>
<td>Shirt Waist</td>
<td>Austin</td>
<td>$2,712</td>
<td>$1,327</td>
</tr>
<tr>
<td>Sweaters</td>
<td>Austin</td>
<td>$182</td>
<td>$95</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

### 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, Desktop Intelligence displays the error message 'Incompatible data type'. You can convert a date into a character string using the FormatDate() function.
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

### 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 Desktop Intelligence 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 Desktop Intelligence document contains the variables for year, margin and city. Using this data, you can write a Desktop Intelligence 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>$253,464</td>
<td>$382,065</td>
<td>$424,790</td>
<td>50.7%</td>
<td>11.2%</td>
</tr>
<tr>
<td>Boston</td>
<td>$111,453</td>
<td>$63,657</td>
<td>$336,574</td>
<td>-42.9%</td>
<td>428.7%</td>
</tr>
<tr>
<td>Chicago</td>
<td>$349,750</td>
<td>$465,473</td>
<td>$439,865</td>
<td>33.5%</td>
<td>-5.5%</td>
</tr>
<tr>
<td>Colorado Springs</td>
<td>$203,701</td>
<td>$294,483</td>
<td>$309,966</td>
<td>-44.6%</td>
<td>5.3%</td>
</tr>
<tr>
<td>Dallas</td>
<td>$189,064</td>
<td>$279,652</td>
<td>$266,146</td>
<td>47.9%</td>
<td>2.3%</td>
</tr>
<tr>
<td>Houston</td>
<td>$568,510</td>
<td>$797,854</td>
<td>$865,542</td>
<td>40.3%</td>
<td>7.2%</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>$444,247</td>
<td>$604,780</td>
<td>$619,368</td>
<td>36.1%</td>
<td>2.4%</td>
</tr>
<tr>
<td>Miami</td>
<td>$192,479</td>
<td>$266,670</td>
<td>$318,132</td>
<td>36.5%</td>
<td>19.3%</td>
</tr>
<tr>
<td>New York</td>
<td>$779,301</td>
<td>$1,104,273</td>
<td>$1,189,166</td>
<td>41.7%</td>
<td>7.7%</td>
</tr>
<tr>
<td>San Francisco</td>
<td>$330,646</td>
<td>$471,743</td>
<td>$502,121</td>
<td>42.7%</td>
<td>6.4%</td>
</tr>
<tr>
<td>Washington</td>
<td>$310,366</td>
<td>$457,231</td>
<td>$385,415</td>
<td>47.3%</td>
<td>-15.7%</td>
</tr>
</tbody>
</table>
To compare yearly margin

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:

   \[ =<\text{Margin}> \text{ Where } (<\text{Year}>="2001") \]

   This formula calculates the margin for the year 2001 only. Note that we use the Desktop Intelligence Where operator in the formula to specify the year.

7. Click **OK**.

   Desktop Intelligence 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:

   \[ =<\text{Margin}> \text{ Where } (<\text{Year}>="2002") \]

   \[ =<\text{Margin}> \text{ Where } (<\text{Year}>="2003") \]

   Desktop Intelligence 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:

   \[ =\text{FormatNumber}((<(2002 \text{ Margin}> - <2001 \text{ Margin}>) / <2001 \text{ Margin}>)) * 100 , "0.0") & "%" \]
and the formula for 2002-2003 growth is:

\[
\text{Formula} = \text{FormatNumber}((\text{<2003 Margin>} - \text{<2002 Margin>})/\text{<2002 Margin>}) \times 100, "0.0" & "$%"
\]

**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 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.

**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.
To create a variable that returns the date of the first day of the previous month

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").

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

   ```
   FormatNumber(Year(CurrentDate()),"0000") &
   FormatNumber(MonthNumberOfYear(CurrentDate()) - 1 ,"00") &"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(Current Date())</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>
</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").

To create a variable that expresses a 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 "FirstDayOfPrevMonthAsDate" in the Name box.

4. Click the Formula tab.

5. Type the formula

   \[ \text{ToDate(<FirstDayOfPrevMonthAsString>),}'yyyymmdd' \]

   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.

To create a variable that returns the last day of the previous month as a date

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.

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

   LastDayOfMonth(<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>US</td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td>Sum for All Resorts: $3,266,524.00</td>
</tr>
</tbody>
</table>
If you insert a Sum calculation on the Revenue column, Desktop Intelligence uses the following formula by default:

\[ \text{Sum(<Revenue>)} \]

which gives the following result:

<table>
<thead>
<tr>
<th>Top 2 Resorts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Country</strong></td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td>US</td>
</tr>
<tr>
<td><strong>Sum:</strong></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 Desktop Intelligence to ignore filters when calculating, so the formula

\[ \text{NoFilter(Sum(<Revenue>))} \]

returns the total revenue for all resorts.
More examples of using formulas
Launching Desktop Intelligence with the Run Command
Overview

This appendix explains how to run Desktop Intelligence by using the Run command on Windows. You can use the Run command as an alternative way of double-clicking the Desktop Intelligence icon. You can also specify command line options such as your user name, password and other options.

To use the Run command

The following procedure describes how to launch Desktop Intelligence 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.
2. The Run dialog box appears.
3. In the Open text box, enter the path to the Desktop Intelligence executable file (Busobj.exe). By default, this file is located in the Desktop Intelligence folder. You can click Browse to specify the path, rather than type it.
4. Click OK.
5. The User Identification dialog box appears.
6. Enter the user name and password that your Desktop Intelligence supervisor provided, then click OK.
7. The Desktop Intelligence application is now launched.

Run command options

Run command options allow you to log in to Desktop Intelligence 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 &quot;user name&quot;.</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 &quot;my password&quot;.</td>
</tr>
<tr>
<td>-online or -offline</td>
<td>By default, the last connection mode of the specified user, or &quot;online&quot; the first time you launch Desktop Intelligence. 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 Desktop Intelligence. You must include the path to this file, for example: \Desktop Intelligence \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 Desktop Intelligence without showing the logo screen.</td>
</tr>
</tbody>
</table>
Name of a text file in which variables are specified. You can specify BOUSER and BOPASS, which manage your access to Desktop Intelligence. 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.

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 "Desktop Intelligence Variables" in the Database Guide included in your Desktop Intelligence package.

## Specifying BOUSER, BOPASS and Other Variables

You can use the BOUSER and BOPASS variables to manage your access to Desktop Intelligence. 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 Desktop Intelligence 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 Desktop Intelligence in online mode, which is the default working mode, Desktop Intelligence connects to the repository and reads your security information. Your user name and password are then written to either the objects.lsi file or the objects.ssi file, located in either the ShData folder or the LocData folder.
Once you have launched Desktop Intelligence 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:\Program Files(x86)\Business Objects\BusinessObjects Enterprise 12.0\win32_x86\busobj.exe -user JOHN -pass SMITH
  ```

- Declare the variables and their values in a text file in the Desktop Intelligence 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:\Program Files(x86)\Business Objects\BusinessObjects Enterprise 12.0\win32_x86\busobj.exe -vars myfile.txt
  ```

**Note:**

User names and passwords that contain spaces must be written in double quotes, for example "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 Desktop Intelligence. 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
```
Specifying prompt values in a text file

If you specify the values for a prompt in a text file and open the report from the command line, Desktop Intelligence uses the values in the file as input to the prompt. If you subsequently open the report in the normal way, the Values button for the prompt is disabled, which means that you cannot see the list of values. This occurs because the link between the report and the prompt/values in the text file has been broken. To resolve this you must rename the prompt in the report.
Desktop Intelligence and Visual Basic for Applications
Overview

You can customize Desktop Intelligence using the Microsoft Visual Basic for Applications programming language. Desktop Intelligence 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 Desktop Intelligence. For information on building VBA data providers, see "Using Visual Basic for Applications procedures."

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 Desktop Intelligence. Add-ins are usually created by those responsible in your company for adding customized features to Desktop Intelligence. 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 Desktop Intelligence. Add-ins (*.rea files) are installed by default in the UserDocs folder in the Desktop Intelligence folder. Loading an add-in makes the feature available in Desktop Intelligence and adds any associated commands to the appropriate menus.

Unloading an add-in removes its features and commands from Desktop Intelligence, 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 Desktop Intelligence SDK Reference Guide.
Using macros

Macros are created and stored inside Desktop Intelligence documents (.rep files) or Desktop Intelligence 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.

To run 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 documents are displayed in the Macro Name list.

3. Select the name of the macro you want to use and click Run.
Note that 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.

**To open the Visual Basic toolbar**

- Right-click on any other open toolbar and click *Visual Basic* on the shortcut menu.
- The arrow button on the left opens the macros dialog box.
- The next button 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**

Desktop Intelligence add-ins are Visual Basic for Applications programs that add optional commands and features to Desktop Intelligence. You can distribute add-ins you have created to other users and retrieve and use add-ins that others have created.
To install an add-in

1. Click Add-Ins on the Tools menu.
   The Add-Ins dialog box opens. Add-ins that have been installed have a check mark. Add-ins without a checkmark are on the computer but not installed.

2. 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.

3. 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.

To uninstall 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 Desktop Intelligence, 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 Desktop Intelligence documents.

Converting scripts to macros

In Desktop Intelligence 4.1, you could create scripts to automate tasks using the ReportScript programming language. This programming language has been replaced in Desktop Intelligence and upwards by the Visual Basic programming language.

Desktop Intelligence 6.5 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.

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 Desktop Intelligence guide.

**Using the Visual Basic editor**

You can open the Visual Basic Editor directly from Desktop Intelligence to create macros and programs to use in Desktop Intelligence. 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 Desktop Intelligence guide.
Using the Visual Basic editor
Overview
This appendix describes the command-line options for Desktop Intelligence.

If you have been using an earlier version of Desktop Intelligence you will notice that the command-line options are slightly different. This is due to architecture changes. The Repository has replaced the CMS, there have been changes in the security domain and the calculation engine.

Command-Line is used essentially for batch processing and most of the command line options deal with authentication.

Note:
Command-Line options are available only on Windows.
The desktop Intelligence binary is located in:

$INSTALLDIR\BusinessObjects Enterprise 11.5win32_x86

Where $INSTALLDIR is the installation directory.

Here is the syntax:

```
busobj .exe [-user <username> [ -pass <password> ] 
..................[ -system <systemname> ] 
..................[ -auth "enterprise" | "ldap" | "ad" | "nt"
..................[ -offline | -online 
..................[ -blind] 
..................[ -nologo] [ -document <documentname>] 
```
Options
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-user &lt;username&gt;</td>
<td>User name is used to log into Desktop Intelligence</td>
</tr>
<tr>
<td>-pass &lt;password&gt;</td>
<td>Password is used to log into Desktop Intelligence</td>
</tr>
<tr>
<td>-system &lt;systemname&gt;</td>
<td>System name is where the user is authenticated</td>
</tr>
<tr>
<td>-auth</td>
<td>Authorization mode is used to authenticate the user.</td>
</tr>
<tr>
<td></td>
<td>Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• &quot;enterprise&quot; for Enterprise</td>
</tr>
<tr>
<td></td>
<td>• &quot;ldap&quot; for LDAP</td>
</tr>
<tr>
<td></td>
<td>• &quot;ad&quot; for Active Directory</td>
</tr>
<tr>
<td></td>
<td>• &quot;nt&quot; for Windows Authentication</td>
</tr>
<tr>
<td></td>
<td>The Default value is &quot;enterprise&quot;</td>
</tr>
<tr>
<td>-online</td>
<td>Start Desktop Intelligence in online mode. This option cannot be used with the -offline option. By default, this is the option selected.</td>
</tr>
<tr>
<td>-offline</td>
<td>Start Desktop Intelligence in offline mode. This option cannot be used with the online option.</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-blind</td>
<td>This option must be used with authentication options. It is intended to run Desktop Intelligence in batch mood. Desktop Intelligence starts, opens/refreshes a document or runs a script then exits, without displaying the user interface.</td>
</tr>
<tr>
<td>-nologo</td>
<td>The Business Objects logo is not displayed when Desktop Intelligence is started. By default the logo is displayed.</td>
</tr>
<tr>
<td>-document &lt;documentname&gt;</td>
<td>Document name is open when Desktop Intelligence is started. Document name can be either relative path or an absolute path. By default no document is specified to be opened. This option can be used in batch mode or in UI mode.</td>
</tr>
<tr>
<td>-help</td>
<td>displays the syntax for Desktop Intelligence command line</td>
</tr>
</tbody>
</table>
Help Message
busobj.exe [-user <username> [-pass <password>]
........................[-system <systemname>]
........................[-auth "enterprise" | "ldap" | "ad" | "nt"
........................[-offline | -online
........................[-blind]
........................[-nologo] [-document <documentname>] [-help]
Error messages
### Error Message

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>The syntax of the command is incorrect.</td>
<td>Error in the syntax of the command-line.</td>
</tr>
<tr>
<td></td>
<td>Displays the help message.</td>
</tr>
<tr>
<td>You did not specify any user name.</td>
<td>If the user uses -pass, -system, -auth, -offline, -online, -blind, options without -user.</td>
</tr>
<tr>
<td>-offline and -online options cannot be used together.</td>
<td>There is only one possible mode.</td>
</tr>
<tr>
<td>Invalid User name in Offline Mode. Please log in first in OnLine mode for that user.</td>
<td>To use offline mode, the user must have logged in at least once in online mode.</td>
</tr>
<tr>
<td>Your login is not valid.</td>
<td>Incorrect credentials.</td>
</tr>
<tr>
<td>Your password is not valid.</td>
<td>Incorrect credentials.</td>
</tr>
<tr>
<td>This document does not exist.</td>
<td>Incorrect path in the filename.</td>
</tr>
<tr>
<td>The authentication mode is not valid.</td>
<td>When the user types a mode different than the possible modes.</td>
</tr>
<tr>
<td>Information Resource</td>
<td>Location</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>SAP BusinessObjects product information</td>
<td><a href="http://www.sap.com">http://www.sap.com</a></td>
</tr>
</tbody>
</table>
|                                          | You can access the most up-to-date documentation covering all SAP BusinessObjects products and their deployment at the SAP Help Portal. You can download PDF versions or installable HTML libraries.  
|                                          | Certain guides are stored on the SAP Service Marketplace and are not available from the SAP Help Portal. These guides are listed on the Help Portal accompanied by a link to the SAP Service Marketplace. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative. |
| SAP Service Marketplace                   | http://service.sap.com/bosap-support > Documentation  
|                                          | • Installation guides: https://service.sap.com/bosap-instguides  
|                                          | • Release notes: http://service.sap.com/releasenotes  
|                                          | The SAP Service Marketplace stores certain installation guides, upgrade and migration guides, deployment guides, release notes and Supported Platforms documents. Customers with a maintenance agreement have an authorized user ID to access this site. Contact your customer support representative to obtain an ID. If you are redirected to the SAP Service Marketplace from the SAP Help Portal, use the menu in the navigation pane on the left to locate the category containing the documentation you want to access. |
| Developer resources                       | https://boc.sdn.sap.com/  
<p>|                                          | <a href="https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary">https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary</a> |</p>
<table>
<thead>
<tr>
<th>Information Resource</th>
<th>Location</th>
</tr>
</thead>
</table>
These articles were formerly known as technical papers. |
| Notes | [https://service.sap.com/notes](https://service.sap.com/notes)  
These notes were formerly known as Knowledge Base articles. |
From traditional classroom learning to targeted e-learning seminars, we can offer a training package to suit your learning needs and preferred learning style. |
| Online customer support | [http://service.sap.com/bosap-support](http://service.sap.com/bosap-support)  
The SAP Support Portal contains information about Customer Support programs and services. It also has links to a wide range of technical information and downloads. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative. |
Consultants can accompany you from the initial analysis stage to the delivery of your deployment project. Expertise is available in topics such as relational and multidimensional databases, connectivity, database design tools, and customized embedding technology. |
Index

## Error Codes
- ####### error 280
- #ALERTER error 280
- #COMPUTATION error 266
- #DICT.ERROR error 281
- #DIV/0 error 282
- #ERROR error 283
- #IERR error 283
- #MULTIVALUE error 271
- #OVERFLOW error 286
- #SYNTAX error 286
- #UNKNOWN error 287

## A
- Abs function
  - Excel equivalent 288
  - accessing data sources 26
- alerters
  - copying 172
  - creating 169
  - deleting 173
  - displaying/hiding 171
  - editing 172
  - naming 170
  - selecting data 170
  - setting conditions 170
  - setting formatting 170
  - switching off 171
  - switching on 171
- analyzing data
  - on-report analysis 110
  - overview 110
  - using drill mode 111
  - using slice-and-dice mode 111

## Asc function
- Excel equivalent 288

## Average function
- Excel equivalent 288

## B
- blocks
  - using separate data providers for 93
- BOPASS variable 330
- BOUSER variable 330
- breaks
  - and ranking 166
- BusinessObjects
  - defined 18
  - demo materials and samples 21
  - launching with the Run command 328
  - source of data 18
- BusinessObjects administrator
  - and free-hand SQL 75

## C
- calculations
  - adding to reports 218
  - examples of 221
  - re-using 222
- Ceil function
  - Excel equivalent 288
- Char function
  - Excel equivalent 288
- charts
  - displaying data in 96
  - combining data 31
complex condition calculations
    examining the SQL of 194
complex conditions
    applying on queries 181
    applying with a calculation 190
    applying with calculations 190
    deleting 190
    editing 190
    tips for applying 187
    using the Different From operator 188
    using the Except operator 188
    using the In List operator 188
    using the Not In List operator 188
    using wildcard characters 187
Concatenation function
    Excel equivalent 288
conditions
    applying to queries 57
    defined 57
connections
    defining 29
    setting up 22, 29
Cos function
    Excel equivalent 288
Count function
    Excel equivalent 288
crosstab
    building from table or 2-D chart 145
    moving data between columns and rows 147
    reducing to table or 2-D chart 145
crosstabs
    displaying data in 95
custom hierarchies 131
data providers (continued)
    basing on existing data providers 97
    cancelling 36
    combinations of 93
    creating a base data provider 107
    deleting links between 101
    deleting the data from 22
    editing 34, 35
    free-hand SQL 66
    getting statistics on 106
    linking 95, 98
    linking dimensions 99
    linking existing 100
    linking when inserting new block 99
    personal data files 78
    prompted linking 98
    purging and deleting 107
    reasons for renaming 105
    renaming 104, 106
    restrictions on access 28
    restrictions on building 28
    situations requiring linking 98
    stored procedures 76
    using efficiently 107
    using for separate blocks 93
    VBA procedures 81
    viewing data from previous execution 22
    viewing partial results 22
    workflows for creating 31
    XML 85
data sources 92
    accessing 26
    combining 31
    combining different in one report 92
    list of 26
database connections
    creating 66, 68
    editing 68, 69
    managing using free-hand SQL editor 68
    selecting 66
demonstration materials 47
eFashion 49
demonstration materials (continued)
generic SQL scripts 47
Island Resorts Marketing 47
dimensions
  adding to hierarchies 132
  restrictions on combining 132, 133
document, schedule export of 38
documents
  creating new 42
  creating new data providers in 42
  editing data providers in 42
  refreshing 20
drill filters
  bringing in new data using 129
drill mode
  analyzing measures 126
  and hierarchies 115
  bringing in new data using filters 129
  changing data in tables 122
  continuing drill-down 117
  default BusinessObjects actions 116
  defined 111, 114
  displaying values in the Drill toolbar 118
  drilling across 119
  drilling down 117
  drilling down to another hierarchy 119
  drilling on charts 120
  drilling on multiple hierarchies 120
  drilling through to the database 129
  drilling up 118
  drilling using custom hierarchies 131
  printing a report from 136
  printing from 135
  re-organizing the drill toolbar 125
  setting options 136
  setting up a report for 115
  switching to 116
  taking report snapshots 128
  undoing drill actions 118
  using 116
  using the drill toolbar 124
drill toolbar
  displaying values in 118
  inserting contents as a report title 135
  moving objects to block from 125
  re-organizing 125
  removing objects from 125
Drill toolbar
  moving variables from 122
  using 124
  drilling down 117, 118
  drilling through 129
  drilling up 118

E
error messages
  ####### 280
  #ALERTER 280
  #COMPUTATION 266
  #DICT.ERROR 281
  #DIV/0 282
  #ERROR 283
  #IERR 283
  #MULTIVALUE 271
  #OVERFLOW 286
  #SYNTAX 286
  #UNKNOWN 287
tips and tricks regarding 288
EuroConvertFrom function
Excel equivalent 288
EuroConvertTo function
Excel equivalent 288
EuroFromRoundErr function
Excel equivalent 288
Euros
  converting from 226
  converting to 225
  converting to and from 223
  displaying rounding errors in calculations 226
  displaying the euro symbol 224
EuroToRoundErr function
   Excel equivalent 288
Exp function
   Excel equivalent 288
export, scheduling of 38
extended calculation syntax 234
   and Rank function 251
context operators 263
ForAll operator 248
ForEach operator 248
keywords 254
   quick reference 258

F
Fact function
   Excel equivalent 288
Fill function
   Excel equivalent 288
filters
   adding 156
      applying in slice-and-dice mode 152
      creating complex filters 158
      displaying names of in special fields 157
      editing complex filters 159
      ignoring 159
      inserting 154
      inserting complex filters 158
      managing 155
      removing 157
      selecting different values for 155
Floor function
   Excel equivalent 288
FormatDate
   Excel equivalent 288
FormatNumber
   Excel equivalent 288
formulas
   creating using Formula Editor 296
   defined 294
   guidelines on syntax 298
   qualifying 134
free-hand SQL
   and sorts 61
      creating a connection 68
      creating interactive reports with 71
      creating reports using 66
      creating/editing a connection for 68
      editing a connection for 69
      editing an SQL script 68
      example of 70
      example of query with prompt 74
      opening an existing script 62
      restrictions on scripts 75
      viewing raw data 66
      writing a new script 62
functions
   Excel equivalents of 309
G
grouped values
   adding to drill hierarchy 304
H
hidden data
   displaying 169
   hiding report data 168
hiding data 168
hierarchies
   adding dimensions to 132
   changing the order of dimensions in 132
   creating 133
   creating custom hierarchies 133
   custom hierarchies defined 131
   deleting 132
   drilling on multiple hierarchies 120
   drilling up on multiple hierarchies 121
   drilling using custom hierarchies 131
   editing 131
   in Island Resorts Marketing 115
   qualifying data for 133
   removing dimensions from 132
hierarchies *(continued)*
renaming 132
restrictions on editing 131
viewing 55
highlighting data see Alerters 169

I

InitCap function
Excel equivalent 288

L

Left function
Excel equivalent 288
LeftTrim function
Excel equivalent 288
Length function
Excel equivalent 288
lists of values
assigning data from a dBase file 214
assigning personal data from a text file 212
assigning personal data from an Excel file 213
assigning personal data to 212
customizing 210
defined 210
displaying, purging and refreshing 214
editing 211
Ln function
Excel equivalent 288
local variables
creating 299
creating by grouping values 301
defined 298
qualifying 134
transforming formulas into 300
Log function
Excel equivalent 288
Log10 function
Excel equivalent 288
Lower function
Excel equivalent 288

M

master/detail reports
activating/deactivating multiple sections 143
deactivating sections of 142
ranking in 166
reactivating sections of 143
reorganizing 140, 141
structuring existing reports as 139, 140
undoing 141
working in slice-and-dice mode with 139
Match function
Excel equivalent 288
matrix chart, 3-D
building from table or 2-D chart 145
Max function
Excel equivalent 288
measure objects 48
measures
analyzing in drill mode 126
collapsing 127
defined 126
expanding 127
Median function
Excel equivalent 288
Min function
Excel equivalent 288
Mod function
Excel equivalent 288
Month function
Excel equivalent 288
MonthNumberOfYear function
Excel equivalent 288

O

objects
changing order of in queries 53
displaying in Query Panel 52
objects (continued)
   including in queries 52
   measure 48
   moving from Drill toolbar to blocks 125
   removing from queries 53
   searching for in Query Panel 52
operators
   Different From 188
   Except 188
   In List 188
   Not In List 188
options
   setting for drill mode 136

P
personal data files 22
   benefits of 78
   creating reports using 79
   setting options 79
Pos function
   Excel equivalent 288
Power function
   Excel equivalent 288
predefined conditions
   applying to queries 58
   in Island Resorts Marketing universe 58
   removing 58
printing
   from drill mode 135
prompts
   in free-hand SQL 71
   syntax in free-hand SQL scripts 71

Q
Quarter function
   Excel equivalent 288
queries
   applying complex conditions on 181, 183
   building in Query Panel 51
   building on universes 49
   building powerful queries 54
   eliminating duplicate rows 61
   generating SQL without retrieving data 61
   running 53
   running on different universes 62
   saving definitions of 54
   setting options 61
   specifying number of rows to return 61
   using output in conditions 198
query panel
   building a query using 51
   changing object order 53
   described 51
   displaying 49
   including objects in queries 52
   removing objects from queries 53
   running queries from 53
   searching for objects in 52

R
ranking
   and breaks 166
ranking data 164
   displaying percentages 166
   displaying subtotals 166
   editing an existing ranking 167
   ranking in master/detail reports 166
   removing a ranking 167
RelativeDate function
   Excel equivalent 288
Replace function
   Excel equivalent 288
Report Manager
   moving variables from 122
reports
   adding columns to 98
   combining data sources in 93
   including data from different data sources 92
   making copies of 128
reports (continued)
  sharing 19
Right function
  Excel equivalent 288
RightTrim function
  Excel equivalent 288
Round function
  Excel equivalent 288

S

scheduling 38
scope of analysis
  and hierarchies 55
  defined 55
  defining 55
  defining default 56
  defining manually 56
  drilling through to the database 129
  extending 128
security 20
send by email 43
send to groups 43
send to users 43
sending documents 42, 44
Sign function
  Excel equivalent 288
simple conditions
  applying 59
  deleting 59
  selecting different values for 59
Sin function
  Excel equivalent 288
slice-and-dice mode
  adding data in 96
  defined 138
  displaying and removing data 148
  making calculations in 151
  moving data in crosstabs 147
  positioning data horizontally in 144
  repositioning data vertically 146
  transforming blocks in 149
slice-and-dice mode (continued)
  using filters in 152
  working in 138
  working with blocks in 148
  working with crosstabs/3-D matrix charts in 144
  working with master-detail reports in 139
sorting data 160
sorts
  adding from the Sorts dialog box 164
  and free-hand SQL 61
  applying custom 161
  applying on report data 161
  defined 60
  defining priority 61
  defining sort priority 163
  inverting 60, 161
  managing 163
  removing 60, 161, 164
SQL
  defined 66
  parsing 66
Sqrt function
  Excel equivalent 288
StDev function
  Excel equivalent 288
StDevP function
  Excel equivalent 288
stored procedures
  defined 22, 76
  example of 76, 77
  restrictions on 76
  supplying parameters to 77
  using to retrieve data 76
Substr function
  Excel equivalent 288
Sum function
  Excel equivalent 288
Index

T
Tan function
Excel equivalent 288
ToDate function
Excel equivalent 288
transparent sorts
defined 61
Trim function
Excel equivalent 288
Truncate function
Excel equivalent 288

U
universes
building basic queries on 49
creating 46
defined 22, 46
Upper function
Excel equivalent 288
user objects
creating 176
creating a time hierarchy for 181
defined 177
deleting 180
editing 180
qualifying 135
reasons for creating 176
restrictions on 177
sharing with other users 178

V
Var function
Excel equivalent 288
variables
compatibility rules 95
VarP function
Excel equivalent 288
VBA procedures
accessing Outlook with 82
creating a report using 82
using 81

W
Week function
Excel equivalent 288
WordCap function
Excel equivalent 288

X
XML
building an XML filter 88
creating a report using 88
defined 86
example of 86
setting file locations 89
XML query panel 89

Y
Year function
Excel equivalent 288