

DataHabitat

DataHabitat Corporation
www.datahabitat.com

Concepts, Tips and Tricks

Version 2.0

INTRODUCTION	3
ACTIONS AND OBJECTS	4
FUNCTIONS	4
DATA EXTRACTION.....	5
DATA MODELING.....	5
DATA PRESENTATION.....	6
QUERIES	7
GRAPHS.....	8
REPORTS.....	8
NEW FIELDS AND FORMULAS.....	9
MULTIDIMENSIONAL QUERIES.....	10
RANKINGS	10
MAPS.....	11
USER PROFILES	12
DATA EXPORT	13
METADATA MANAGEMENT.....	13
CONCLUSION	13

Introduction

DataHabitat is a complete, packaged Business Intelligence solution that utilizes Data Warehouse technology to simplify reporting and data analysis tasks. DataHabitat has been compared to different kinds of tools, from report writers to sophisticated high-end OLAP software. The fact is that DataHabitat provides a gamut of features that enable users to build and maintain data warehouses, create simple and complex reports, create dynamic queries and graphs, and perform multidimensional, OLAP data analysis. And all these features come together in a single, integrated package.

This paper will review some concepts and facilities of the product, emphasizing how the concepts are applied and what tools DataHabitat provides to its users. Reading this material is recommended as a pre-requisite for those taking the DataHabitat web-based training course. Also, before reading this paper, it is recommended that you see our online demos available at <http://www.datahabitat.com/demos.html>.

The functional areas discussed include:

- Data Extraction
- Data Modeling
- Data Presentation
- Queries
- Graphs
- Reports
- New Fields and Formulas
- Multidimensional Queries
- Rankings
- Maps
- User Profiles
- Data Export
- Metadata Management

This functionality will allow you to:

- Perform tendency analysis
- Do comparative analysis between time periods
- Create and analyze rankings
- Audit systems and databases
- Perform ad-hoc queries
- Create geographic analysis based on maps
- Define alerts, flags and business indicators
- Perform simulations
- Create new information and build simple and complex reports

Actions and Objects

The software user interface is based on **actions** and **objects**. An action (define, browse, modify) combined with an object (graph, report, summary) will initiate a dialog, such as **define graph** and **modify report**.

Following is a list of all the actions and objects available and a short description:

Action	Description
Browse	Allows access to different types of information and objects
Define	Creates different objects
Delete	Deletes different objects
Import	Extracts data and metadata from transactional databases
Modify	Changes object properties
Refresh	Reloads imported data and recalculates formulas
Save	Menu option for saving queries and interface formats

Object	Description
Field	Represents a column in a database table
Table	Represents a database table
Relationship	Represents a connection between two tables
Filter	Defines a sub set of records in a table
Formula	An equation defining the value of a field
Group	A named group of tables or queries
Graph	A figure that shows relationships between quantities
Interface	A particular way of displaying icons, bars and work area
Map	Maps that can be associated to tables
Dimension	A fact table (i.e. sales order) seen from different dimensions
Sort	A sort order classifying data by certain fields
Profile	Set of user rights to access data and functions
Query	A set of windows containing data and graphs about a subject
Report	Data formatted for printer, HTML or other export format
Summary	Sum, maximum, minimum, average values of a field
User	Belongs to a user profile

Functions

Each window displayed on the work area has a set of functions associated to it. These functions can be accessed via the function bar or by right-clicking the mouse on the active window. When a table data is being displayed, like Customers or Sales Orders, the associated functions include reports, sorts, filters, summaries and graphs. So to access the reports for the Customer table, simply open the table via the Browse Table dialog, right-click on the table and select the function Report. The Customer reports will then be listed.

Data Extraction

The **Import Table** dialog provides several tools for efficient data extraction tasks. Naturally, you can extract data from any ODBC source. In the case of specific application mappings, like **DataHabitat for MAS 90/200**, the main tables have already been imported from the sample company database. This means that the data structure (metadata) is ready to go. In this case, you just need to run the **Refresh Table** dialog to get to your real data.

When importing a database for the first time, you will be able to select tables and columns. It is also possible to import the same table several times, selecting different columns, and storing the table under different names (similar to creating views in SQL).

When importing data, you can also specify filters. This way you can, for example, just import data from the last three years.

Other powerful functionality is the ability to group data when importing. For example, you can import the Sales Order table grouping by customer number, and create aggregated values (fields) for total order amount and average order amount. The ability to group data comes in handy specially when dealing with large data volumes. A source table with millions of sales order records can be reduced to a few thousand records when grouped.

Again, you can import, for instance, Sales Order many times, using different filters and groups, in order to best fit your requirements.

In order to update your imported data, use the **Refresh Table** dialog. Note that you can schedule data refresh with the DH Scheduler or Windows Task Manager. You can also refresh the data bringing from the source database only the records that have been added since a certain date or according to certain criteria. Importing and refreshing properties can be modified via the **Modify Table** dialog.

Data Modeling

The tools for data modeling allow you to create sort orders (primary and foreign keys) and relationships (joins) between tables. In the case of **DataHabitat for MAS 90/200**, **DataHabitat for ACCPAC Advantage Series** and other Datahabitat accounting packages, most of the sort orders and relationships have been defined. You can, however, create new relationships at will. This can be helpful when integrating the accounting application to vertical applications and other systems.

The basic requirement to create a relationship between two tables is to have the same field in both tables. For instance, you can create a relationship between any two tables that have the field **customer number**.

Note that for optimal performance and to avoid errors, make sure the fields have the same properties (type, length, decimal positions, for example). You can modify the properties of a field via the Modify Table dialog.

Also note that you can modify table and field names, for better legibility. Field names can have up to three lines. Use the Modify Table dialog to edit table and field headers.

Data Presentation

At the core of DataHabitat's software architecture is its **Relationship Engine**. This engine manages the relationships between tables, and drives the way queries and reports are created.

Relationships help you quickly get to data related to a certain table. They also help you drill-down throughout the whole relational model. In fact, you navigate from table to table by using relationships.

Even though DataHabitat lets you only define ONE-TO-MANY (1-N) and ONE-TO-ONE (1-1) relationships, you can navigate using also the MANY-TO-ONE (N-1) relationship function. This means you can navigate both down and upward in the data model, as shown below:



If **Customers** is the active table, pressing the **Relationship** function will list **Sales Orders** and **Invoices**. On the other hand, pressing the **Relationship N-1** function will list **Regions** and **Salespersons**.

The Relationship Engine allows you to gather data from several tables for creating formulas, reports and graphs.

When creating a Sales Order report, for instance, you can use fields from Customers and Regions, as well as Sales Order Items. For reports, you can select fields from up to six hierarchical levels¹ in the data model.

¹ A parent-child relationship represents two hierarchical levels. Parent-child-grandchild represents three levels, and so on.

Queries

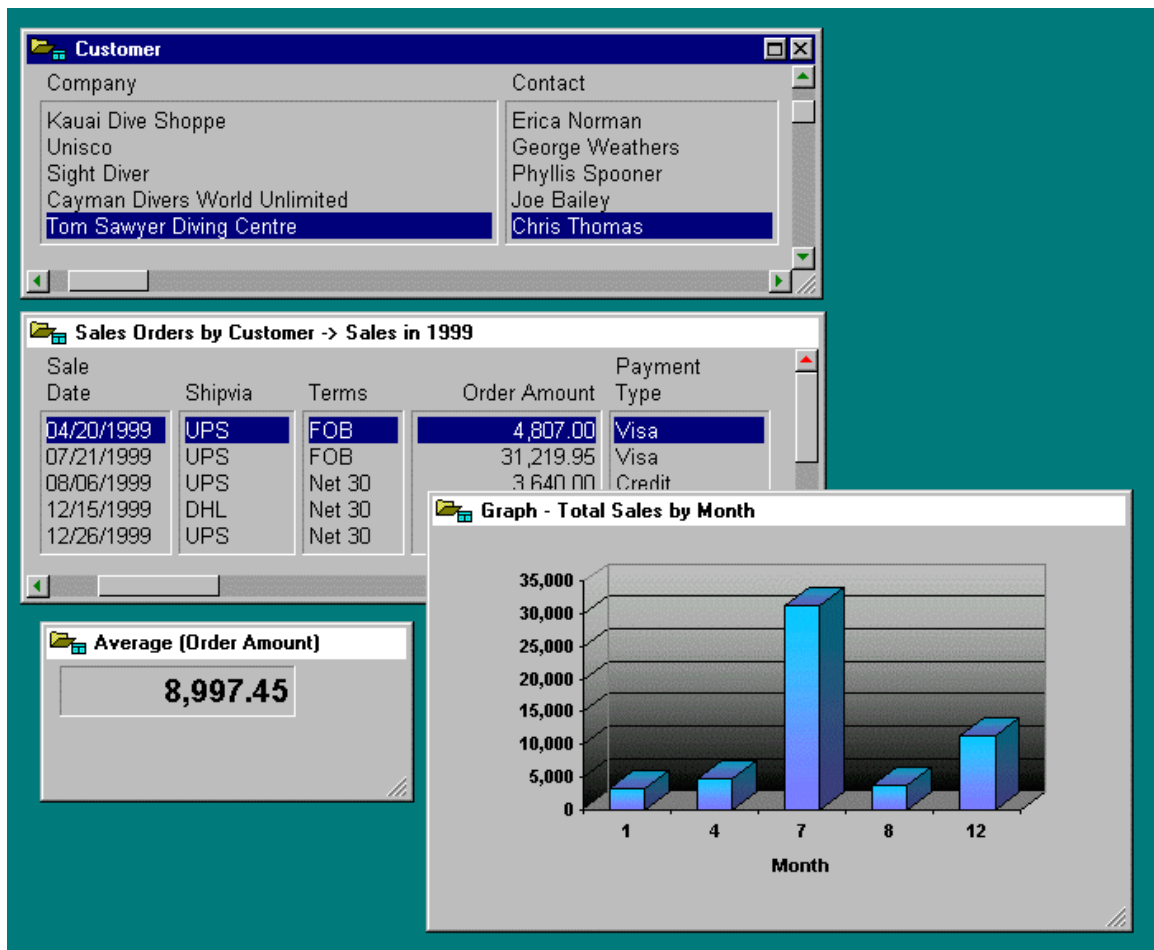
In DataHabitat, a query is simply a set of windows saved under a specific name. These windows may contain table data, a summary value, a graph, etc. There may also be several related tables in a query.

Using the Browse Table dialog, you can access data from any table in the model. While looking at a particular table, you can select different sort orders, filters, summaries, and graphs by selecting the appropriate function.

Then, by using the **Relationship** and **Relationship N-1** functions, you can drill-down and navigate through the data model.

In the example below, we have the Customer table, which is related to Sales Order. The Sales Order table has a filter for Orders in 1999. And the graph on the side shows orders month by month. In addition, there is a summary with average order amount.

The queries are dynamic because you can then select any particular customer, and the related information will change.



Graphs

Graphs are always associated to a field. When you start the **Define Graph** dialog, DataHabitat prompts you to select a table.

Graphs become dynamic when they are used in queries with multiple tables. In the example above, the graph shown is associated to the Sales Orders table, which in turn is related to Customer. So when you select a different customer, the graph will dynamically display new information. The graph will also change if you set a different filter to Sales Order.

The Define Graph dialog uses a wizard to assist you in building graphs. The wizard lets you group data as well. For example, you can define a graph based on the Customer table, group data by Customer Name (in the X-axis) and plot the sum of Invoice total in the Y-axis. This way you are combining data from two tables in the same graph.

DataHabitat uses Microsoft Graph as its graph editor. To launch the graph editor, just right-click on a graph and select the option *open* to change graph properties. Users of Microsoft Excel will find the editor familiar.

Reports

When creating a report, always use the **child** table as the base table. **One-to-many** relationships are sometimes called **parent-child** relationships, i.e. one customer has many sales orders (one parent has many children).

If you want a report listing Sales Orders and Customer data, start with the Sales Order table. This way you will be able to list all sales orders, as well as use **any field** from the Customer table. You will also be able to use **summaries** from the Sales Order Items table.

You can select fields from up to six hierarchical levels in the data model. But remember, you can use all fields from tables "above" the base table, but only summaries from tables "below".

After creating the basic report with the wizard, you can create data groupings, totals, and sub-totals, modify fonts, etc. by using the Microsoft **Report Designer**, which can be invoked via the **Modify Report** dialog, using the **Edit Report** function. Users of Microsoft Visual Basic and Microsoft Visual FoxPro will find the designer familiar.

New Fields and Formulas

The power to create new fields and derive their values, through formulas, from anywhere in the relational data model enables you to develop new data structures and create new information.

The **Define Field** dialog presents a series of field categories and models you can choose from. If you do not find a field model that fits your needs, you can create a new one. With the **Field Category** window active, use the **New Field Model** function and follow the wizard.

When defining a new field for a formula, make sure its properties (format, size, decimal positions, etc.) are coherent with possible formula results. For example, if you create a new field called **Average Monthly Sales Amount** in the Customer table, and the formula includes the **Total Invoice Amount** value from the Invoice table, make sure the new field size and decimal positions are adequate.

The **Define Formula** dialog enables you to define equations using operands, constant values, functions, relationships and filters.

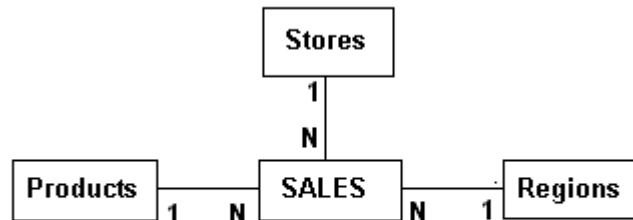
Operands include fields and summaries from the table itself, as well as fields from related tables. You can use any fields from **parent** tables, but only **summaries from child tables**.

Functions include absolute value, month, week, integer, square root, and substring.

Formulas can also contain **filters from child tables**. This means you can create a formula in the Customer table to calculate the average sales amount in 1999, and use a filter to set this period in the Invoice table.

Multidimensional Queries

The *Browse Dimension* dialog lists the tables that have more than one MANY-TO-ONE relationship. For example, the table Sales has this type of relationship with Regions, Products, and Stores.



After Selecting Sales, for example, choose the dimension tables through the *Relationship N-1 function*. If you wish, you can lock (freeze) one or more dimensions with the *Lock* function.

To enrich your analysis, activate graphs and summaries associated to the Sales table. As you change dimensions, the results will be reflected automatically.

Save the query with the menu option <File - Save Query>.

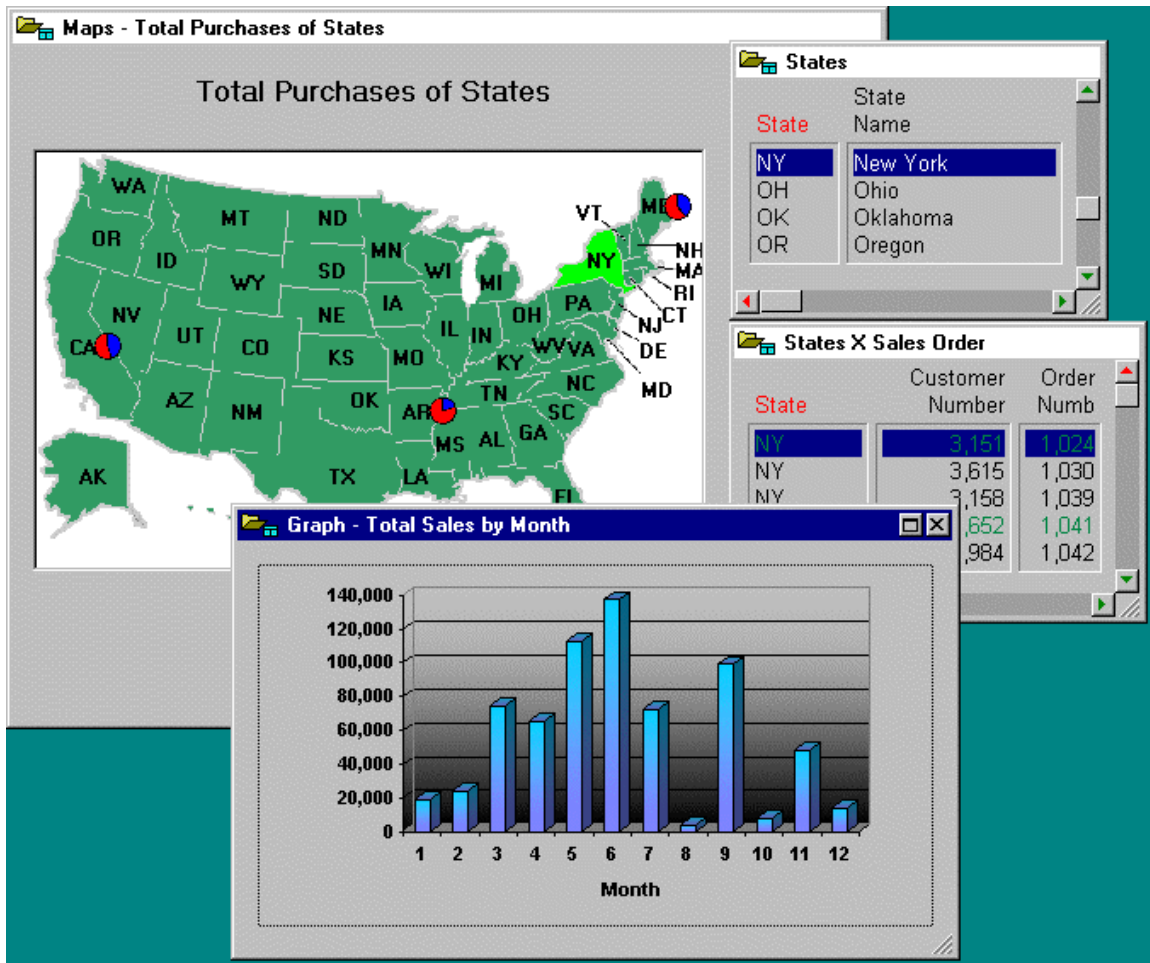
Rankings

In the *Modify Table* dialog, select the Ranking function and follow the wizard. Then use the Refresh Table dialog and the table will be imported again according to the ranking criteria.

Warning: If you define ranking criteria, this will affect the way the table is imported and data refreshed. If you want to keep the original table intact, import the table again under a new name, and then define the ranking criteria.

Maps

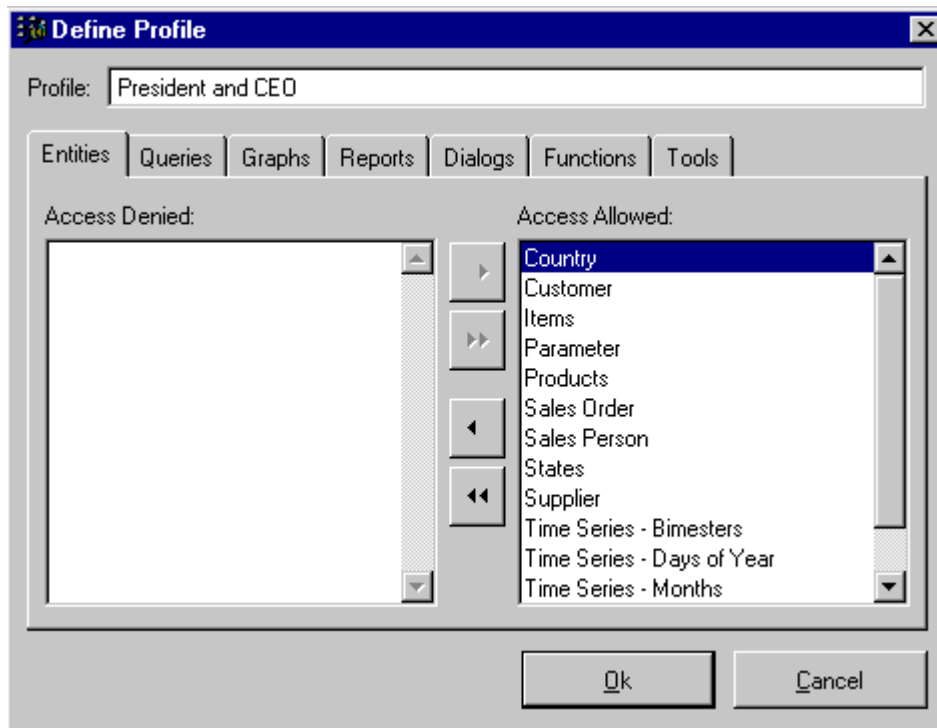
DataHabitat comes with a United States state map. This map can be associated to any table that has a field with the two digit state code. Using the Browse Table dialog, select the State table and then start the desired relationship (in this example, Sales Order). Now go back to the State table and use the Map function to invoke a list of maps. Now clicking on a particular state will cause the related tables and graphs to dynamically display new information.



User Profiles

Use the *Define Profile* dialog to determine user rights. You can establish user rights to tables, queries, graphs, reports, dialogs, functions and tools.

If you are creating a profile for a company CEO, you can provide only the necessary dialogs and tools like Browse Query and Browse Report. Use profiles to streamline the software interface, as well as manage access to data.

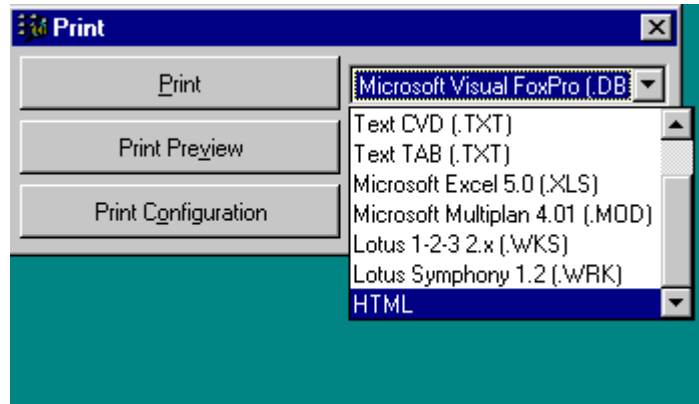


As you will notice, DataHabitat does not yet provide access control at the field level. However, you can work around this by importing the table again, leaving out the restricted field.

Data Export

You can export data in two ways. One of the functions associated to a table window is **Export to Excel**. This function will invoke a wizard that allows you to select fields, and either save the result as a file or launch Excel directly. Note that prior to exporting, you can set a **filter** to restrict the records to be exported.

The other way to export data is through the Export Report capability. In the Browse Report dialog, double click a report title. This will launch the dialog box below. Then simply choose a format and click the Export button.



Metadata Management

The **View Table** dialog provides access to metadata relating to all objects associated to a table. Try also using the Report function in that dialog. This will run a report that lists all fields, filters, summaries, relationships and formulas (with equations) for that particular table. This is a good way to document your decision support projects.

Conclusion

DataHabitat provides a wide range of functionality to meet the most demanding information management needs. Developing solutions with the product requires familiarity with its concepts and knowledge of its tools. The product has an easy-to-use interface, and you can master its functionality in little time. Learning the product will enable you to significantly decrease the time it takes to produce queries and reports, and more important yet, it will allow you to discover and analyze new business information, as you could not before.