DBANALYST™ USER MANUAL

v3.0
Seabird Software, LLC
November 2005

Seabird Software, LLC
Lake Bluff, Illinois, United States of America

Contents

Introduction

dbAnalyst is a database administration solution for managing diverse and multiple relational databases[1]. Its cross-platform capability allows users to efficiently manage heterogeneous database platforms easily using a single front-end tool. With dbAnalyst, users boost their productivity by utilizing a single tool for all their databases, regardless of the database vendor.

Product Benefits

Database Administrators

dbAnalyst enables database administrators to accomplish more with the time they have available in their workday. It eliminates the tedious tasks associated with researching schema dependencies when making object changes. Also included are a host of utilities, which condense DBA tasks that usually take hours or days down to minutes.

Developers

dbAnalyst provides a number of tools to help developers eliminate repetitive tasks and simplify their day to day work. While some of these features are available elsewhere, dbAnalyst brings them together in one convenient package. Using the powerful *Schema Explorer*, *SQL Console* and *Query Builder*, developers can quickly extract and move schema from development to other environments, as well as create and query objects much quicker than using old-fashioned techniques.

Casual Users

DbAnalyst supports the casual user in several ways. dbAnalyst explains processes such as database structure exploration in simple terms and provides graphs to illustrate database schemas. DbAnalyst makes it easier to query schemas for database content and to reverse-engineer database schemas to understand the database structure. dbAnalyst effectively translates the meaning of any database into a language that a casual user can understand.

[1] At the time of this writing, dbAnalyst supports Oracle, Microsoft SQL Server, Microsoft Access, Sybase iAnywhere, PostgreSQL, dBase and MySQL.

Concept

All relational databases differ in functionality, feature, scalability, and quite often in the degree of support offered by *Structured Query Language* (SQL). One of the difficulties met everyday by database developers and database administrator alike is the limited quantity of database tools for a single database vendor, and the even more limited number of cross-platform tools.

In general, database tools must support the current version as well as previous versions of the vendor database they support. Any time a vendor issues a new database release, the database tool must also be issued in an updated version. For the user, this means constantly keeping track of various versions. dbAnalyst eliminates this problem.

dbAnalyst was designed to be database agnostic: dbAnalyst does not 'know' specifics about any database, nor does it 'know' administration rules. All information about databases is provided as external XML files, and database access is exclusively provided using the Microsoft Data Access Components. This approach makes dbAnalyst more resilient to vendor database changes. This also enables dbAnalyst to provide a uniform database tool interface to products from different database vendors.

Getting started with dbAnalyst

Installing dbAnalyst

dbAnalyst distribution media is a self expandable InstallShield© executable. Executing the installation media will result in the creation of a new folder under the Program File folder, usually 'Seabird Software, LLC/dbAnalyst', and copying the dbAnalyst binary executable, as well as several XML files, and several PDF documents including this document. The InstallShield© executable also includes *Modify*, *Repair*, *Remove* features to alter your dbAnalyst installation.

The latest version of dbAnalyst is always available at:

http://www.seabirdsoftware.com/dbAnalyst.exe

Previous versions are also available, and are typically named 'dbAnalyst x.y.z.w .exe'.

Registering dbAnalyst

A time-limited or permanent license is required to activate dbAnalyst. A time-limited license is obtained by registering at Seabird Software when downloading dbAnalyst. Note that a valid email is required to receive registration information.

Permanent license keys can be purchased at the product main page, at:

http://www.seabirdsoftware.com/products.php

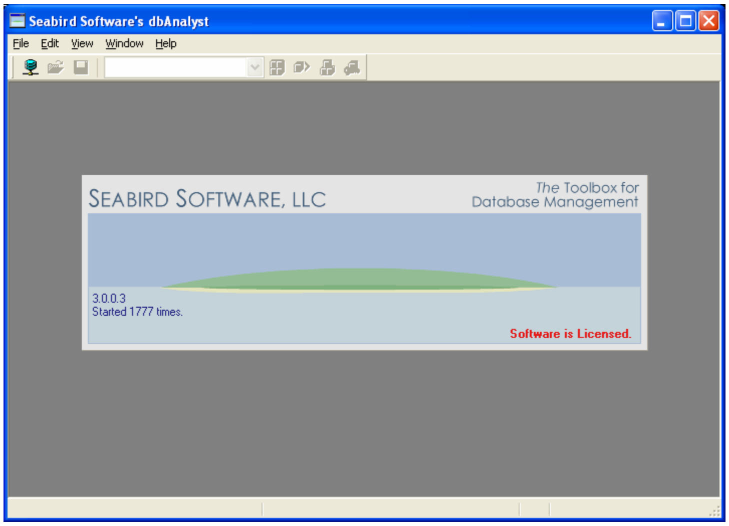
Please Note:

Seabird Software, LLC considers all information given by users to be confidential. Seabird Software does not share collected information in any manner with any other entity.

Seabird Software has contracted with Kagi, a third party e-commerce provider. Seabird Software only collects basic user information such as user name and email address, and does not have access to user financial information such as credit card numbers, etc.

Using dbAnalyst

Once installed, dbAnalyst will appear as below when started:


Seabird Software's dbAnalyst
File Edit View Window Help
SEABIRD SOFTWARE, LLC
The Toolbox for Database Management
3.0.0.3

Started 1777 times.

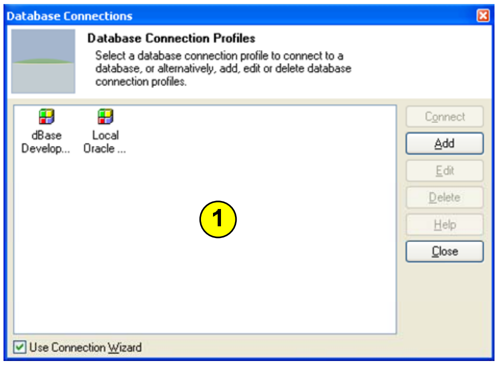
Software is Licensed.

Figure 1- dbAnalyst Initial Screen

The very next step consists of either connecting to a database: connecting to a database is done either by using the File/Connect, or Ctrl-N, or by clicking on the toolbar icon.

Connecting to a database

Selection of the File/Connect menu or the Ctrl-N, or clicking on the toolbar icon will bring about the Database Connection Profile Management modal window. This window allows connection to a database using existing connection information, or modification and management of such information.



Select a database connection profile to connect to a database, or alternatively, add, edit or delete database connection profiles.

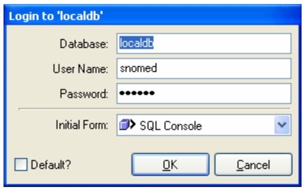
☑ Use Connection Wizard

Figure 2 - Database Connection Profile Management Window

The Database Connection Management modal window allows for database connection selection and management. This window either allows the use of previously established connection parameters to connect to the database, or the management of new database connection parameters.

To utilize previously established connection parameters, select a connection entry (①), and click on the Connect button, or double-click on the database entry. Upon connecting to a database using connection settings, several actions will occur:

a. Connection login window may be presented to the user. This login window allows modifying connection parameters. This window is only displayed if the 'Present a login prompt' check box is checked in Page 2 of the Connection Property Modal Window.



Login to 'localdb'

Database: localdb

User Name: snomed

Password: ••••••

Initial Form: SQL Console

□ Default?

Figure 3 - Connection Login Window

Optionally, the default check box allows for altering connection parameters from entry in this window.
Note that the connection Login window appearance (Figure 3) is based on the Database Type.

b. A connection to the selected database is attempted. If successful, the selected initial form is instantiated.

Note that the connection entry list view (①) can be changed by right-clicking on the list, by selecting the View menu on the popup menu, and by selecting from 'Large Icons', 'Small Icons', 'List', or 'Details' display types.

Creating a new Database Connection

Prior to establishing a connection to any database, several software components are required:

- The Microsoft Data Access Components (MDAC) and Microsoft XML 4.0 services.
  Both components are checked, upgraded or installed when installing dbAnalyst on the host computer.

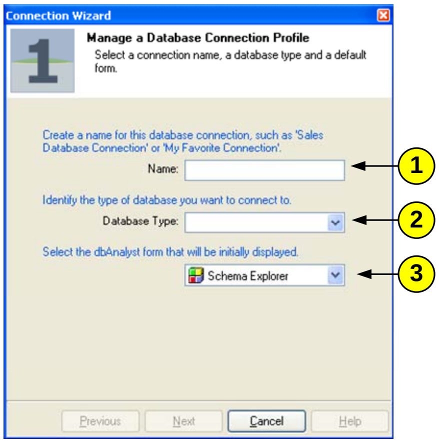
- Database client software is installed on the client computer. Client software must either include the ADO database OLE provider, or the ODBC database client software, as well as vendor specific libraries and components.

- A target database, accessible via networking, or on the current host computer is also needed. Please consult with your System Administrator to make sure that the target database is available.

Clicking on the Add button while the Database Connection Management modal window will start the Database Connection Profile Wizard. The database connection flow is a two-step process:

- the first step consists of naming a database connection, selecting the database handler type to connect to (i.e. Oracle, Microsoft SQL Server, IBM DB2, etc), and selecting the tool that will be opened by default,
- the second step consists of providing information specific to the database handler as specified in step 1.

STEP 1



Connection Wizard

Manage a Database Connection Profile

Select a connection name, a database type and a default form.

Create a name for this database connection, such as 'Sales Database Connection' or 'My Favorite Connection'.

Name:

Identify the type of database you want to connect to.

Database Type:

Select the dbAnalyst form that will be initially displayed.

Schema Explorer

Figure 4 - Connection Property Modal Window, Page 1

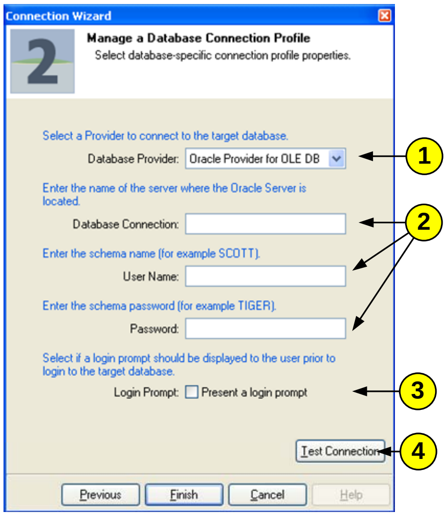
1 Connection Name.
The connection name is required for later use. This is a name created by the user, such as "My Favorite Connection."

2 Database Type
A database type must be selected. Database types consists of the different database handlers and versions supported by dbAnalyst. At this time, support is provided for Oracle, Microsoft SQL Server, Microsoft Access, Sybase iAnywhere, PostgreSQL, dBase, MySQL, IBM DB2, Caché, and Text Files. Future support is planned for Informix, Sybase Enterprise Database and IngresII.

3 Initial Tool.
Once the database connection is established, a dbAnalyst tool is selected and opened by default. Possible tool choices are the *Schema Explorer*, the *SQL Console* and the *Query Builder* and *Session Manager*. Choose the one you need from the drop-down list.

STEP 2

Step 2 is specific to the database handler specified in step 1. Complexity may range from selecting a folder (for dBase, and Text files) to providing host name, port number, user name and password, etc.



Manage a Database Connection Profile
Select database-specific connection profile properties.

Select a Provider to connect to the target database.
Database Provider: Oracle Provider for OLE DB

Enter the name of the server where the Oracle Server is located.
Database Connection:

Enter the schema name (for example SCOTT).
User Name:

Enter the schema password (for example TIGER).
Password:

Select if a login prompt should be displayed to the user prior to login to the target database.
Login Prompt: □ Present a login prompt

Figure 5 - Connection Property Modal Window, Page 2

1 ADO/ODBC Provider Reference.
A Database vendor, and third party software vendors, may provide many different means of connection to a database, using different ‘providers’. A provider is the software layer between MDAC and the database connected to, while dbAnalyst connect to MDAC.
Select a provider that will fit your particular configuration.

2 Database Specific Elements.
This section of the form is dependant over the database handler, as specified by Seabird Software.

3 Login Prompt.
When connecting to a database, dbAnalyst provides the user the opportunity to change connection parameters before establishing the connection to the database, by providing a login modal window.

4 Connection Test.
Once parameters have been selected, it is quite useful to check that a connection can be established by clicking on the ‘Test Connection’ button. A status message will be provided back to the user with (hopefully) useful information.

Editing a Database Connection

To edit an existing database connection parameter, click the icon, select the connection you wish to edit, and click the edit button. This will open the database connection property wizard. Select the parameters you wish to alter and proceed with changing them.

Deleting a Database Connection

To delete a database connection, select an entry in the database connections modal window and then click “delete.”

Technical Note

All database connection information is stored into a single XML file, named ‘Connection.xml’, located into the same folder that the dbAnalyst binary file.

Introducing the dbAnalyst Toolbox

dbAnalyst is designed to meet several needs: from providing a quick overview of an overall database, to providing the necessary tools to query and modify a database, to eventually being able to create complex queries with an intuitive user interface.

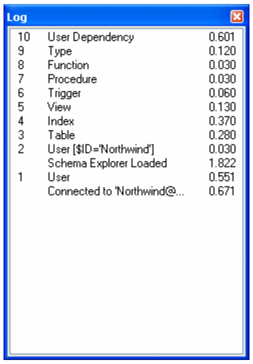
Once a database connection is established, choose the tool you wish to use. Possible choices include the *Schema Explorer*, *SQL Console*, *Query Builder* and *Session Manager* tools. After a connection is established, one can change tools by either:

- Using the *Navigation Palette*
  The Navigation palette is located on the desktop toolbar, and shows the current list of database connections, and for each connection, opened tools, while also allowing selection and opening of new tools.
- Using the Tools menu
- Using shortcuts
  Alt-1 for The *Schema Explorer.*
  Alt-2 for the *SQL Console.* (Note that a single database connection can support one or several *SQL Console* tools.)
  Alt-3 for the *Query Builder*
  Alt-4 for the *Session Manager.*

Palettes

Several modeless tool palettes are available for assistance:

- The *SQL Log Palette*
  The SQL Log palette window shows all SQL statements that have been executed against the current database connection, since the connection was established. This log includes all operations, including *Schema Explorer* SQL statements, *SQL Console* statements and *Query Builder* statements. Double-clicking on a statement will re-execute the statement in the *SQL Console* tool.



```
10    User Dependency                0.601
9     Type                           0.120
8     Function                       0.030
7     Procedure                      0.030
6     Trigger                        0.060
5     View                           0.130
4     Index                          0.370
3     Table                          0.280
2     User [$ID='Northwind']         0.030
      Schema Explorer Loaded         1.822
1     User                           0.551
      Connected to 'Northwind@...    0.671
```

Figure 6 – SQL Log Palette

- The 'Memorize SQL Statements' Palette.
  This palette shows all SQL statements that have been 'memorized'. SQL statement memorization can only occurs within the SQL Console. SQL statements executed during reverse engineering operations are not memorized. Memorization occurs under two distinct circumstances:
  - By selecting the SQL Console, Action Menu, memorize Statement menu item, or pressing Ctrl-M, only selected SQL statements are memorized. This menu item is disabled if the 'Memorize All Statements' check box, Preference menu, SQL Console tab is check on.
  - By checking on the 'Memorize All Statements' check box, Preference menu, SQL Console tab, all statements are memorized when executed.

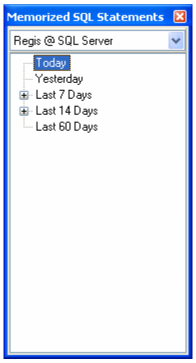
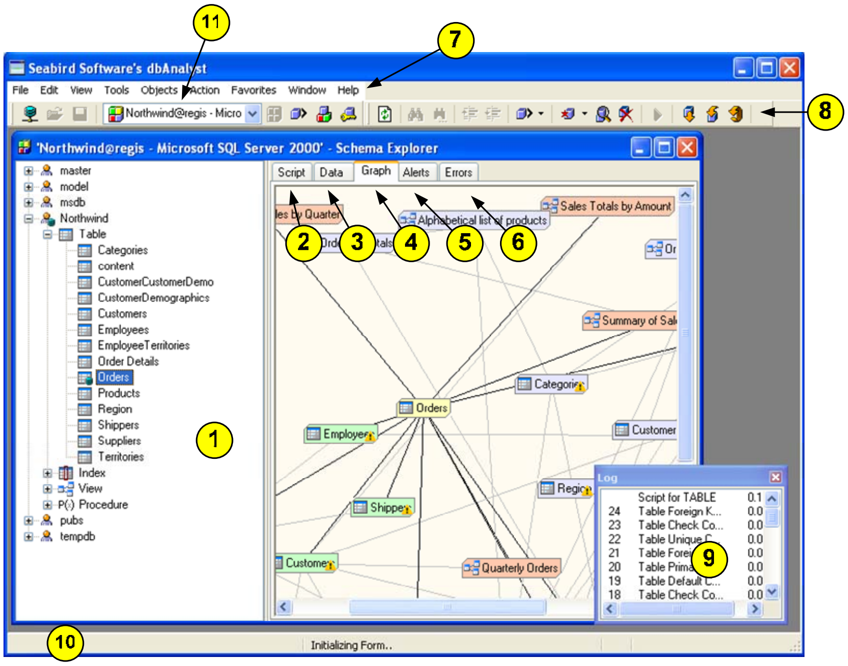


Figure 7 – 'Memorize SQL Statements' Palette

The dbAnalyst *Schema Explorer*

Introduction

The *Schema Explorer* is the tool that will reverse engineer or extract information from a database, and present that information in the most intuitive, readable form. Figure 8 shows an example of a reverse-engineered database.



```
     Script for TABLE     0.1
24   Table Foreign K...   0.0
23   Table Check Co...    0.0
22   Table Unique C...    0.0
21   Table Forei          0.0
20   Table Prima          0.0
19   Table Default C...   0.0
18   Table Check Co...    0.0
```

Figure 8 - Overview of the Schema Explorer

The *Schema Explorer* window consists of the following:

1 The *database objects hierarchy*.
All database objects are referenced in context of type, in alphabetical fashion. Selecting an object will drive the window right hand side display. Objects are represented with an icon and a name. The icon represents the object class; this icon may be altered to represent that the object has been explored, or that the object has alerts.

2 The *Script View*.
This panel shows text generated by the reverse engineering operation. Particularly, Syntax-highlighted *create* scripts, or *drop* scripts will be displayed in this panel.
Note that script generation elements are provided by the database XML script. Different script types are exclusively provided by the XML database script.

Switching to a different view (3 , 4, 5) will present further information related to the same object.

③ The *Data View*.
This panel shows data content from the object selected in the database objects hierarchy. Data is presented is as spreadsheet.

Switching to a different view (②, ④, ⑤) will present further information related to the same object.

④ The *Database Schema Graph View*.
This panel shows the schema as a network layout graph. Selecting an object will:
1. Highlight the object in the *database objects hierarchy*,
2. Highlight the object on the graph using the selected object color,
3. Highlight child and parent objects respectively using the child and parent colors.

Switching to a different view (③, ④, ⑤) will present further information related to the same object.
(Colors mentioned above can be modified under File/Preferences menu.)

⑤ The *Alerts View*.
This panel shows database alerts generated from rules that were processed after reverse engineering the database. Common alerts include disabled objects, such as triggers and foreign key constraints, and out-of-space errors such as database data files reaching size limits. Selecting an alert may cause more information about the alert to be shown, as well as SQL statements that may help resolve the alert. Note that alert generation elements are provided by the Database Definition Source XML script.

Switching to a different view (②, ③, ④) will present further information related to the same object.

⑥ The *Errors View*.
This panel shows errors that might have occurred during the reverse engineering phase, or during script parsing and rule-based alert generation. This panel is targeted toward the needs of a database script writer.

⑦ The *Context Sensitive Menus.*
Menus are exclusively directed at operations available within the *SQL Console* tool. Most menu items are enabled in context to user actions.

⑧ The *Navigation Palette.*
The Navigation palette shows the current list of database connections, and for each connection, opened tools, while also allowing selection and opening of new tools.

⑨ The *SQL Log.*
The SQL Log palette window shows all SQL statements that have been executed against the current database connection, since the connection was established. This log includes all operations, including *Schema Explorer* SQL statements, *SQL Console* statements and *Query Builder* statements. Double-clicking on a statement will re-execute the

statement in the *SQL Console* tool.

10 The *Status Bar.*
The statement bar shows run-time information, including the execution status of the reverse-engineer phase, cursor position, etc..

11 The *Navigation Palette.*
The Navigation palette shows the current list of database connections, and for each connection, opened tools, while also allowing selection and opening of new tools.

The next sections provide detailed information about the *Schema Explorer*.

Schema Reverse Engineering

Once connected to, schemas are dynamically explored[2] as the user expand the database object hierarchy. The SQL Log window shows all SQL statements that have been executed against the database connection. Portions of the schema may be refreshed using the F5 refresh function. Objects that been 'explored' show a modified icon including a database platter mini-icon (■).

Features

a. Objects Hierarchy Generation

The Database Object Hierarchy is generated from statements provided by the Database Definition Source XML file. In effect, dbAnalyst does not 'know' about the database and only processes statements to extract an object hierarchy from within the database. In addition, the script defines dependency relationships, which will be primarily used to create the database schema graph view.

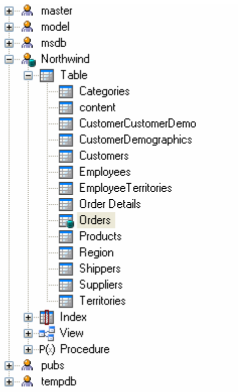


Figure 9 - Example of Database Object Hierarchy

[2] As opposed to previous versions, file caching is no longer occurring.

Refresh

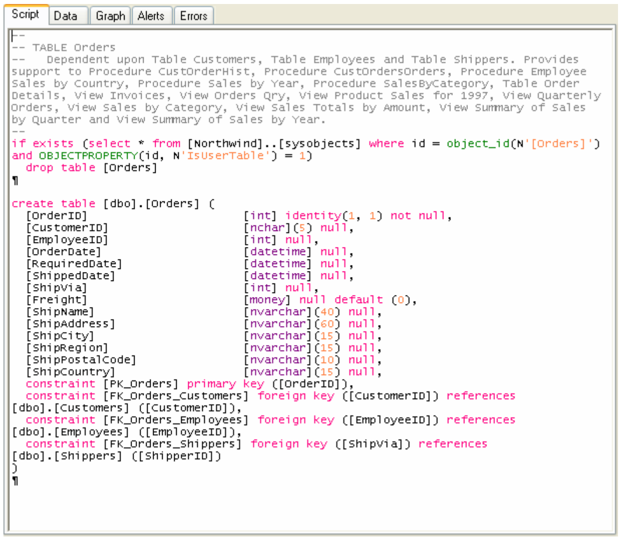
Any part of the database can and will be altered after the database is reverse engineered. Therefore, the *Schema Explorer* should allow refreshing parts of the database object hierarchy, at any time, and as often as necessary.

Any part of the object hierarchy can be refreshed to reflect the current state of the database connection, by either accessing the Edit/Refresh menu item, or the F5 shortcut key, or the toolbar icon.

Note that the graph and alerts views will be refreshed as well, to reflect the updated state of the database.

b. Script Generation

Object Script generation is exclusively driven by the ***Database Definition Source XML file***. Selecting an object in the Database Object Hierarchy will trigger object script generation. Clicking on a table name, for example, will display the create statement (by default) in the Script view. Figure 10 shows such text generation. Clicking on the Table heading will cause dbAnalyst to compute and display the script for all tables, sorted by table dependencies[3].



```
--
-- TABLE Orders
--   Dependent upon Table Customers, Table Employees and Table Shippers. Provides
support to Procedure CustOrderHist, Procedure CustOrdersOrders, Procedure Employee
Sales by Country, Procedure Sales by Year, Procedure SalesByCategory, Table Order
Details, View Invoices, View Orders Qry, View Product Sales for 1997, View Quarterly
Orders, View Sales by Category, View Sales Totals by Amount, View Summary of Sales
by Quarter and View Summary of Sales by Year.
--
if exists (select * from [Northwind]..[sysobjects] where id = object_id(N'[Orders]')
and OBJECTPROPERTY(id, N'IsUserTable') = 1)
  drop table [Orders]
¶

create table [dbo].[Orders] (
  [OrderID]                         [int] identity(1, 1) not null,
  [CustomerID]                      [nchar](5) null,
  [EmployeeID]                      [int] null,
  [OrderDate]                       [datetime] null,
  [RequiredDate]                    [datetime] null,
  [ShippedDate]                     [datetime] null,
  [ShipVia]                         [int] null,
  [Freight]                         [money] null default (0),
  [ShipName]                        [nvarchar](40) null,
  [ShipAddress]                     [nvarchar](60) null,
  [ShipCity]                        [nvarchar](15) null,
  [ShipRegion]                      [nvarchar](15) null,
  [ShipPostalCode]                  [nvarchar](10) null,
  [ShipCountry]                     [nvarchar](15) null,
  constraint [PK_Orders] primary key ([OrderID]),
  constraint [FK_Orders_Customers] foreign key ([CustomerID]) references
[dbo].[Customers] ([CustomerID]),
  constraint [FK_Orders_Employees] foreign key ([EmployeeID]) references
[dbo].[Employees] ([EmployeeID]),
  constraint [FK_Orders_Shippers] foreign key ([ShipVia]) references
[dbo].[Shippers] ([ShipperID])
)
¶
```

Figure 10 – Example of Table Create Script Generation

The create script is the default text output; our experience shows that it is the preferred output by database developers. Script types are exclusively driven by the ***Database Definition Source XML file***. Other usual script types include ***drop*** and ***delete*** scripts. Accessing the Objects/Scripts menu item, or right clicking on the Script view (Figure 10), causes display of the

[3] A database object dependency consists of an object relying onto another object for creation, existence, and/or for execution. This is notably the case for tables and foreign key constraints having dependencies over tables with primary key constraints, or triggers providing services to tables, or again stored procedures requiring sets of database objects. In effect, a relational database is a graph: it is rather meaningless to consider objects individually, while it makes so much more sense to consider objects as part of a *continuum*.

available scripts for the current database object. Note on Figure 11 that the *Instance Data XML* menu items, if selected, will display the data that was reversed engineered from the database.

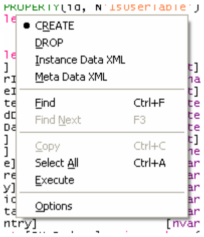


Figure 11 – The Script view popup menu.

Script Syntax Highlighting

Text generation also includes syntax highlighting capability, defined by the ***Database Definition Source XML file.*** Syntax highlighting consists of different colors for comments, SQL keywords, data types, and functions. Syntax highlighting colors are defined as user preferences, accessible under the File/Preference menu item.

Script Text processing functions

Any selected (highlighted) part of the generated script can be copied (either Edit/Copy menu item, or Ctrl-C shortcut, or the 📋 toolbar item) to the clipboard, or can be searched using Find (either Edit/Find menu item, or Ctrl-F shortcut, or the 🔍 toolbar item) and Find Next (either Edit/Find Next menu item, or the F3 shortcut, or the 🔍 toolbar item) functions.

Script Statement Execution function

Any selected part of the generated script can be executed as a SQL statement in the *SQL Console* window. To do so, select text from the Script view, and to execute the text statement, using either the Edit/Execute menu item, or the Ctrl-E shortcut, or the ▶ toolbar item.

c. Data View

d. Graph View

The database reverse engineering phase produces a set of objects that are related by dependencies. In Figure 12, “orders” is the selected object, in yellow. The green objects, “Shippers”, “customers” and “Employees” are considered to be parents of “Orders,” meaning that “Orders” is dependent upon them. The orange objects are all children of orders because they are dependent upon the “Orders” object. Blue objects are not related to the “Orders” object. This graph conveniently represents the database as a two-dimensional network layout. (Figure 12).

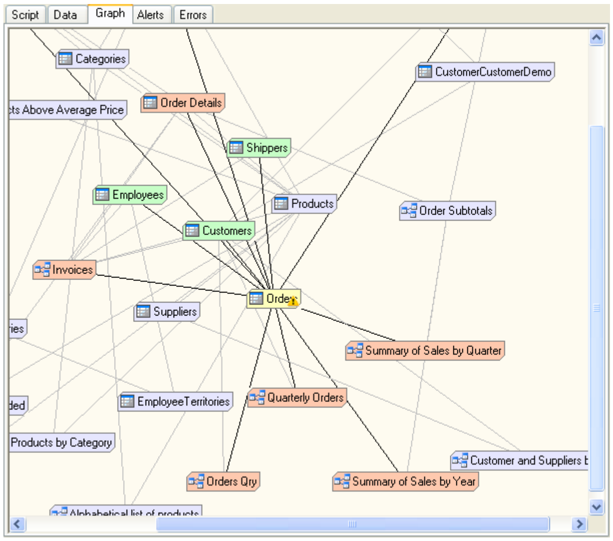


Figure 12 – Example of Graph.

Additional information is also pictured on each graph node: the type of database object represented as an icon on the left side of the node, and possibly alert icons on the right hand side.

Selecting an object into the Database Object Hierarchy will select the corresponding graph object, or node, if such object is graphed. Vice versa, clicking on a graph node will select the Database Object Hierarchy corresponding object (Figure 13).

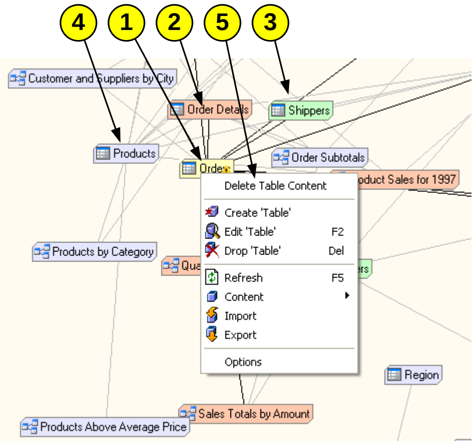


Figure 13 – Graph Node Types.

1 Selected Node[4].

2 Selected Node's Child node[2].

3 Selected Node's Parent node[2].

[4] This color can be set to user preferences, using the File/Preference menu item.

4 Unselected Node[2].

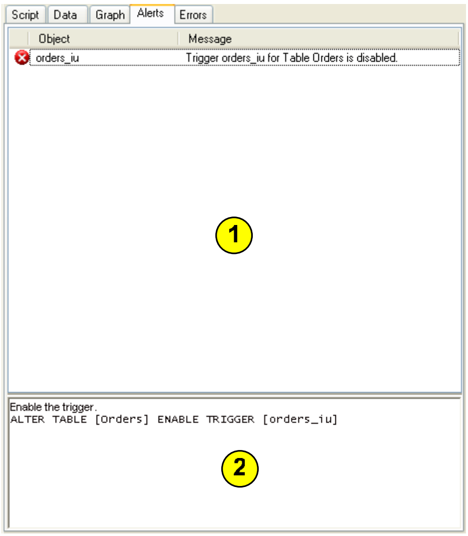
5 Right-click popup.
Right clicking on a node will display the node popup. This popup provides:
1. Context sensitive commands, specifically related to the object selected.
2. Generic commands, such as graph options, object content display, and import & export operations.

Moving Graph Objects

Graph object location can be changed. To do so, click on the object, do not release the mouse button, and move the mouse. The object location will follow the cursor. Release the mouse button to 'settle' the object location.

e. Database Alerts

Alerts are generated at the completion of the reverse engineering phase. Alerts are the result of rule processing that occurs when a database is reversed engineered. Figure 14 shows a simple example of Alert output.



Object	Message
orders_iu	Trigger orders_iu for Table Orders is disabled.

Enable the trigger.

```
ALTER TABLE [Orders] ENABLE TRIGGER [orders_iu]
```

Figure 14 - Example of Alert view.

1 The alert line output. An alert is directed to a specific database object. Alerts have a priority level, a targeted object, and a message containing a short description.

2 For each alert, a longer description can also be supplied, describing in detail resolution, bibliography source, etc, as well as possibly the SQL statement that will resolve the alert.

f. Object Edition

Object edition consists of creating new objects, modifying or deleting existing objects. Objects that can be created or modified include tables, sequences, functions, procedures, triggers, indexes, views, etc. Availability or behavior of object creation or edition is highly dependent on database vendor and database version: each database version, in combination with OLEDB drivers introduces new issues and capability. *Appendix A* shows dbAnalyst capability matrix, database by database, and version by versions, for combinations of database and database versions supported by dbAnalyst.

Toolbar and Popup Menus

For consistency reasons, dbAnalyst presents editing of similar objects for different databases in a similar fashion, thus hiding as much as possible edition disparities among databases.

To create an object, either select the "Objects/Create Object" menu item and select the object type, or right click on the object and select the "Create 'x'" popup menu item, or click on the toolbar create object button.

To edit an object, select the object, and either select the "Objects/Edit 'x'" menu item, or press F2, or right click on the object and select the "Edit 'x'" popup menu item, or click on the toolbar object edit button.

To delete an object, select the object, and either select the "Objects/Drop'x'" menu item, or press Del, or right click on the object and select the "Drop 'x'" popup menu item, or click on the toolbar object drop button.

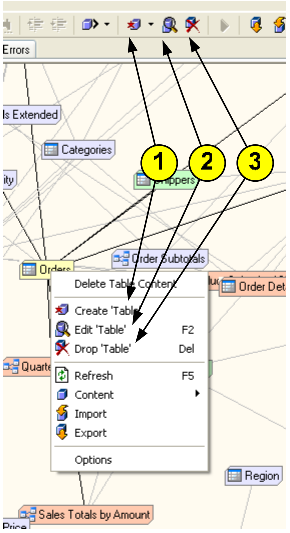


Figure 15 - Edition Toolbar and Popup Menus

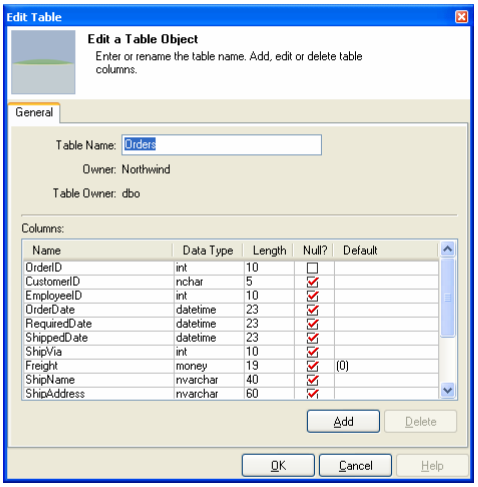
Object creation and edition concepts

Creating or editing any object will bring about a modal window that will allow for setting or altering object parameters. The behavior of such window is set by the definition source, for each database object, as a script.

Validating data entry will execute one or more SQL statements to reflect structural changes. Any failed statement will stop any further processing.

For example, creating or editing a table is kept as straightforward as possible:

- The table name can be set or changed;
- Table Columns can be added, renamed or removed. Column data types, data type length, null-able or default value can be revised.



Edit a Table Object

Enter or rename the table name. Add, edit or delete table columns.

Table Name: Orders

Owner: Northwind

Table Owner: dbo

Columns:

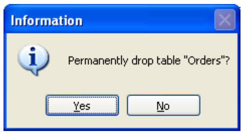
Name	Data Type	Length	Null?	Default
OrderID	int	10	☐	
CustomerID	nchar	5	☑	
EmployeeID	int	10	☑	
OrderDate	datetime	23	☑	
RequiredDate	datetime	23	☑	
ShippedDate	datetime	23	☑	
ShipVia	int	10	☑	
Freight	money	19	☑	(0)
ShipName	nvarchar	40	☑	
ShipAddress	nvarchar	60	☑	

Figure 16 - Table Edition

Appendix A shows dbAnalyst capability matrix for managing database objects database-by-database, version-by-version.

Object deletion concepts

Attempting to delete any object will bring about a message box prompting the user to confirm the object deletion.



Permanently drop table "Orders"?

Figure 17 - Example of Object Deletion

Deleting an object might present difficulties such as existence of foreign constraints; under such circumstances, deleting the object will result in refusal from the database. The error message turned by the database will be presented as such as a message box to the user.

Supported Objects

Object types supported for creation, edition across a wide range of databases include tables, views, triggers, functions, procedures, and packages. Object types that are not supported for dbAnalyst v3.0 include users, constraints, indexes and grants. Support for such object types is planned for v3 upcoming point releases.

All object types are supported for deletion.

g. Import & Export Operations

The *Export & Import operations* goal is to be able to move database schema content from one schema to another schema, possibly across database handlers (I.e. database Vendors). The Export & Imports operations are not to be confused with the File/Save operation that will only save the schema structure as an XML file.

The export operation produces an export file that can be imported to any of the dbAnalyst-supported database types. For example, one could export tables from dBase, and import these into Microsoft Access; if the target tables do not exist in the target Access database, then dbAnalyst will create the tables.

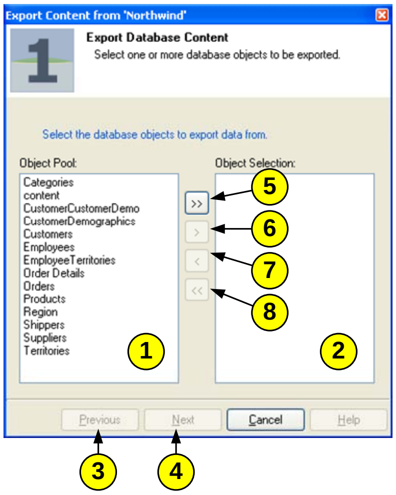
ⓘ *dbAnalyst 3.x export file format has been revised and is not backward compatible with 2.x versions.*

Export Operation

The export operation is started by accessing either the Tools/Export menu item, or the toolbar icon.

Exporting a database schema is a three-step process:

1. SELECT THE OBJECTS THAT WILL BE EXPORTED.



Export Content from 'Northwind'

Export Database Content

Select one or more database objects to be exported.

Select the database objects to export data from.

Object Pool:

Categories
content
CustomerCustomerDemo
CustomerDemographics
Customers
Employees
EmployeeTerritories
Order Details
Orders
Products
Region
Shippers
Suppliers
Territories

Object Selection:

Figure 18 – First step of the Export operation.

1 The list of objects (*Object Pool*) whose content can be exported. Any object on this list can be selected to be exported by clicking on 5 or 6.

2 The list of objects (*Object Selection*) that will be exported. Any object on this list can be selected to be removed from the object export list by clicking on 7 or 8.

3 Use this button to go to the previous Export Step. This button is always disabled during the first step.

4 Use this button to go to the next Export Step. This button will lead to the second step of the export process. This button is only enabled if objects are present in the *Object Selection* list.

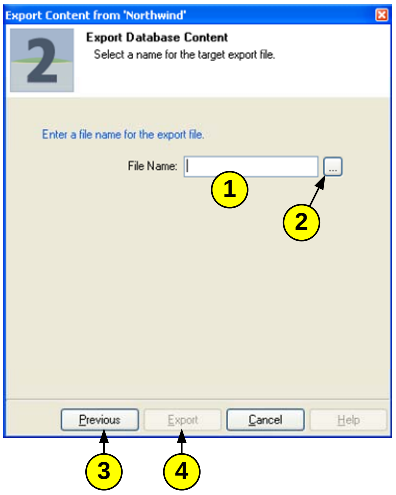
5 Click on this button to move *all* objects from the *Object Pool* list to the *Object Selection* list.

6 Click on this button to move a selected object from the *Object Pool* list to the *Object Selection* list. To select a unique object, click on the object using the mouse. To select multiple objects, use the Ctrl key (unique object selections) or Shift key (range object selection) in addition to the mouse.

7 Click on this button to move selected object from the *Object Selection* list to the *Object Pool* list.

8 Click on this button to move *all* objects from the *Object Selection* list to the *Object Pool* list.

2. SELECT A FILE NAME FOR THE EXPORT FILE



Export Content from 'Northwind'

Export Database Content

Select a name for the target export file.

Enter a file name for the export file.

File Name:

Figure 19 – Second step of the Export operation.

1 The file name. The name can be either typed by hand, or selected by clicking 2.

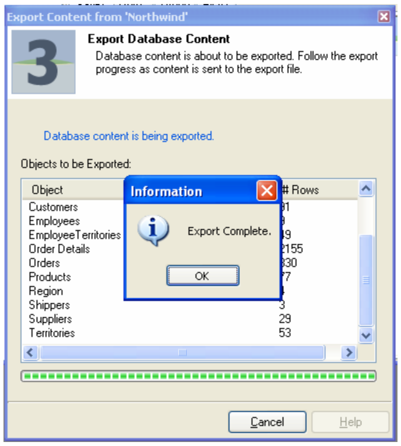
2 Click on this button to select a folder and file name. dbAnalyst will pop up a standard *File Save* modal dialog box.

3 Click this button to go to the previous Export Step.

4 Click this button to go to the next Export Step. This button will lead to the third step of the export process. This button is only enabled if a valid file name is present.

3. Proceed to Export

The third step consists of the export operation itself. Before the export operation, the user will be prompted to confirm the export operation, and if confirmed, the export operation will proceed.



Export Content from 'Northwind'

Export Database Content

Database content is about to be exported. Follow the export progress as content is sent to the export file.

Database content is being exported.

Objects to be Exported:

Object	# Rows
Customers	1
Employees	
EmployeeTerritories	9
Order Details	2155
Orders	330
Products	77
Region	
Shippers	3
Suppliers	29
Territories	53

Information

Export Complete.

Figure 20 - Third step of the Export operation.

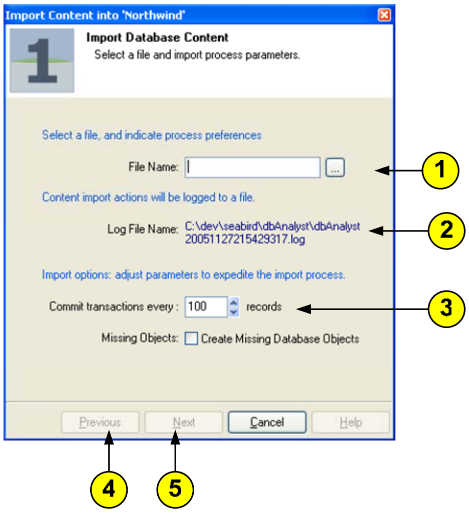
Import Operation

The import operation is started by accessing either the Tools/Import menu item, or the toolbar icon.

Importing content from a database export file first requires selecting a valid schema export file:

The import process involves four steps.

1. Select database Content Import Options.



Import Content into 'Northwind'

Import Database Content

Select a file and import process parameters.

Select a file, and indicate process preferences

File Name:

Content import actions will be logged to a file.

Log File Name: C:\dev\seabird\dbAnalyst\dbAnalyst 20051127215429317.log

Import options: adjust parameters to expedite the import process.

Commit transactions every : 100 records

Missing Objects: □ Create Missing Database Objects

Figure 21 - Selecting Import Options.

1 The file name of the export file. This value is required to proceed with the import operation.

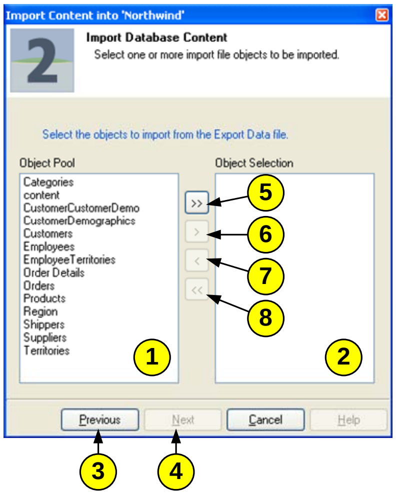
2 The log file name: all operations will be recorded into a log file, for later user analysis if necessary.

3 The *Transaction Size* option. To balance speeding up the import process, without overflowing the database server with pending transactions, a threshold number must be provided.

4 Go to the previous Import Step. This button is always disabled in the first step.

5 Go to the next Import Step.

2. Select the objects that will be Imported.



Import Content into 'Northwind'

Import Database Content

Select one or more import file objects to be imported.

Select the objects to import from the Export Data file.

Object Pool: Categories, content, CustomerCustomerDemo, CustomerDemographics, Customers, Employees, EmployeeTerritories, Order Details, Orders, Products, Region, Shippers, Suppliers, Territories

Object Selection

Figure 22 - Second step of the Import operation.

1 The list of objects (*Object Pool*) whose content can be imported. Any object on this list can be selected to be imported by clicking on 5 or 6.

2 The list of objects (*Object Selection*) that will be imported. Any object on this list can be selected to be removed from the object export list by clicking on 7 or 8.

3 Click this button to go to the previous Export Step.

4 Click this button to go to the next Import Step. This button will lead to the second step of the import process. This button is only enabled if objects are present in the *Object Selection* list.

5 Click on this button to move *all* objects from the *Object Pool* list to the *Object Selection* list.

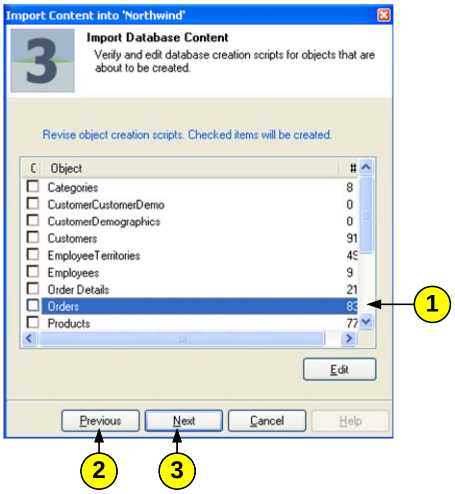
6 Click on this button to move selected object from the *Object Pool* list to the *Object Selection* list. To select a unique object, click on the object using the mouse. To select multiple objects, use the Ctrl key (unique object selections) or Shift key (range object selection) in addition to the mouse.

7 Click on this button to move selected object from the *Object Selection* list to the *Object Pool* list.

8 Click on this button to move *all* objects from the *Object Selection* list to the *Object Pool* list.

3. Review and Modify Object Creation

Step 3 allows the review and modification of the create scripts used to create missing objects, if necessary.



Import Content into 'Northwind'

Import Database Content

Verify and edit database creation scripts for objects that are about to be created.

Revise object creation scripts. Checked items will be created.

C	Object	#
☐	Categories	8
☐	CustomerCustomerDemo	0
☐	CustomerDemographics	0
☐	Customers	91
☐	EmployeeTerritories	49
☐	Employees	9
☐	Order Details	21
☐	Orders	83
☐	Products	77

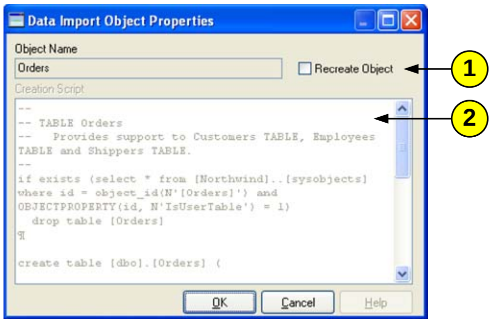
Figure 23 - Third step of the Import operation.

1 The list of objects that will be imported. The first column, if checked, indicates if the object will be created. The second column is the name of the object. The third column is the number of rows that will be created.

2 Click on this button to move selected object from the *Object Selection* list to the *Object Pool* list.

3 Click on this button to move *all* objects from the *Object Selection* list to the *Object Pool* list.

Objects might require creation. In this case, the check box at the left of the object name will be checked. The user can modify the object creation script by double clicking the object name.



Object Name: Orders

□ Recreate Object

```
--
-- TABLE Orders
--   Provides support to Customers TABLE, Employees
TABLE and Shippers TABLE.
--
if exists (select * from [Northwind]..[sysobjects]
where id = object_id(N'[Orders]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
  drop table [Orders]
¶

create table [dbo].[Orders] (
```

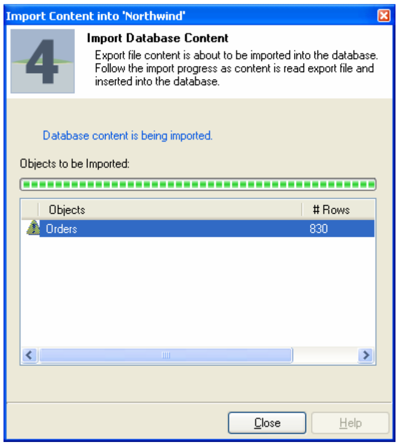
Figure 24 - Review and modify an object creation script.

1 The object name, as well as the option to recreate the object. This check box is enabled if the object already exists; checking the box will enable the creation script, so that the user can modify the script. This check box is disabled if the execution of the create script is required (i.e. the object is missing). The Creation Script is enabled.

2 The Object Creation script, either enabled or disabled.

4. Proceed to Import

The fourth step is the import operation. Before import, the user will be prompted to confirm the import operation, and if confirmed, the import operation will proceed.



Import Content into 'Northwind'

Import Database Content

Export file content is about to be imported into the database. Follow the import progress as content is read export file and inserted into the database.

Database content is being imported.

Objects to be Imported:

Objects	# Rows
Orders	830

Figure 25 - Review and modify an object creation script.

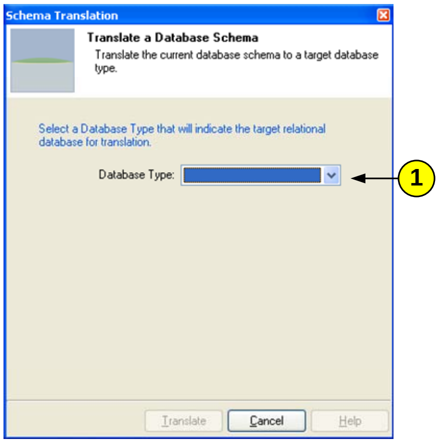
Once complete, the object list shows individual objects, the number of rows initially set for import, and the final count of imported rows. If an icon is present at object left, review the log file to gain knowledge of import errors.

h. Database Script Translation

Database translation is the translation of the current reversed engineered schema to another database. During the translation process, mappings are established between the source and target database definition sources, and then the reverse-engineered data is migrated. Definition Source Script mappings include database specific attributes translation, such as data types, and other attributes.

TRANSLATION OPERATION

The import operation is started by accessing either the Tools/Translate menu item, or the toolbar icon.



Schema Translation

Translate a Database Schema

Translate the current database schema to a target database type.

Select a Database Type that will indicate the target relational database for translation.

Database Type:

Figure 26 - Selecting a target database for translation.

1 Select the target database for translation. The list of database targets is dependant on the list of *Definition Source* files present in the same folder as the dbAnalyst binary executable.

RESULTING OUTPUT

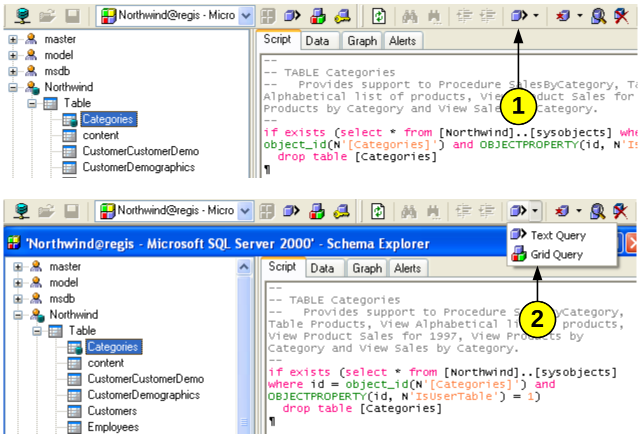
The translation result is the creation of a new *Schema Explorer* window; all attributes are related to the new database definition source.

Note that this new window does not have *yet* a database connection; any attempt to execute SQL statements – for example creating objects – will result into the *Database Connection* window being presented.

i. Content Display

The content display operation is started by accessing either the Objects/Content Text Query or Grid Query menu items, or the / toolbar icons. The content icon or menu is only enabled if the object selection allows for content display.

Content display will either occur into the *SQL Console* or in the *Query Builder*. The SQL statement necessary to display content is automatically built and is based on the characteristics of the selected object.



```
--
-- TABLE Categories
--   Provides support to Procedure SalesByCategory, Ta
Alphabetical list of products, Vie     duct Sales for
Products by Category and View Sale     Category.
--
if exists (select * from [Northwind]..[sysobjects] whe
object_id(N'[Categories]') and OBJECTPROPERTY(id, N'Is
  drop table [Categories]
¶
```

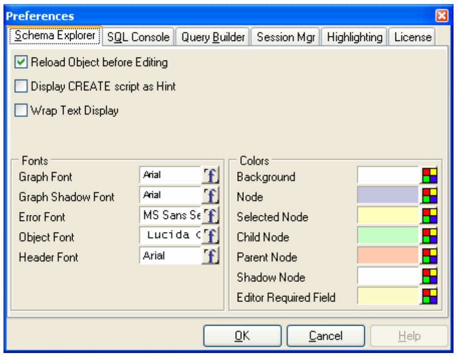
```
--
-- TABLE Categories
--   Provides support to Procedure S     yCategory,
Table Products, View Alphabetical li     products,
View Product Sales for 1997, View Pro     ts by
Category and View Sales by Category.
--
if exists (select * from [Northwind]..[sysobjects]
where id = object_id(N'[Categories]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
  drop table [Categories]
¶
```

Figure 27 - Selecting the Content Display Type.

Selection of the Content display type consists of clicking on the down arrow (1) and choosing the content display type (2).

Options

Preferences for the Schema Explorer tool are located under the File Menu, Preference Menu Item, as a modal dialog window (Figure 28).



☑ Reload Object before Editing

☐ Display CREATE script as Hint

☐ Wrap Text Display

Graph Font: Arial
Graph Shadow Font: Arial
Error Font: MS Sans Se
Object Font: Lucida C
Header Font: Arial

Background
Node
Selected Node
Child Node
Parent Node
Shadow Node
Editor Required Field

Figure 28 – Preference Display, Schema Explorer Tab.

Preferences consist of:

- Reload Object Before Editing.
  This option indicates if objects should be reloaded from the database, before the Object Edition modal window appears.

- Display CREATE script as Hint.
  This option enabled displaying the object CREATE script as a graph panel hint.

- Wrap Text Display.
  This option enables wrapping of the text displayed in the script tab.

- Fonts
  - Graph Font.
    The font used on the Graph Object Panel Display.
  - Graph Shadow Font.
  - Error Font.
    The font used on the Error Tab.
  - Object Font.
    The font used on the Script Tab, for script display.
  - Header Font.
    The font used on the Script Tab, for object header display.

- Colors
  - Background Color.
    The color of the graph background.
  - Node Color.
    The color of unselected graph panels.
  - Selected Node Color.
    The color of a selected graph panel.

- Child Node Color.
  The color of child panels of a selected graph panel.

- Parent Node Color.
  The color of parent panels of a selected graph panel.

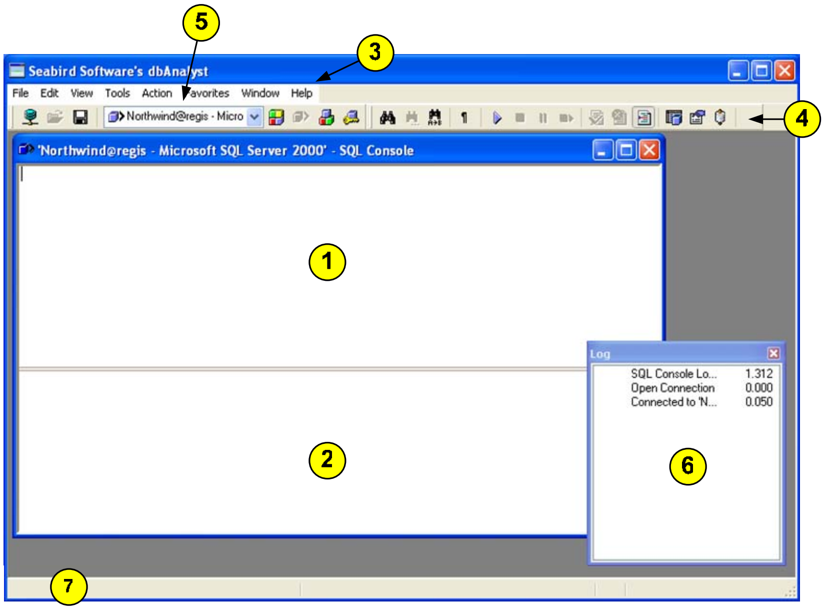
- Shadow Node Color.
  The color of panel-to-panel linkages.

- Editor Required Field Color.
  The color of a required field.

The dbAnalyst *SQL Console*

Introduction

The *SQL Console* allows the execution of SQL statements, for both modifying the database structures and for viewing database data. Structure-modifying statements can be used to alter structures and add, update or delete data.


5
3
Seabird Software's dbAnalyst
File Edit View Tools Action Favorites Window Help
Northwind@regis - Micro
4
'Northwind@regis - Microsoft SQL Server 2000' - SQL Console
1
Log
2
6
7

SQL Console Lo...	1.312
Open Connection	0.000
Connected to 'N...	0.050

Figure 29 - Overview of the SQL Console

The *SQL Console* window consists of the following:

1 The *SQL Command* Panel.
This panel is reserved for the user to input SQL commands.

2 The *Query Result* Panel.
This panel displays the results of SQL commands executed within the SQL command panel. The Query result panel can either display information as console-type (or text-based output) or grid-type output.

3 The *Context Sensitive Menus.*
Menus are exclusively directed at operations available within the *SQL Console* tool. Most menu items are enabled in the context of the user's actions.

4 The *Toolbar.*
Toolbar items represent shortcuts to selected menu items; As with menu items, toolbar items are enabled in the context of the user's actions

5 The *Navigation Palette.*
The Navigation palette shows the current list of database connections, and for each connection, opened tools, while also allowing selection

and opening of new tools.

6 The *SQL Log.*
The SQL Log palette window shows all SQL statements that have been executed against the current database connection, since the connection was established. This log includes all operations, including *Schema Explorer* SQL statements, *SQL Console* statements and *Query Builder* statements. Double-clicking on a statement will re-execute the statement in the *SQL Console* tool.

7 The *Status Bar.*
The *Status Bar* shows run-time information, including the *SQL Command* Panel cursor position, etc.

The next sections provide detailed information about the *SQL Console* main elements.

> The *SQL Console may* create transactions that could conflict with other transactions pending against the same database connection. Most relational databases do not support concurrent and/or nested transactions; for this reason, dbAnalyst always creates a new database connection for each new *SQL Console* window, with identical parameters to the initial database connection, in order to guarantee transaction isolation.
>
> In other words, it is possible to open two *SQL Console* instances against the same connection (or seemingly so), and to create transactions in each *SQL Console* without having transaction overlaps between *SQL Console* and thus guaranteeing transaction isolation

Features

SQL Statements: ; vs. ¶

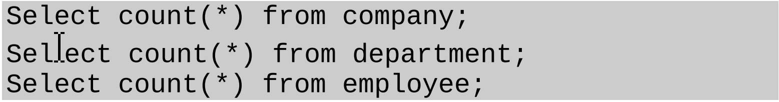
SQL Statements are exclusively entered in the SQL Command Panel. SQL stands for *Structured Query Language*, and consists either of instructions part of the structured query language Data Manipulation Language (DML) or Data Definition Language (DDL). A block of SQL may consist of one statement ended by a semicolon (;) or many statements, each statement ended by a semicolon (;), and the statement block ended by a paragraph mark (¶).

A paragraph is inserted into the SQL Command Panel by either using the Edit/Paragraph Mark menu item, the Alt+P shortcut, or the ¶ toolbar item.

Specifying a SQL Statement for execution

Many SQL statements may be present at once into the SQL Command Panel; a specific statement can be executed by:

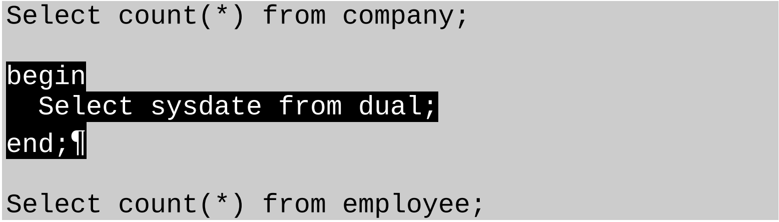
- By setting the cursor inside the text.
  In the next example, the cursor is set into the second statement (inside the select keyword). The second statement is delimited by the semicolon at the end of the first statement, and by the semicolon at the end of the second statement.

  ```
  Select count(*) from company;
  Select count(*) from department;
  Select count(*) from employee;
  ```

  Note that if the cursor is located after the last semicolon, then the last statement is targeted for execution.

- By selecting a block of text.
  In the next example, text selection includes several statements that may or may not be terminated by a semicolon.
  A block of SQL statements may consist of several DDL or DML distinct statements, or a block of Oracle™ PL/SQL or Microsoft SQL Server™ T-SQL procedural SQL statements.

  ```
  Select count(*) from company;

  begin
    Select sysdate from dual;
  end;¶

  Select count(*) from employee;
  ```

Executing – or Starting – a Statement

Once specified, a statement is executed by either using the Action/Start menu item, the Ctrl-E shortcut, or the ▶ toolbar item. “Statement execution in progress” is then both indicated by the hourglass cursor and an animation of the toolbar.

> Most relational database structured query language statements have two phases of execution: the statement execution phase, occurring within the relational database, and the statement fetch phase, where the result record set is transferred from the database server to the client.

Stopping a Statement Execution

A statement can be stopped by either using the Action/Stop menu item, the Ctrl-D shortcut, or the ■ toolbar item.

Execution is stoppable at both the execution and fetch phases; in some cases, client relational database software may not process the request, or may do so after a few seconds. (At times it is possible to re-attempt and gain a correct response from the client relational database software.)

Result Display

Statement execution result display occurs in the Query result panel. The display either consists of console-type text (or text-based output) or grid-type output. The display mode can be switched by using either the Action/Report Output menu item, or the toolbar item.

Note that execution of more than one SQL statements, by selecting text from the SQL Command panel, will display consecutively results in the Query result panel, if the query panel mode is text-based output.

Transaction Options

A transaction consists of one or many SQL statements affecting (or not) the database state. Completion of the transaction is accomplished by either confirming the transaction (COMMIT operation), or canceling the transaction (ROLLBACK operation.)

dbAnalyst allows either automatic transaction commit(s), or manual transaction commits or rollbacks. Transaction mode can be switched from automatic to manual commit mode or from manual to automatic commit mode by using the Action/Auto Commit menu item, or the toolbar item.

Transactions can be confirmed by using the Action/Commit menu item, the Alt-C shortcut, or the toolbar item. Transactions can be cancelled by using the Action/Rollback menu item, the Alt-R shortcut, or the toolbar item

By default, automatic transaction confirmation is enabled at startup. This default can be changed within the *SQL Console* Preference Screen, accessible using the File/Preferences menu item.

Execution Options

Result Pooling

Results displayed in the Query result panel can be streamed to a text file. Note that the Query result panel mode must be in text mode.

To start streaming, use the Action/Spool menu item, or the toolbar item. Prior to streaming, the user will be required to provide a file folder and a file name to proceed with the spooling operation. To stop streaming, use the Action/Spool menu item, or the toolbar item.

SQL Statement Analysis

A common chore of SQL statement optimization is that of analyzing and tweaking the SQL statement execution plan, to obtain optimum execution (lowest execution cost, shortest execution time, minimum I/O requests). Much of the statement analysis is vendor-specific, and this guide will not delve into details.

In most cases, the execution plan is displayed as a hierarchical record set, specifying each step of the execution plan. Each step will describe the database object affected, and the resource costs.

To start statement analysis, use the Action/Analyze SQL Statements menu item, or the 📝 toolbar item. Follow the same steps to cancel statement analysis.

Text Processing Features

Script Syntax Highlighting

Text displayed in the ***SQL Command*** panel has syntax highlighting enabled, as defined by the ***Database Definition Source XML file.*** Syntax highlighting includes different colors for comments, SQL keywords, data types, and functions. Note that the colors are driven by user preferences, accessible under the File/Preference menu item.

SQL Command Panel Text Processing Functions

Any of the following operations may be used while in the SQL Command panel:

- Clipboard cut operation - either using the Edit/Cut menu item, or the Ctrl-X shortcut, or the ✂ toolbar item.
- Clipboard copy operation - either using the Edit/Copy menu item, or the Ctrl-C shortcut, or the ⎘ toolbar item.
- Clipboard paste operation - either using the Edit/Paste menu item, or the Ctrl-V shortcut, or the 📋 toolbar item.
- Text Find operation - either using the Edit/Find menu item, or the Ctrl-F shortcut, or the 🔍 toolbar item. A Find Next operation is performed using either the Edit/Find Next menu item, or the F3 shortcut, or the 🔍 toolbar item.
- Text Replace operation - either using the Edit/Replace menu item, or the Ctrl-R shortcut, or the 🔍 toolbar item. Find-and-Replace Next operations are performed within the Replace modal window.

Query Result Panel Text Processing Functions

Any part of Query Result panel can be affected by:

- Clipboard copy operation - either using the Edit/Copy menu item, or the Ctrl-C shortcut, or the ⎘ toolbar item.
- Text Find operation - either using the Edit/Find menu item, or the Ctrl-F shortcut, or the 🔍 toolbar item. A Find Next operation is performed using either the Edit/Find Next menu item, or the F3 shortcut, or the 🔍 toolbar item.

Error Display

For any statement executed, dbAnalyst will report statement success or failure, possibly with a vendor-dependant error number and statement error position.

dbAnalyst relies heavily on the relational database ADO driver support for error detection and reporting mechanisms. It is to be noted that not all OLE DB drivers are equal in quality (and stability) and may provide very unequal error reporting output.

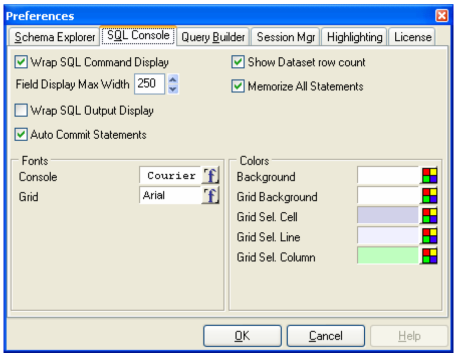
Safekeeping a Favorite Query

Any *SQL Console* query can be stored for later recall by accessing the Favorite/Add menu item, and by giving a name to the query. The query will then be listed under favorite queries.

Accessing the Favorite/Add menu, and selecting a named query will add the query to the command panel, and execute the query.

Options

Preferences for the SQL Console are located under the File Menu, Preference Menu Item, as a modal dialog window (Figure 30).



☑ Wrap SQL Command Display
Field Display Max Width 250
□ Wrap SQL Output Display
☑ Auto Commit Statements
☑ Show Dataset row count
☑ Memorize All Statements

Fonts
Console Courier
Grid Arial

Colors
Background
Grid Background
Grid Sel. Cell
Grid Sel. Line
Grid Sel. Column

Figure 30 – Preference Display, SQL Console Tab.

Preferences consist of:

- Wrap SQL Command Display.
  This option enables wrapping of the SQL Console upper quadrant.
- Field Display Max Width.
  The maximum number of characters used to display a single column, in the SQL Console lower quadrant, in text display mode.
- Wrap SQL Output Display.
  This option enables wrapping of the SQL Console lower quadrant.
- Auto Commit Statements.
  This option enables all SQL statements to be considered as

individual transactions. If a statement affects data content, then the statement is committed as executed.

- Show Dataset Row Count.
  This option enables display of the number of retrieved rows in the SQL Console lower quadrant, in text display mode.

- Memorize All Statements.
  This option enables all SQL statements executed in the SQL console to be memorized.

- Show Graph.
  This option indicates if the Graph tab is available.

- Reload Object Before Editing.
  This option indicates if objects should be reloaded from the database, before the Object Edition modal window appears.

- Display CREATE script as Hint.
  This option enabled displaying the object CREATE script as a graph panel hint.

- Fonts

  - Console Font.
    The font used to display text on upper and lower SQL Console Quadrants.

  - Grid Font.
    The font used to display grid text on the lower SQL Console Quadrant.

- Colors

  - Background Color.
    The color of the upper and lower SQL Console Quadrants.

  - Grid Background Color.
    The color of the SQL Console lower quadrant Grid display.

  - Grid Sel. Cell.
    The color of a selected grid cell.

  - Grid Sel. Line.
    The color of a selected grid line.

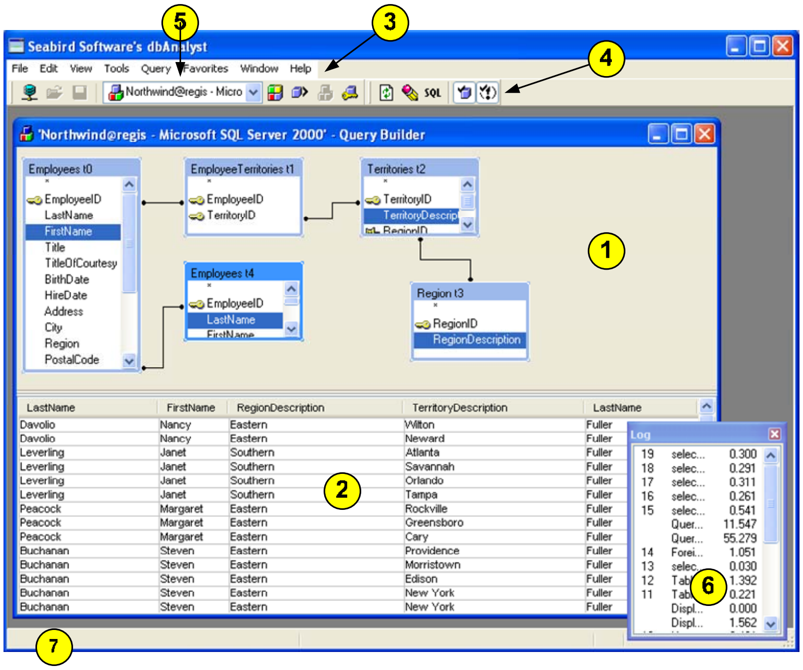
  - Grid Sel. Column.
    The color of a selected grid column.

The dbAnalyst *Query Builder*

Introduction

The *Query Builder* allows the building of complex database queries using an intuitive graphical user interface. Queries are exclusively built by selecting objects from objects lists, and/or by using drag-and-drop operations to establish object relationships. Note that Query Builder establishes object relationships whenever possible, (resulting in inner joints) by recognizing primary key and foreign key relationships.

Objects and object relationships are displayed on the upper half of the window. Data is displayed in grid format in the lower half-window. Note that both the lower and upper half can be resized by setting the cursor between the two panes, and dragging the cursor. Data displayed on the lower pane is modifiable by double-clicking on a cell. Data columns can be added by using drag-and-drop from the upper panel to the lower panel.



LastName	FirstName	RegionDescription	TerritoryDescription	LastName
Davolio	Nancy	Eastern	Wilton	Fuller
Davolio	Nancy	Eastern	Neward	Fuller
Leverling	Janet	Southern	Atlanta	Fuller
Leverling	Janet	Southern	Savannah	Fuller
Leverling	Janet	Southern	Orlando	Fuller
Leverling	Janet	Southern	Tampa	Fuller
Peacock	Margaret	Eastern	Rockville	Fuller
Peacock	Margaret	Eastern	Greensboro	Fuller
Peacock	Margaret	Eastern	Cary	Fuller
Buchanan	Steven	Eastern	Providence	Fuller
Buchanan	Steven	Eastern	Morristown	Fuller
Buchanan	Steven	Eastern	Edison	Fuller
Buchanan	Steven	Eastern	New York	Fuller
Buchanan	Steven	Eastern	New York	Fuller

```
19   selec...   0.300
18   selec...   0.291
17   selec...   0.311
16   selec...   0.261
15   selec...   0.541
     Quer...   11.547
     Quer...   55.279
14   Forei...   1.051
13   selec...   0.030
12   Tab        1.392
11   Tab        0.221
     Displ...   0.000
     Displ...   1.562
```

Figure 31 - Overview of the Query Builder

The *Query Builder* window consists of the following:

1 The *Object* Panel.
This panel is used to select and display objects and object relationships.

2 The *Data* Panel.
This panel displays query result data. Data can be edited.

3 The *Context Sensitive Menus.*
Menus are exclusively directed at operations available within the *Query Builder* tool. Most menu items are enabled in context in the context of the user's actions.

4 The *Toolbar.*
Toolbar items represent shortcuts to selected menu items; As with menu items, toolbar items are enabled in the context of the user's actions.

5 The *Navigation Palette.*
The Navigation palette shows the current list of database connections, and for each connection, opened tools, while also allowing selection and opening of new tools.

6 The *SQL Log.*
The SQL Log palette window shows all SQL statements that have been executed against the current database connection, since the connection was established. This log includes all operations, including *Schema Explorer* SQL statements, *SQL Console* statements and *Query Builder* statements. Double-clicking on a statement will re-execute the statement into the *SQL Console* tool.

7 The *Status Bar.*
The *Status Bar* shows run-time information, including the *SQL Command* Panel cursor position, etc.

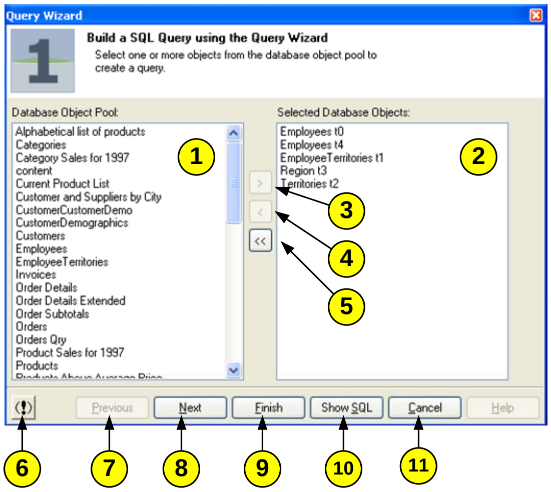
The next sections provide detailed information about the *Query Builder's* main elements.

Features

Creating Data Queries using the *Query Wizard*

Queries are mainly created using the *Query Wizard*. By default, the *Query Wizard* is displayed when a new Query Builder window is created. Creating a query using the *Query Wizard* is a multi step process:

- Selecting objects, such as tables and views.
  Objects are selected from the Database Object Pool (left panel) to the Selected Database Objects List (right panel).



Query Wizard

Build a SQL Query using the Query Wizard

Select one or more objects from the database object pool to create a query.

Database Object Pool:

Alphabetical list of products
Categories
Category Sales for 1997
content
Current Product List
Customer and Suppliers by City
CustomerCustomerDemo
CustomerDemographics
Customers
Employees
EmployeeTerritories
Invoices
Order Details
Order Details Extended
Order Subtotals
Orders
Orders Qry
Product Sales for 1997
Products
Products Above Average Price

Selected Database Objects:

Employees t0
Employees t4
EmployeeTerritories t1
Region t3
Territories t2

Figure 32 - Query Wizard Object Selection

1 The Database Object Pool List, showing all tables and views from the database connection.

2 The selected Database Object list, selected from the Database Object Pool list.

3 Allows selected objects from the Database Object Pool to be moved into the Selected Database Object List.

4 Allows selected objects from the Selected Database Object List to be moved back into the database object pool.

5 Allows *all* objects from the Selected Database Object List to be moved back into the database object pool.

6 Refresh.
Clicking on Refresh allows refreshing of the Database Object Pool list from the database.

7 Previous Step.
Allows going to the *Query Wizard* previous step. Since the object selection is the first step, this button is disabled in this step.

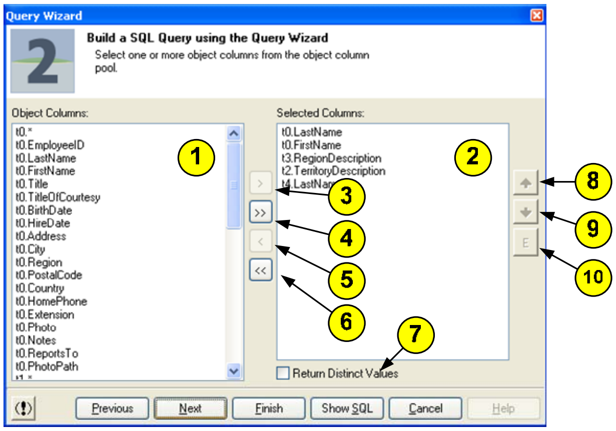
8 Next Step.
Allows going to the *Query Wizard* next step.

9 Finish.
Allows completing the *Query Wizard*.

10 Show SQL.
Allows expanding the *Query Wizard* window to show the SQL statement created as selecting are made within the *Query Wizard*.

11 Cancel.
Allows cancellation of the *Query Wizard*.

- Selecting object columns.
  Object columns are selected from the Object Column List (left panel) to the Selected Column List (right panel)



Build a SQL Query using the Query Wizard

Select one or more object columns from the object column pool.

Object Columns:

t0.*
t0.EmployeeID
t0.LastName
t0.FirstName
t0.Title
t0.TitleOfCourtesy
t0.BirthDate
t0.HireDate
t0.Address
t0.City
t0.Region
t0.PostalCode
t0.Country
t0.HomePhone
t0.Extension
t0.Photo
t0.Notes
t0.ReportsTo
t0.PhotoPath

Selected Columns:

t0.LastName
t0.FirstName
t3.RegionDescription
t2.TerritoryDescription
t4.LastNam

□ Return Distinct Values

Figure 33 - Query Wizard Object Column Selection

1 The Object Column List, showing all columns from the Object Selection List.

2 The selected Column list, showing column selections from the Object Column List.

3 The '>' button allows selected object columns from the Object Column List to be moved into the Selected Column List.

4 The '>>' button allows *all* object columns from the Object Column List to be moved into the Selected Column List.

5 The '<' button allows selected objects from the Selected Object Column List to be moved back into the Object Column list.

6 The '<<' button allows *all* objects from the Selected Object Column List to be moved back into the Object Column list.

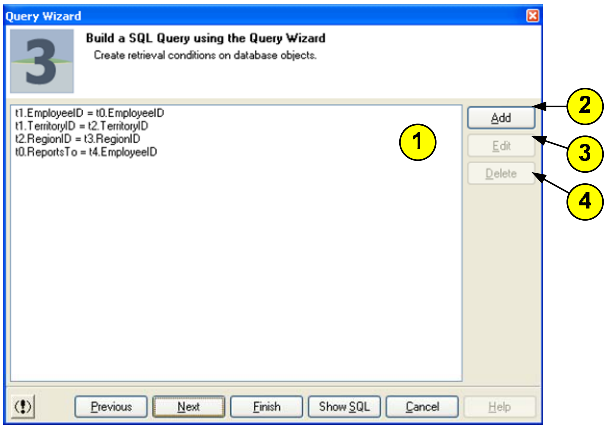
7 Returns Distinct Values.
This check box allows only distinct values to be returned as a data set.

8 Moves Column Up.
Columns can be reordered by selecting one or many columns and pressing on this button.

9 Move Column Down.
Reorder columns by selecting one or many columns and pressing on this button.

10 Edit Column Expression.
Column expressions can be modified and expanded to accept function, and other types of expressions.

- Creating query conditions.
  Conditions represent relationships between tables and views, in terms of inner and outer joins. Conditions can be established automatically by dbAnalyst when selecting objects, or by drag-and-drop operations on the Object Panel, or in the Query Wizard Condition tab.


Query Wizard
3
Build a SQL Query using the Query Wizard
Create retrieval conditions on database objects.
Add
Edit
Delete
(!)
Previous
Next
Finish
Show SQL
Cancel
Help
1
2
3
4

```
t1.EmployeeID = t0.EmployeeID
t1.TerritoryID = t2.TerritoryID
t2.RegionID = t3.RegionID
t0.ReportsTo = t4.EmployeeID
```

Figure 34 - Query Wizard Condition Creation

1 The list of existing conditions.

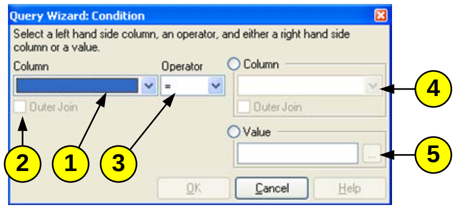
2 Click on ***Add*** to create a new condition. Refer to Figure 35 for the description of new condition creation.

3 Select a condition and click on ***Edit*** to modify the condition.

4 Select a condition and click on ***Delete*** to remove the condition.

The Condition detail modal window allows for creation of a new condition or editing of an existing condition. A condition consists of an expression with a left hand side expression, a logical operator, and a right hand side expression. The left hand side expression must be a column, and may be optionally a left outer join. The right hand side expression may be either:

- A column expression, optionally a right outer join
- A value expression consisting of a string, a date or a number. The '…' button allows distinct values to be presented for easier entry.



Query Wizard: Condition

Select a left hand side column, an operator, and either a right hand side column or a value.

Figure 35 - Condition Detail Creation and Modification

1 Allows for selection of a column, from any of the objects selected in the object selection tab.

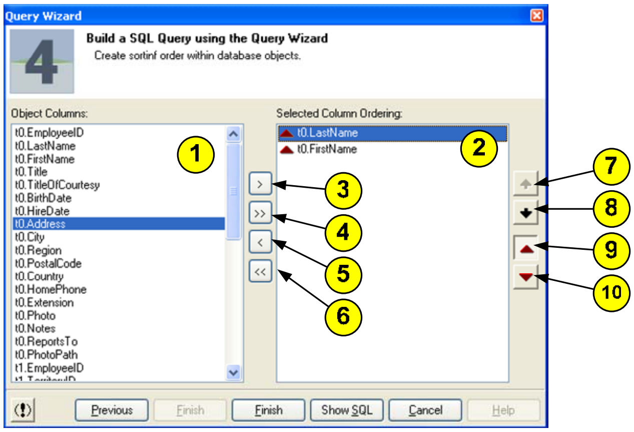
2 The left outer join check box.

3 The condition operator.

4 The right hand side expression as a column expression. Click on the radio button to enable right hand side column expression.

5 The right hand side expression as a value expression. Click on the radio button to enable right hand side value expression.

- Creating query Ordering.
  Conditions represent relationships between tables and views, in term of inner and outer joins. Conditions can be established automatically by dbAnalyst when selecting objects, or by drag-and-drop operations on the Object Panel, or in the Query Wizard Condition tab.



Query Wizard

Build a SQL Query using the Query Wizard

Create sortinf order within database objects.

Figure 36 - Query Wizard Ordering

1 The Object Column List, showing all columns from the Object Selection List.

2 The selected Column list for ordering.

3 The '>' button allows object columns from the Object Column List to be moved into the Selected Column ordering List.

4 The '>>' button allows *all* object columns from the Object Column List to be moved into the Selected Column ordering List.

5 The '<' button allows selected objects from the Selected Object Column ordering List to be moved back into the Object Column list.

6 The '<<' button allows *all* objects from the Selected Object Column ordering List to be moved back into the Object Column list.

7 Allows reordering a sorting column ***up***.

8 Allows reordering a sorting column ***down***.

9 Allows changing the column ordering to be ***ascending***.

10 Allows changing the column ordering to be ***descending***.

Upon completion of the query wizard, the resulting query is executed and the results are displayed in the data panel. The Query Wizard can be recalled by either using the Query/Query Wizard menu item, or by clicking on the toolbar icon.

Object Panel Drag-and-Drop Operations

Moving Object Panels

Panel location can be changed by clicking on the Panel header, and holding the mouse button, while moving the mouse. The panel will follow the cursor. Release the mouse button to 'settle' the panel location.

Conditions

Additional conditions can be created by dragging an object column on top of another object column.

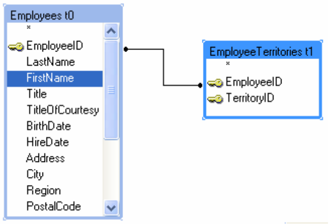
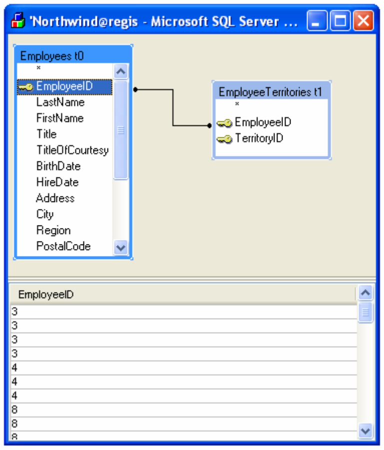


Figure 37 - Object Panel Drag-and-Drop Operation

A condition is then created to reflect the relationship.

Data Panel Columns

Additional columns can be created in the Data panel by dragging an object column into the Data Panel.

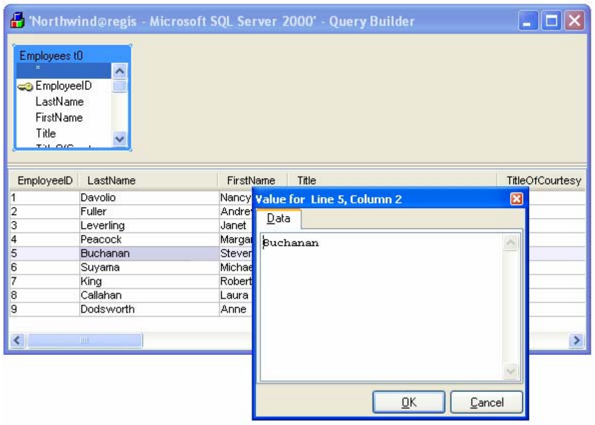


EmployeeID
3
3
3
3
4
4
4
8
8

Figure 38 - Data Panel Drag-and-Drop Operation

Editing Data

Double-clicking any data panel cell brings about the data modification modal window.


'Northwind@regis - Microsoft SQL Server 2000' - Query Builder
Employees t0
*
EmployeeID
LastName
FirstName
Title
Data
OK
Cancel

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
1	Davolio	Nancy		
2	Fuller	Andre		
3	Leverling	Janet		
4	Peacock	Margai		
5	Buchanan	Steven		
6	Suyama	Michae		
7	King	Robert		
8	Callahan	Laura		
9	Dodsworth	Anne		

Value for Line 5, Column 2

Buchanan

Figure 39 - Data Panel data modification modal window

The Query Builder may represent any level of SQL statement complexity. Editing a single data cell within a complex SQL expression might prevent data editing; it is up to the OLE DB driver to support (or not) the data modification.

Query *Manual* modification

The current SQL statement can be viewed and edited at any time by hand using the Query/Edit SQL menu item, or by using the Ctrl-E shortcut, or by clicking on the SQL toolbar icon.

Refreshing the Data Panel

The data panel can be refreshed at any time by using the Edit/Refresh menu item, or by using the F5 shortcut, or by clicking on the (!) toolbar icon.

Note that each modification of the query will cause the Data Panel to refresh automatically. Automatic Refresh operations can be enabled or disabled by using the Query/Auto Refresh menu item, or by clicking on the ✓(!) toolbar icon.

Hiding the Object Panel

The Object Panel can be hidden or shown at any time by using the Query/View Object menu item, or by clicking on the ■ toolbar icon.

Safekeeping a Favorite Query

Any *Query Builder* query can be stored for later recall by accessing the Favorite/Add menu item, and by giving a name to the query. The query will then be listed under favorite queries.

Accessing the Favorite/Add menu and selecting a named query will clear the *Query Builder* Object Panel, and recreate the saved favorite query.

Appendix A: dbAnalyst Object Edition Capability Matrix

Object Class	Action		Oracle				Microsoft SQL Server		PostgreSQL		MySQL 4.0	dBsase	iAnywhere 8.0	Microsoft Access 2000
			Oracle8 8.0.5	Oracle8i 8.1.6	Oracle9i 9.0, 9.1	Oracle9i 9.2	MS SQL7	MS SQL 2000	7.1, 7.2	7.3, 7.4				
Table	Create		√	√	√	√	√	√	√	√	√	√	√	√
	Drop		√	√	√	√	√	√	√	√	√	√	√	√
	Rename		√	√	√	√	√	√	√	√	√	▨	√	▨
	Change Owner		Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
	Change Table Space		Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
	Cluster		Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
	Column	Add	√	√	√	√	√	√	√	√	√	√	√	√
		Delete	▨	√	√	√	√	√	√	√	√	√	√	√
		Rename	▨	▨	√	√	√	√	√	√	√	▨	√	▨
		Change Data type	√	√	√	√	√	√	▨	▨	√	▨	√	▨
		Change Default Value	√	√	√	√	Ø	Ø	√	√	√	▨	√	▨
		Change Nullable	√	√	√	√	√	√	▨	√	√	▨	√	▨
		Auto-indent Mgt	▨	▨	▨	▨	Ø	Ø	▨	▨	Ø	▨	Ø	▨
	Constraints	Primary Key	Ø: Constraint Management not yet supported											
		Foreign Key												
		Check Constraint												
		Unique Constraint												
Sequence	Create		√	√	√	√	▨	▨	√	√	▨	▨	▨	▨
	Drop		√	√	√	√	▨	▨	√	√	▨	▨	▨	▨
	Edit		√	√	√	√	▨	▨	√	√	▨	▨	▨	▨
Index	Create		Index creating or editing not yet supported											
	Drop		√	√	√	√	√	√	√	√	√	√	√	√
View	Create		√	√	√	√	√	√	√	√	▨	▨	√	▨
	Drop		√	√	√	√	√	√	√	√	▨	▨	√	▨
	Edit		√	√	√	√	√	√	√	√	▨	▨	√	▨
	Note			⇨ Lost CR/LF at edit					⇨ Lost CR/LF at edit		▨	▨		▨
Trigger	Create		√	√	√	√	√	√	√	√	▨	▨	√	▨
	Drop		√	√	√	√	√	√	√	√	▨	▨	√	▨
	Edit		√	√	√	√	√	√	√	√	▨	▨	√	▨
	Note								Partial support		▨	▨		▨

Object Class	Action	Oracle				Microsoft SQL Server		PostgreSQL		MySQL 4.0	dBsase	iAnywhere 8.0	Microsoft Access 2000
		Oracle8 8.0.5	Oracle8i 8.1.6	Oracle9i 9.0, 9.1	Oracle9i 9.2	MS SQL 7	MS SQL 2000	7.1 7.2	7.3 7.4				
Procedure	Create	√	√	√	√	√	√	Ø	Ø	▨	▨	√	▨
	Drop	√	√	√	√	√	√	√	√	▨	▨	√	▨
	Edit	√	√	√	√	√	√	Ø	Ø	▨	▨	√	▨
	Note		➪ CR/LF are lost during edits.							▨	▨		▨
Function	Create	√	√	√	√	▨	√	Ø	Ø	Ø	▨	▨	▨
	Drop	√	√	√	√	▨	√	√	√	Ø	▨	▨	▨
	Edit	√	√	√	√	▨	√	Ø	Ø	Ø	▨	▨	▨
	Note		➪ CR/LF are lost during edits.								▨	▨	▨
Package	Create	Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
	Drop	Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
	Edit	Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
	Note	Ø: Package mgt not yet supported				▨	▨	▨	▨	▨	▨	▨	▨
Synonym	Create	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Drop	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Edit	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
Database Link	Create	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Drop	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Edit	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
Grant	Create	Ø: Grant editing not yet supported								▨	▨	Ø	▨
	Drop	√	√	√	√					▨	▨	Ø	▨
Library	Create	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Drop	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Edit	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
Directory	Create	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Drop	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
	Edit	√	√	√	√	▨	▨	▨	▨	▨	▨	▨	▨
Type		Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	▨	▨	▨	▨
Comment		Ø	Ø	Ø	Ø	▨	▨	Ø	Ø	▨	▨	▨	▨
Cluster		Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
Outline		Ø	Ø	Ø	Ø	▨	▨	▨	▨	▨	▨	▨	▨
Language	Create	▨	▨	▨	▨	▨	▨	Ø	Ø	▨	▨	▨	▨
	Drop	▨	▨	▨	▨	▨	▨	√	√	▨	▨	▨	▨
	Edit	▨	▨	▨	▨	▨	▨	Ø	Ø	▨	▨	▨	▨
Missing in Postgres: Aggregate, Cast, Conversion, Domain, Group, Operator, Operator Class, Rule, Schema		▨	▨	▨	▨	▨	▨	Ø	Ø	▨	▨	▨	▨
Features	Analyze (Explain Plan)	√	√	√	√	√	√	▨	√	√	▨	Ø	▨
	Describe	√	√	√	√	√	√	√	√	√	▨	√	▨
	Session Manager	√	√	√	√	√	√	▨	√	Ø	▨	▨	▨