DBMaker

JDBA Tool User's Guide

CASEMaker®

CASEMaker Inc./Corporate Headquarters
1680 Civic Center Drive
Santa Clara, CA 95050, U.S.A.
www.casemaker.com
www.casemaker.com/support

Document No. 645049-231062/DBM41-M05262003-JDBA
Publication Date: 2003-05-20

For a description of updated functions that do not appear in this manual, read the file named README.TXT after installing the CASEMaker DBMaker software.

Trademarks
CASEMaker, the CASEMaker logo, and DBMaker are registered trademarks of CASEMaker Inc. Microsoft, MS-DOS, Windows, and Windows NT are registered trademarks of Microsoft Corp. UNIX is a registered trademark of The Open Group. ANSI is a registered trademark of American National Standards Institute, Inc.

Other product names mentioned herein may be trademarks of their respective holders and are mentioned only form information purposes. SQL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

Notices
The software described in this manual is covered by the license agreement supplied with the software.

Contact your dealer for warranty details. Your dealer makes no representations or warranties with respect to the merchantability or fitness of this computer product for any particular purpose. Your dealer is not responsible for any damage caused to this computer product by external forces including sudden shock, excess heat, cold, or humidity, nor for any loss or damage caused by incorrect voltage or incompatible hardware and/or software.

Information in this manual has been carefully checked for reliability; however, no responsibility is assumed for inaccuracies. This manual is subject to change without notice.

Contents

1 Introduction

Welcome to the JDBA Tool User's Guide. JDBA Tool is a cross-platform user-friendly graphical user interface (GUI) that helps users to easily manage database objects in DBMaker, a powerful and flexible SQL Database Management System. JDBA Tool hides the complexity of the DBMS and query language and provides an easy to understand and convenient interface. You can perform database functions without having to learn SQL. JDBA Tool also provides statistical data and information on who is using your database with its monitoring functions.

This manual gives systematic instructions on how to use the JDBA Tool to manage database objects in DBMaker. The features of the program are explained with a brief description of their function followed by procedures that guide you through proper use of the feature. The procedural steps are periodically reinforced by figures of the JDBA Tool screen.

This book is intended for users who will design and administer DBMaker databases using JDBA Tool. It is intended to teach those who are unfamiliar with using DBMaker, but have some understanding of how a relational database works. It is assumed that the user does have some knowledge of the platform on which they are working from. Information in this manual may also be helpful as a reference for experienced users.

To perform JDBA Tool tasks, refer to specific sections in the chapters. You should read any descriptions included before the procedures and follow each of the steps in the procedures. It is also recommended that you read other sections within the same chapter, as this will increase your understanding of JDBA Tool.

1.1 Additional Resources

DBMaker provides a complete set of DBMS manuals in addition to this one. For more detailed information on a particular subject, consult one of the books listed below:

For an introduction to DBMaker's capabilities and functions, refer to the *DBMaker Tutorial*.

For more information on designing, administering, and maintaining a DBMaker database, refer to the *Database Administrator's Guide*.

For more information on database server management, refer to the *JServer Manager User's Guide*.

For more information on configuring DBMaker, refer to the *JConfiguration Tool Reference*.

For more information on the native ODBC API, refer to the *ODBC Programmer's Guide*.

For more information on the dmSQL interface tool, refer to the *dmSQL User's Guide*.

For more information on the SQL language used in dmSQL, refer to the *SQL Command and Function Reference*.

For more information on the ESQL/C programming, refer to the *ESQL/C User's Guide*.

For more information on error and warning messages, refer to the *Error and Message Reference*.

For more information on the DBMaker COBOL Interface, refer to the *DCI User's Guide*.

1.2 Technical Support

CASEMaker provides thirty days of complimentary email and phone support during the evaluation period. When software is registered an additional thirty days of support will be included. Thus, extending the total support period for software to sixty days. However, CASEMaker will continue to provide email support for any bugs reported after the complimentary support or registered support has expired (free of charges).

Additional support is available beyond the sixty days for most products and may be purchased for twenty percent of the retail price of the product. Please contact sales@casemaker.com for more details and prices.

CASEMaker support contact information for your area (by snail mail, phone, or email) can be located at: www.casemaker.com/support. It is recommended that the current database of FAQ's be searched before contacting CASEMaker support staff.

Please have the following information available when phoning support for a troubleshooting enquiry or include the information with a snail mail or email enquiry:

Product name and version number

Registration number

Registered customer name and address

Supplier/distributor where product was purchased

Platform and computer system configuration

Specific action(s) performed before error(s) occurred

Error message and number, if any

Any additional information deemed pertinent

1.3 Document Conventions

This book uses a standard set of typographical conventions for clarity and ease of use. The NOTE, Procedure, Example, and CommandLine conventions also have a second setting used with indentation.

CONVENTION	DESCRIPTION
Italics	Italics indicate placeholders for information that must be supplied, such as user and table names. The word in italics should not be typed, but is replaced by the actual name. Italics also introduce new words, and are occasionally used for emphasis in text.
Boldface	Boldface indicates filenames, database names, table names, column names, user names, and other database schema objects. It is also used to emphasize menu commands in procedural steps.
KEYWORDS	All keywords used by the SQL language appear in uppercase when used in normal paragraph text.
SMALL CAPS	Small capital letters indicate keys on the keyboard. A plus sign (+) between two key names indicates to hold down the first key while pressing the second. A comma (,) between two key names indicates to release the first key before pressing the second key.
NOTE	Contains important information.
➲ **Procedure**	Indicates that procedural steps or sequential items will follow. Many tasks are described using this format to provide a logical sequence of steps for the user to follow
➲ **Example**	Examples are given to clarify descriptions, and commonly include text, as it will appear on the screen. Other forms of this convention include Prototype and Syntax.
`CommandLine`	Indicates text, as it should appear on a text-delimited screen. This format is commonly used to show input and output for dmSQL commands or the content in the dmconfig.ini file

Table 1-1Document Conventions

2 Getting to know JDBA Tool

This chapter introduces the layout of the JDBA Tool workspace, and contains general information about many of its components. You will also learn how to connect and disconnect from a database.

JDBA Tool as well as all the other *DBMaker* Java tools (JServer Manager, Java Configuration Tool) operate in the Java Run Time Environment. To start these programs the jdk1.3 (Java Development Kit 1.3) or the JRE (Java Run Time Environment) for Java should be installed on the system. JRE 1.3.1 is automatically installed on the system if DBMaker release 4.1 is installed from disk.

2.1 Summary of Features

JDBA Tool has a wide variety of features that will enable you to effectively manage your database.

You can easily create table schema in a database and add columns to it. You can also define constraints for data in a table.

After creating the table schema and adding columns, you can insert records to a table. Moreover, you can define a constraint to ensure that each inserted record falls within a specified range.

You can define a column using a domain. The column inherits the properties of the domain (data type, default value, and value constraint) without requiring you to explicitly specify them.

After a table is created successfully, it is easy to create an index, primary key, or foreign key in a table. A foreign key lets you establish a relationship between different tables in a database.

DBMaker allows you to query easily by creating a view from various tables or specifying a synonym for a specific table. Views and synonyms let you refer to and manipulate data from tables quickly and easily.

After creating all database objects, you can easily drop a table, view, synonym, or domain. (To drop some objects, you should take into account the mutual relationships between objects. For example, you should drop a foreign key before dropping the referred primary key.)

Any authorized user can create database links to access database objects on remote databases

You can transfer table data between local and remote table objects through the Table Replication command.

To manage frequently used SQL commands, you can create stored commands which can be reused whenever needed.

To perform frequently repeated tasks easily, stored procedures are used. Thus, eliminating the need for repeated SQL compilation and optimization.

For handling complex transactions in the database, you can create triggers that will automatically execute when the user inserts, deletes, or updates data in the database.

A stored procedure can be called by a trigger action

Text indices can be created for columns with CHAR, VARCHAR, LONGVARCHAR and FILE data types.

As the system administrator, you can add users or groups to a database. After adding users and groups to a database, you can assign privileges to a user or a group. The benefit of a group is that you can assign the same privileges to multiple users. You only need to assign these privileges to a group once.

Monitor the current system environment such as which database users are currently connected to the database, locks placed on data objects, and all current runtime settings.

You can set default values for creating tables and choose to display BLOB data using SQL statements.

2.2 About the JDBA Tool Workspace

Figure 2-1 shows the information required for logging on to the JDBA Tool. It is recommended that all the database settings be made before logging on to the database. The database settings are changed using the configuration tool. The **Setup** button launches the JConfiguration Tool. Enter the **Database Name**, **User Name** and **Password** and click **Setup** to check or modify the advanced settings of the database.

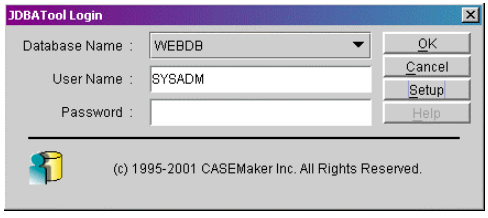


Figure 2-1 The JDBA Tool Login window

For details regarding settings, please refer to the *JConfiguration Tool Reference*. After you log onto a database the elements of the GUI will appear on the screen. These elements are summarized in Figure 2-2 and described in the following subsections

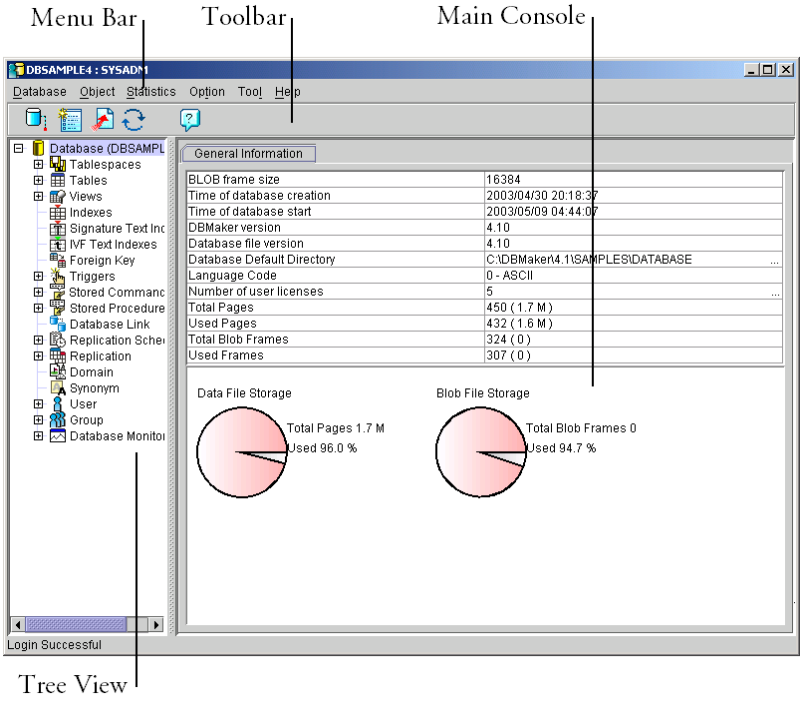


Figure 2-2 The elements of the JDBA Tool Graphical User Interface

Menu Bar

The menu bar displays the JDBA Tool's pull-down menus. Each menu item is used for performing different functions and is discussed in detail in the following chapters of this book.

The Database in Tree View

The database tree displays all the objects in the database. The root of the tree is the name of the database, the rest of the objects follow down the tree. The name of each

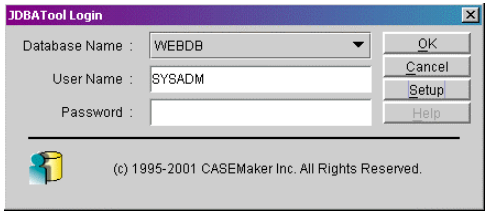
object is also displayed. With the help of the database monitor in the tree, the user can monitor the database and keep track of users and their transactions.

2.3 Connecting to a Database

To manipulate data in a database, you must connect to the database first. JDBA Tool allows you to connect to a single database at a time. In order to connect to a multi-user database with JDBA Tool, a database server program must already be running. The database can be started with JServer Manager or DBMaker Server. After the database is started, proceed with the following steps to connect to the database.

➲ **To connect to a database:**

1. Select a database from the **Database Name** menu in the **JDBA Tool Login** window.



2. Enter your user ID in the User Name field.

   NOTE *If a default user ID has already been set, then the default user ID is automatically displayed in the User Name field.*

3. If you have been assigned a password, enter it in the **Password** field
4. Click **Setup** to open the Connect Database Advanced Setting window.
5. Set options in the following categories:

   Connection

   DmServer

   Data Manipulation

   SQL Attributes

   NOTE *Refer to the JConfiguration Tool Reference for more information about advanced settings.*

6. Click **OK**. The **General Information** window displays the following information:

The DBMaker version (4.1).

The file version.

The date and time of database creation

The date and time at which the database was started.

The BLOB frame size.

The database default directory

The language code

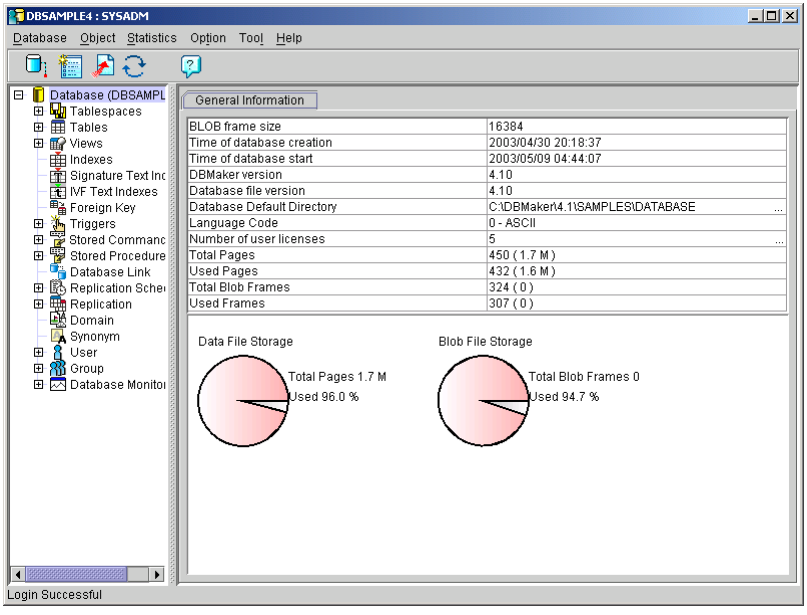
The number of users who have permission to access the database.

The total number of data pages reserved by the database.

The number of data pages reserved by the database that are used.

The total number of BLOB frames reserved by the database

The number of used BLOB frames.



NOTE *A graphical representation of the Data and BLOB File storage is also displayed.*

7. If you wish to connect to another database that has been started by your server program, click the Database pull-down menu and select Connect Database. Repeat steps 1 to 4. Connecting to a new database will automatically disconnect the database that you were connected to earlier.

2.4 Disconnecting from a Database

Once you have finished using the database, you can disconnect from the database.

NOTE *Since JDBA Tool works in an AUTOCOMMIT mode, all database transactions are automatically written to the disk before disconnecting from the database.*

➲ **To disconnect from a database:**

1. Select **Disconnect Database** from the database menu. The JDBA console will appear empty.

3 Working with Tablespaces

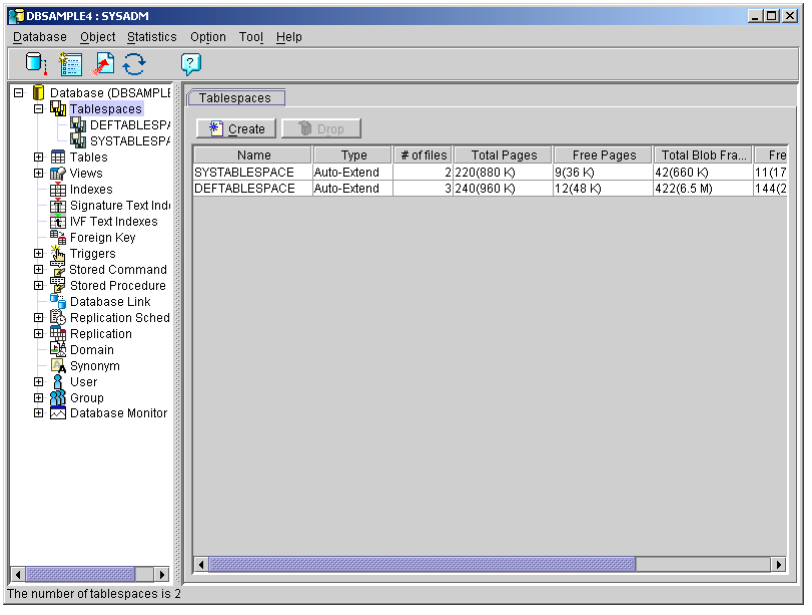
A DBMaker database is composed of one or more logical divisions known as *tablespaces*. With tablespaces, the database can be divided into manageable areas. In the logical view, a tablespace contains one or more tables and indexes. In the physical view, a tablespace is the physical storage (directory path) that consists of one or more files. By the proper use of tablespaces, data is better organized both at the physical and logical level, and storage space is used more efficiently.

3.1 Browsing Tablespaces

All DBMaker databases have at least two tablespaces. The first is called the system tablespace (SYSTABLESPACE), and the second is the default user tablespace (DEFTABLESPACE). Whenever you create a database, DBMaker generates a system tablespace to record the *system catalog tables*. The system catalog tables store information about the entire database. The default tablespace (DEFTABLESPACE) is used to store user tables if users do not define their own tablespace. You can view information about all the tablespaces in a database by selecting the Tablespace item from the tree.

➲ **To browse a tablespace:**

1. Click the **Tablespace** object in the tree. A list of all tablespaces is displayed.



2. Browse the tablespaces listed on the right panel by using the upward and downward arrows. You can see all the tablespace information listed under the following fields.

Name – tablespace name

Type – tablespace type, can be regular tablespace or autoextend tablespace.

# of files – number of files in the tablespace, including data files and BLOB files

Total Pages – number of total pages in data files

Free Pages – number of free pages in data files

Total Frames – number of total frames in BLOB files

Free Frames – number of free frames in BLOB files

Creation Time – tablespace creation time

3.2 Creating a Tablespace

Smaller tables that contain data of a similar nature can be grouped in a single tablespace, but very large tables should be placed in their own tablespaces. A regular tablespace has a fixed size and contains one or more data files. If the physical file of a regular tablespace is too small to hold all the data you wish to store in it, you must enlarge it manually. An autoextend tablespace is a tablespace that will automatically grow as needed.

Consider the following when creating a tablespace:

The logical name must be a unique name.

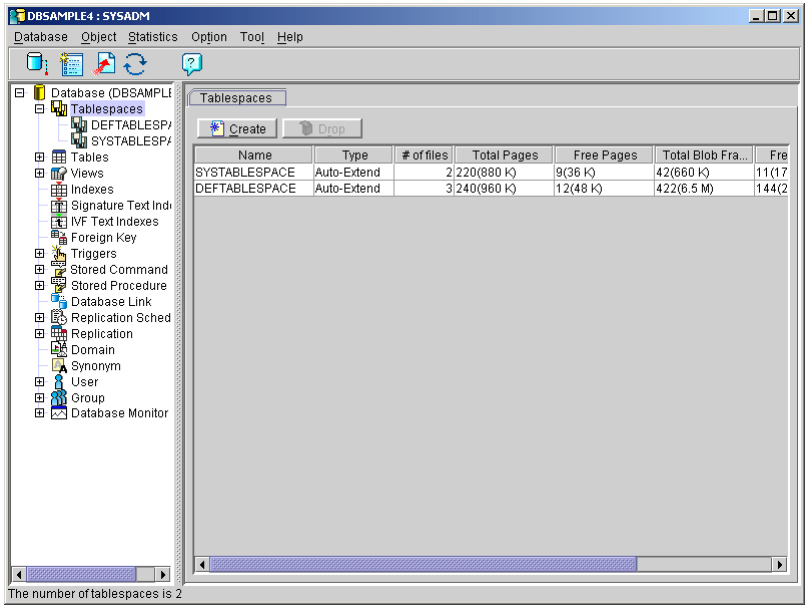
There are two types of files available: DATA or BLOB.

In the case of a data file in a tablespace, the size is represented as the number of pages. In the case of a BLOB file, the size is represented as the number of frames.

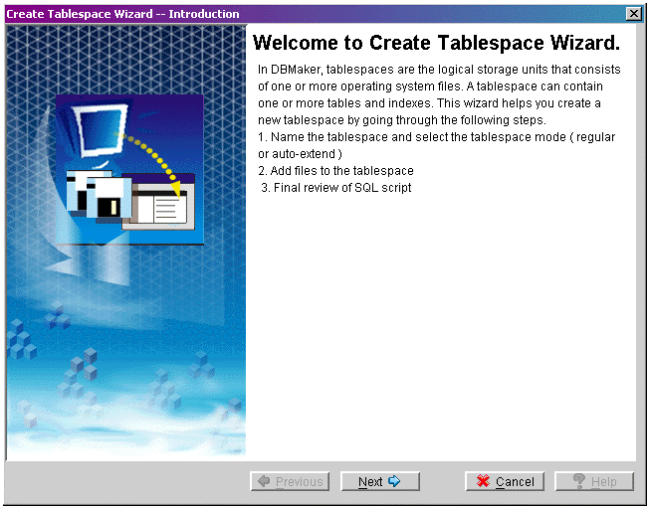
A BLOB file cannot be the only file in a single tablespace. There should be at least one data file in a tablespace. Tablespace creation will fail if no data file is included.

➲ **To create a tablespace:**

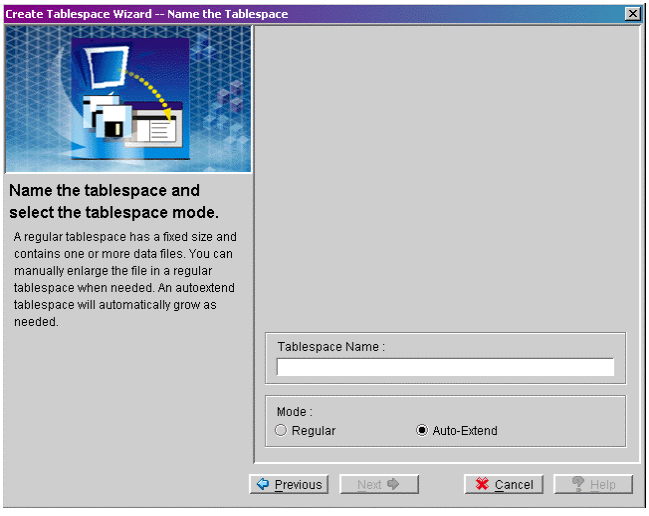
1. Click the **Tablespaces** object in the tree. The **Tablespaces** page is displayed



2. Click the **Create** button in the **Tablespaces** window. The **Create Tablespace Wizard** window is opened.



3. Click the **Next** button. The **Name the Tablespace** window is displayed.



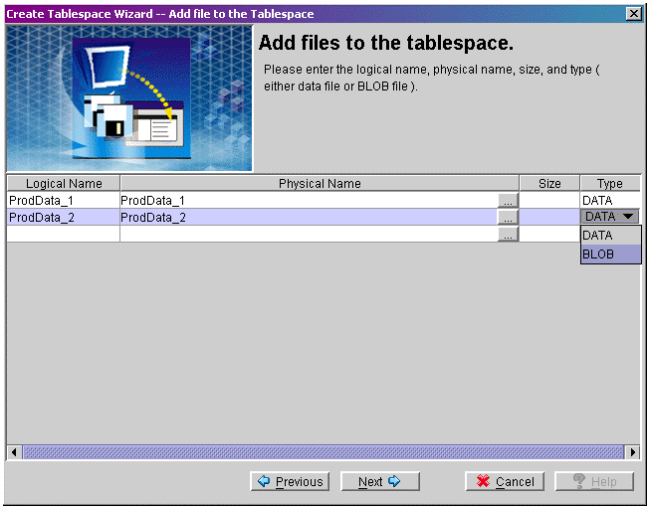
4. Enter the name of the tablespace in the **Tablespace Name** field.

5. Select a tablespace type:

To select a regular tablespace, select the **Regular** option button.

To select an Auto-Extend tablespace, select the **Auto-Extend** option button.

6. Click the **Next** button. The **Add file to the Tablespace** window is displayed.



7. Enter the logical name of the tablespace in the **Logical Name** column.

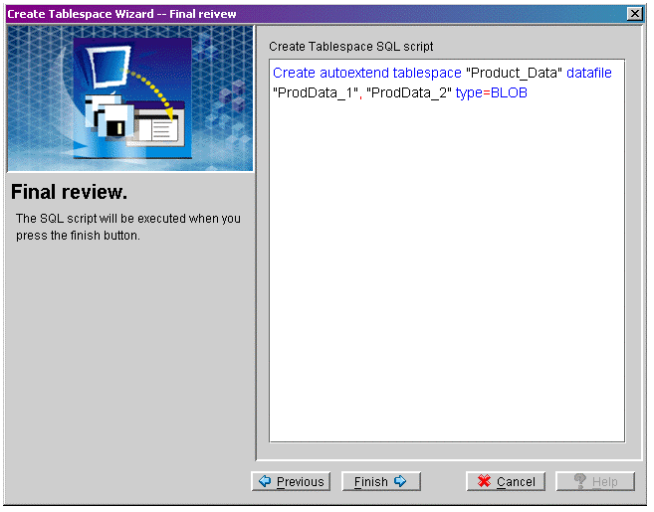
8. Enter the physical name of the tablespace in the **Physical Name** column.

NOTE *You can also select the physical name of the tablespace from a list by clicking the Browse button.*

9. Enter the size of the tablespace in the **Size** column.

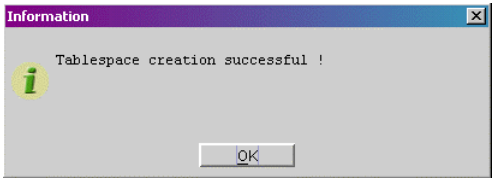
10. Select the tablespace type from the **Type** column.

11. Click the **Next** button to proceed. The wizard will display the SQL command for creating a new tablespace. This command can be edited.



12. To edit the command, enter or delete command syntax in the command line area.

13. Click the **Finish** button. The **Information** window of the **Create Tablespace Wizard** is displayed.



3.3 Altering a Tablespace

After creating a tablespace, the user may want to alter the tablespace in the following ways:

Add data or BLOB files to a tablespace.

Compress a tablespace.

Add pages to files in a tablespace.

Change the tablespace from autoextend to regular or vice versa.

Adding Files to a Tablespace

Once you have created a tablespace, you can add data or BLOB files to a tablespace. Consider the following when adding files to a tablespace:

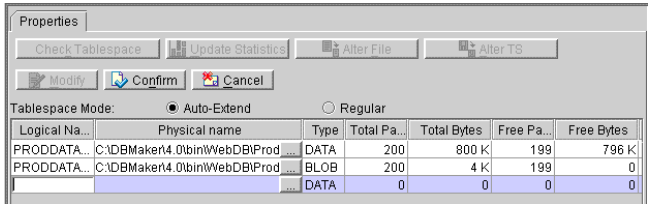
The logical name must be a unique name.

There are two types of files available: DATA or BLOB.

In the case of a data file in a tablespace, the size is represented as the number of pages. In the case of a BLOB file, the size is represented as the number of frames.

➲ **To add files to a tablespace:**

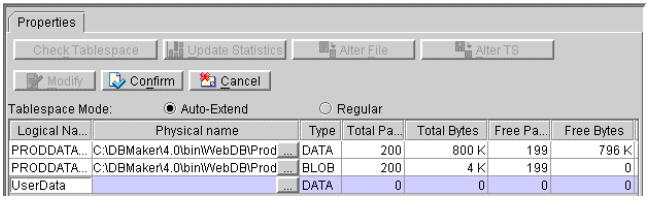
1. Select the tablespace to which you want to add one or more files. The properties for the tablespace will be displayed in the main console.
2. Click the **Modify** button. An empty row is displayed in the tablespace list.



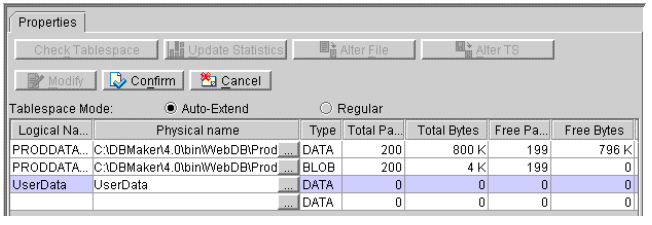
3. To change the Tablespace mode from Auto-Extend to Regular, select the **Regular** option button above the list.

4. To change the **Tablespace Mode** from Regular to Autoextend, select the **Auto-Extend** option button above the list.

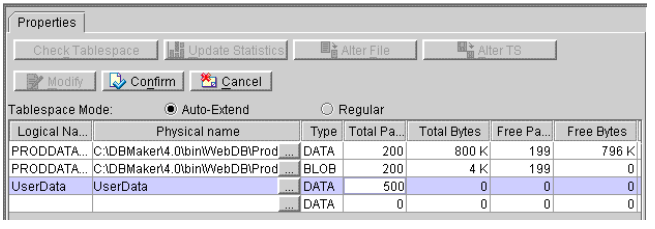
5. Enter the logical name of the tablespace in the **Logical Name** column.



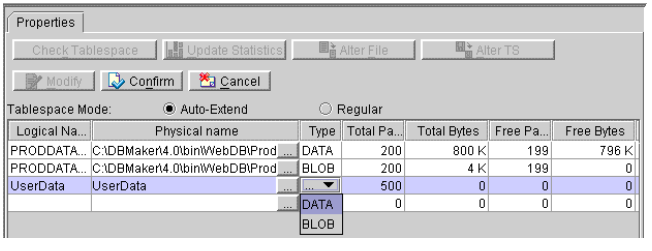
6. Enter the physical name of the tablespace in the **Physical Name** column.



7. Enter the size of the tablespace in the **Total Pages** column.



8. Select the tablespace type from the **Type** column



9. Press on the **Confirm** button. You can see the new file added to the rest of the files in the tablespace.

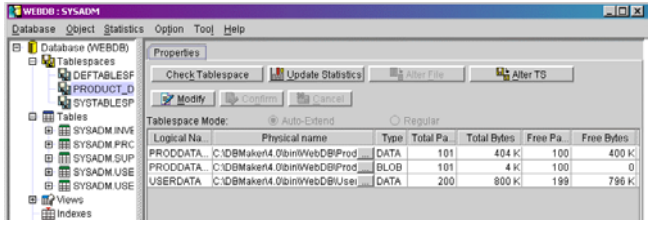
Shrinking and Compressing a Tablespace

In order to optimize storage space, the user may choose to compress the tablespace by moving the data pages and BLOB frames to the beginning of the file or shrink the tablespace by changing the number of free pages remaining in each of the files in the tablespace.

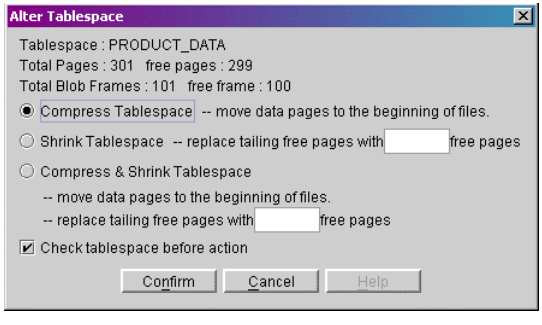
The number of tailing free pages you want the tablespace to contain should be less than the number of free pages that are being replaced. By replacing the number of tailing free pages in a tablespace the total number of free pages in each of the files in the tablespace is altered.

➲ **To shrink or compress a tablespace:**

1. Select the tablespace that you want to compress or shrink. The **Properties** page is displayed.



2. Click the **Alter TS** button on top of the window. The **Alter Tablespace** dialog box is displayed.



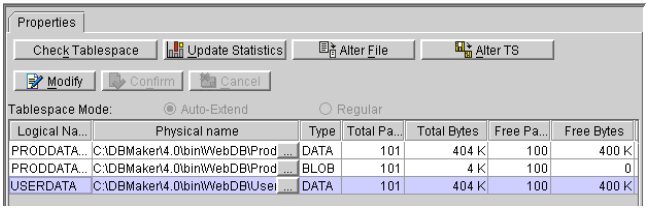
3. To compress the tablespace, select the **Compress Tablespace** option button.
4. To reduce the tailing or remaining free pages in tablespace files, select the **Shrink Tablespace** option button.
5. If you want to reduce the tailing or remaining free pages in the files of the tablespace, enter the number of tailing free pages you want the files in the tablespace to contain.
6. To compress and shrink the tablespace:
   a) Select the **Compress and Shrink Tablespace** option button.
   b) Enter the number of tailing free pages you want the tablespace to contain in the text field.
7. In order to check the consistency of the database before changes are made to the tablespace, click the **Check tablespace before action** checkbox
8. Click the **Confirm** button. The **Alter Tablespace** confirmation dialog box is displayed.
9. Click **OK**.

Modifying a File in a Tablespace

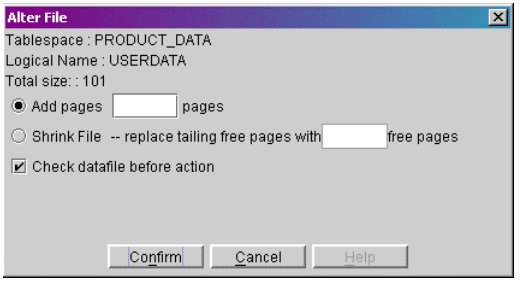
After adding files to a tablespace, it may be necessary to change one particular file in the tablespace or alter the tablespace itself

➲ **To modify a single file in a tablespace:**

1. Select the tablespace from the tree or the **Tablespaces** page that contains the file you want to modify. The properties of the tablespace will be displayed in the main window.



2. Click the **Alter File** button. The **Alter File** dialog box is displayed.



NOTE *The Alter File button is disabled until a file is highlighted in the Tablespace. The highlighted file will be altered.*

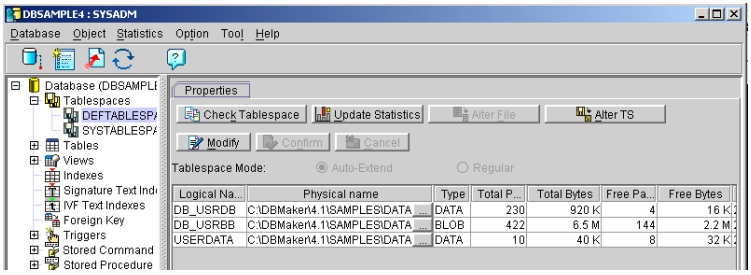
3. Enter the number of pages you want to add to the file.
4. If you want to shrink the file by reducing the tailing free pages, select the **Shrink File** option button and enter the number of tailing free pages you want the file to contain.
5. Click the **Check datafile before action** check box if you want to check the consistency of the data file before altering the file.
6. Click the **Confirm** button. A message saying the file has been altered is displayed.
7. Click **OK**. The message clears.

Changing Tablespace Mode

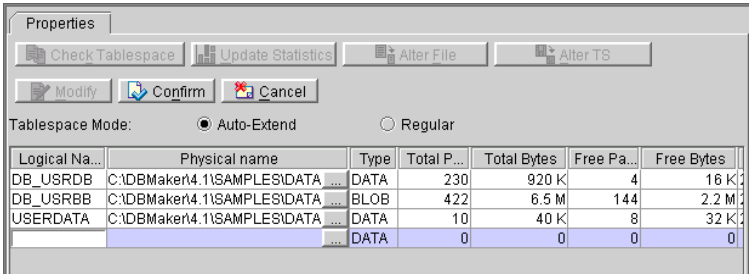
Changing the mode of a tablespace from autoextend to regular or from regular to autoextend is supported in DBMaker. Regular tablespaces generally provide better performance but require close management to prevent them from filling too quickly. Autoextend tablespaces are easier to manage but may become fragmented over time.

➲ **To change the tablespace mode:**

1. Select the tablespace to which you want change the mode. The properties for the tablespace will be displayed in the main console.

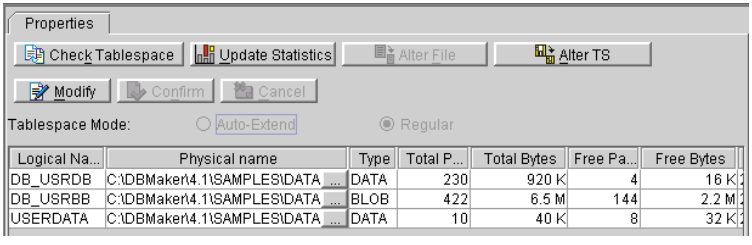


2. Click **Modify**.



3. Select the tablespace mode: **Auto-Extend** or **Regular**.

4. Click **Confirm**.

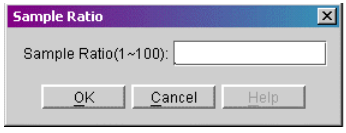


Updating Statistics in a Tablespace

As part of a strategy to optimize the performance of a tablespace, you can update the statistics of a tablespace. The Sample Ratio value denotes the percentage of data in the object that will be used to calculate and update the statistics of the object. If nothing is entered, the default ratio is 100. The sample ratio and the amount of data in the database determine how long it takes to update statistics.

➲ **To update statistics in a tablespace:**

1. Select the tablespace you want to update the statistics of from the tree or the tablespaces page. The **Properties** page is displayed.
2. Click the **Update Statistics** button. The **Sample Ratio** dialog box is displayed.



3. Enter a sample ratio in the range of 1 to 100.
4. Click **OK**. The statistics of the tablespace are updated.

3.4 Dropping a Tablespace

You can drop a tablespace that is no longer required in the database.

➲ **To drop a tablespace:**

1. Select the tablespace you want to delete from the tree.
2. Click the **Drop** button at the top of the **Tablespace** page. Alternatively, right click the tree node of the table and select **Drop Tablespace** from the popup menu. A window will open to confirm that you want to drop the selected table.
3. Click **OK** and the table is dropped.

4 Working With Tables

Tables are the logical unit of storage used by DBMaker to store data. A table consists of several columns and rows. A column can also be called a field or attribute, and a row can be called a record or tuple.

In DBMaker, each table is identified by a unique owner name and table name

This chapter describes:

Creating a table

Modifying table schema

Setting table properties

Granting privileges

Updating table statistics

Editing Data in a table

Renaming a table

Dropping a table

Table and Column Constraint Syntax

NOTE *You can choose to view the system tables by selecting preferences from the* ***Option*** *menu in the menu bar. Select the* ***Show System Tables*** *option button from the* ***Table*** *page in the* ***Preferences*** *window. Refer to Chapter 21for more information..*

4.1 Creating a Table

Tables provide the essential framework for designing a database. When creating a table, you can import schema from other tables belonging to the same database

When creating a table, you will select options from the Create Table wizard. The wizard lets you do the following:

Set table, tablespace, and schema options

Set column options

Set column and table constraints

Create a unique constraint

Create foreign keys

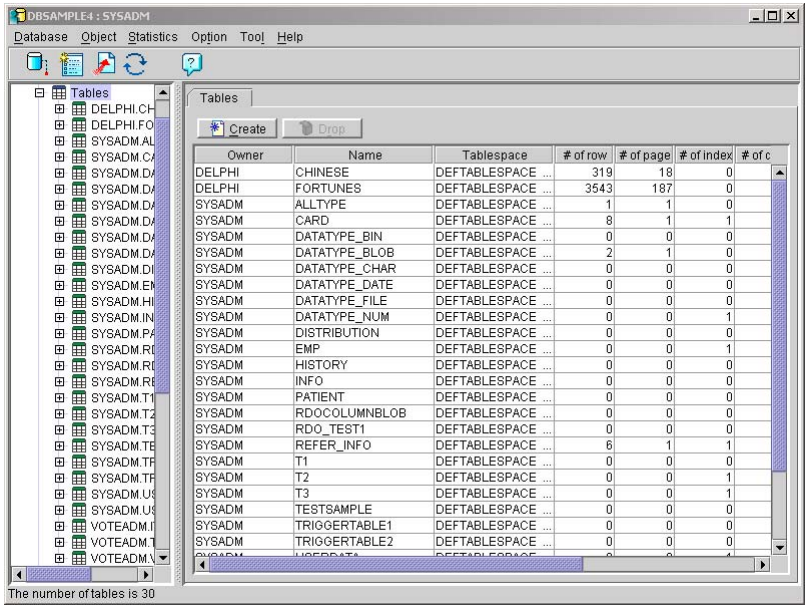
Set advanced table options

Starting the Create Table Wizard

The create table wizard provides an intuitive interface for creating tables. All options for a table are available. Options that are important to set at the time of table creation are presented to ensure that the table is created in the way intended.

➲ **To start the Create Table Wizard:**

1. Select the **Table** object from the tree. The **Table** page will appear.



2. Click **Create**. This will open the **Welcome to Create Table Wizard** window.

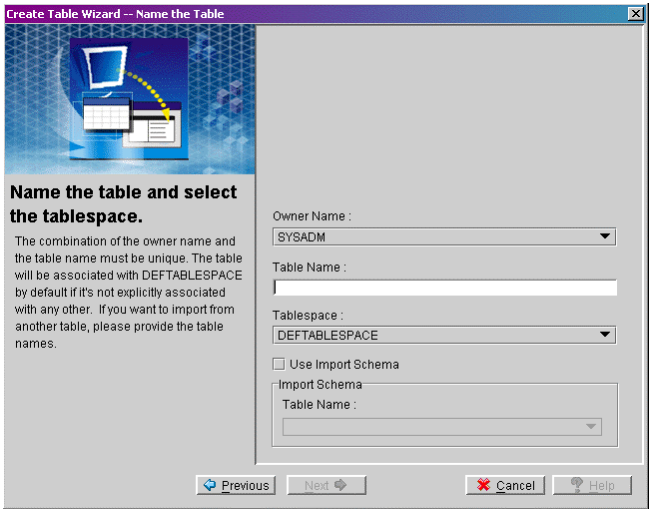
3. Click **Next**. The **Name the Table** window will open.

Setting Table, Tablespace, and Schema Options

Once you have started the **Create Table Wizard**, you can set table, tablespace, and schema options. Table schema may be imported from another table. If you import table schema, all columns (name, length, data type) will initially have the same structure as the table the columns were imported from. The table schema may be modified after has been imported.

➲ **To set table, tablespace, and schema options:**

1. Follow the steps in the Create Table wizard until the **Name the Table** window will open.



2. Select the **Owner Name** from the drop down list. Your user login name is shown by default.

3. Enter a table name in the **Table Name** field.

   NOTE *The table names should be unique for a single owner.*

4. Select a tablespace from the **Tablespace** menu.

   NOTE *By default, tables are placed in the default user tablespace (shown in the **Tablespace** menu as **DEFTABLESPACE**).*

5. To import schema from another table:

   a) Click the **Import Schema** check box if you want the schema of the new table to be the same as the schema of another table in the database.

   b) Select the table to import from in the **Table Name** menu.

6. Click **Next**. The **Add columns to the table** window will open.

Setting Column Options

You can set a variety of column options to meet your data requirements. The settings follow:

Name: Lets you specify a column name.

Type: Lets you specify the following data types: BINARY, CHAR, NCHAR, DATE, DECIMAL, DOUBLE, FILE, FLOAT, INTEGER, LONG VARBINARY, LONG VARCHAR, OID, SERIAL, SMALLINT, TIME, TIMESTAMP, VARCHAR, NVARCHAR , DOMAIN, BLOB, CLOB, and NCLOB. For complete descriptions of the data types, refer to the *SQL Command and Function Reference*.

Precision: Lets you specify the length of the columns. This is necessary for the following data types that have variable length: VARCHAR, NVARCHAR, CHAR, NCHAR, DECIMAL, SERIAL, and BINARY.

Scale: Lets you specify the scale of the data type. This is required if you specify the DECIMAL data type.

Nullable: Lets you restrict or permit null values on your columns. By specifying NOT NULL, null values cannot be added to the column. By default, the column is set as null. NOT NULL is usually specified for primary key columns.

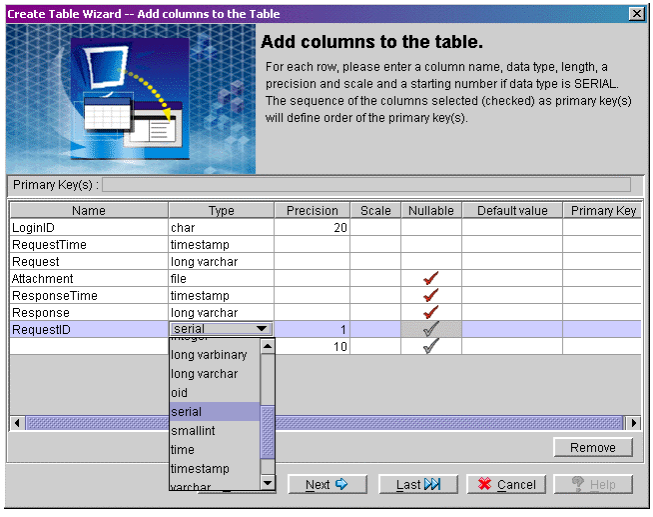
Primary Key: Lets you specify a column as a primary key in a table.

Default Value: Lets you specify a default value for the column data.

➲ **To set column options:**

1. Follow the steps in the Create Table wizard until the **Name the Table** window will open.

2. Enter column information under the respective headings.



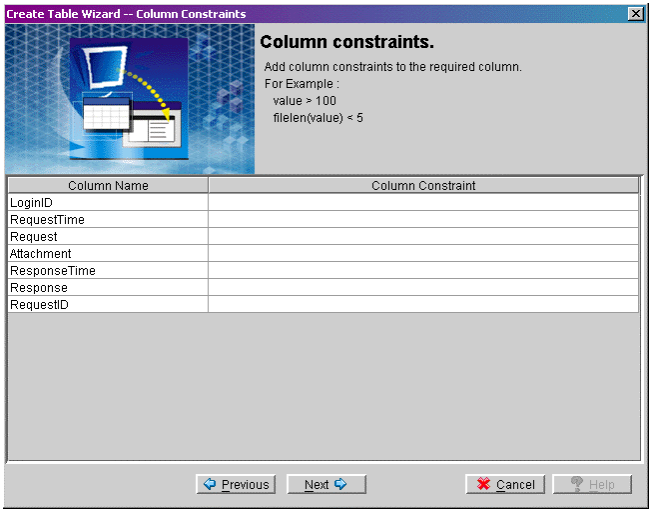
3. Click the **Next** button. The **Column constraints** window will open. If you do not need to add constraints, a foreign key, or set advanced options for the table you may click **Last**; the **Final Review** window will open.

Setting Column and Table Constraints

You can further customize your data requirements by setting column and table constraints. Column and table constraints must follow SQL syntax. See the section on Constraint Syntax at the end of this chapter for more information. Table constraints cannot be altered after the table has been created.

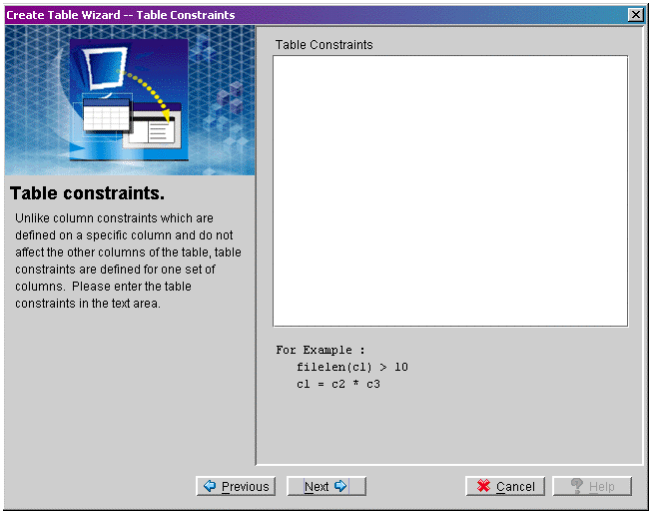
➲ **To set column and table constraints:**

1. Follow the steps in the Create Table wizard until the **Column Constraints** window will open.



2. Enter column constraints in the corresponding row under the **Column Constraint** heading.

3. Click the **Next** button. The **Table Constraint** window will open.



4. Enter table constraints in the **Table Constraints** field.

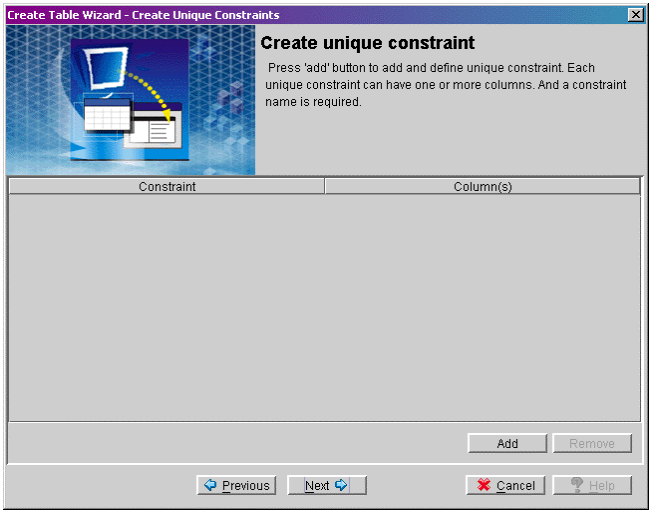
5. Click the **Next** button. The **Create Unique Constraint** window will open.

Adding a UNIQUE Constraint

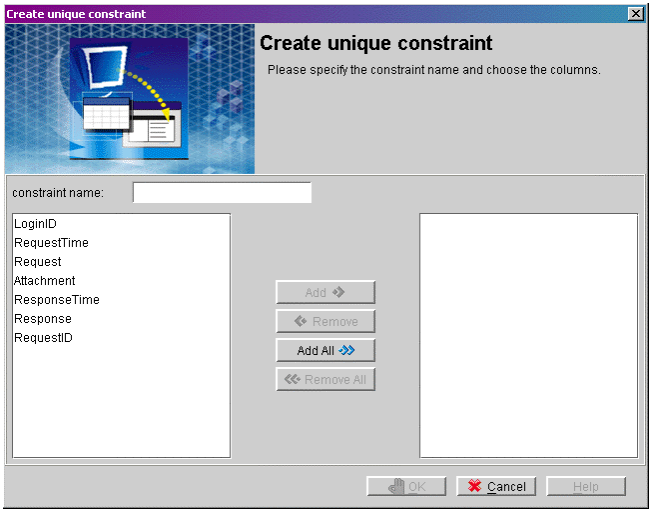
A UNIQUE constraint can be applied to one or more columns of a table at the time it is created. A column that has a UNIQUE constraint cannot have any two tuples with like values, but may contain null values. The UNIQUE constraint must be applied when a column is first created.

➲ **To add a unique constraint**

1. Follow the steps in the Create Table wizard until the **Create Unique Constraint** window will open.



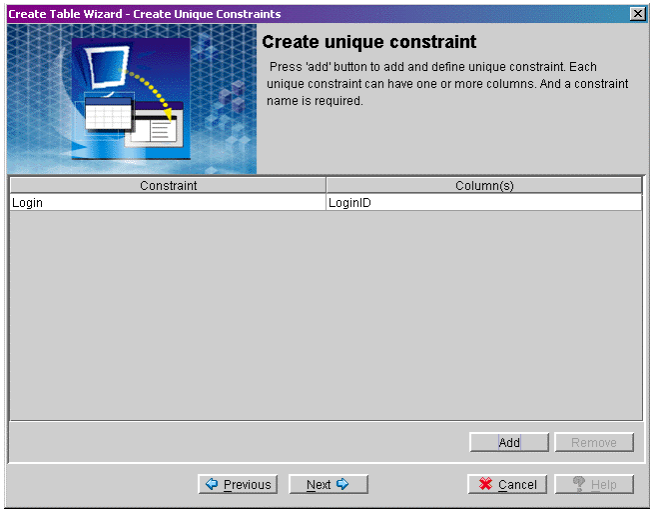
2. Click **Add** to select columns to apply the UNIQUE constraint to. The **Create Unique Constraint** window will open.



3. Enter a constraint name into the **constraint name** field.

4. Select columns to apply the UNIQUE constraint to by double clicking on the column names on the left. Alternatively, use the buttons in the middle to add or remove columns. Selected columns will appear on the right.

5. Click **OK**, the **Create unique constraint** window will reopen.



6. Click Next. The **Create Foreign Key to identify Column(s)** window will open.

Adding a Foreign Key While Creating a Table

You can choose a foreign key as well as select foreign key options. Foreign key columns in the referencing table must map exactly to the primary key columns or unique index columns of the referenced table. The number of columns used in the foreign key and the schema of those columns must be the same in both the referencing and referenced tables.

A foreign key ensures referential integrity between two tables. How it performs this depends on the action chosen for the foreign key. When a record in the referenced column(s) is deleted or updated, the foreign key performs the action on the referencing column(s). The possible actions follow:

Set Null: Records in the foreign key column are set to Null if the user updates / deletes the corresponding records in the primary key or unique index column.

Cascade: The corresponding records in the referenced table are also updated or deleted.

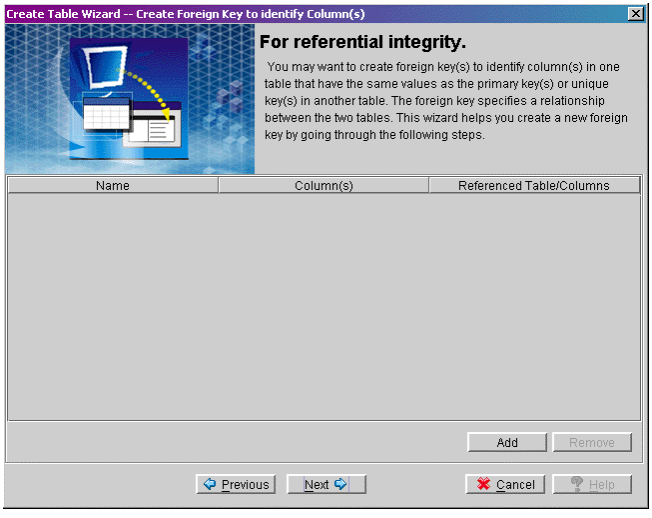
Set Default: The foreign key record is set to a default value if the user updates / delete s corresponding records in the primary key or unique index column.

No Action: There is no action on the foreign key column if the user updates / deletes corresponding records in the primary key or unique index column.

The delete action of a foreign key may be different from the update action.

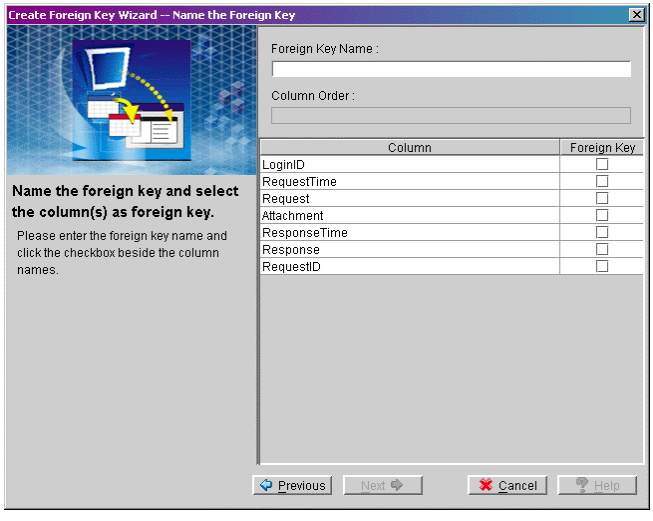
➲ **To add a foreign key during table creation:**

1. Follow the steps in the Create Table wizard until the **Create Foreign Key to identify Column(s)** window will open.



2. Click **Add**. The **Create Foreign Key – Introduction** window will open.

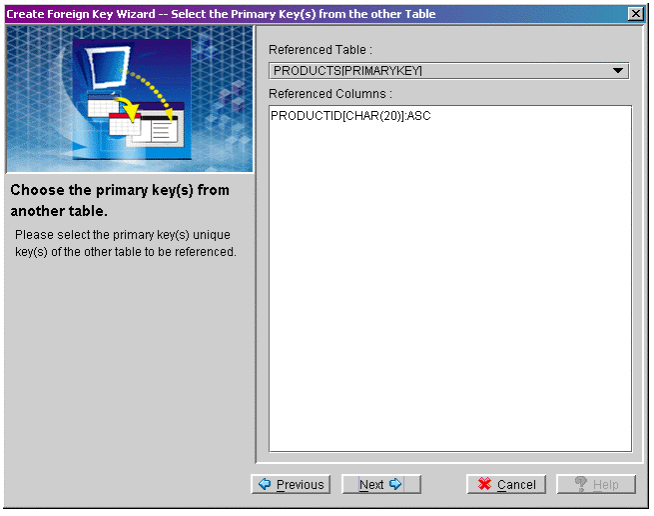
3. Click **Next**. The **Name the Foreign Key** window will open.



4. Enter the foreign key name in the **Foreign Key Name** text field.

5. Click checkboxes beside the column names to indicate which columns to include in the foreign key.

6. Click **Next**. The **Choose the primary key(s) from another table** window of the **Create Foreign Key Wizard** is displayed.

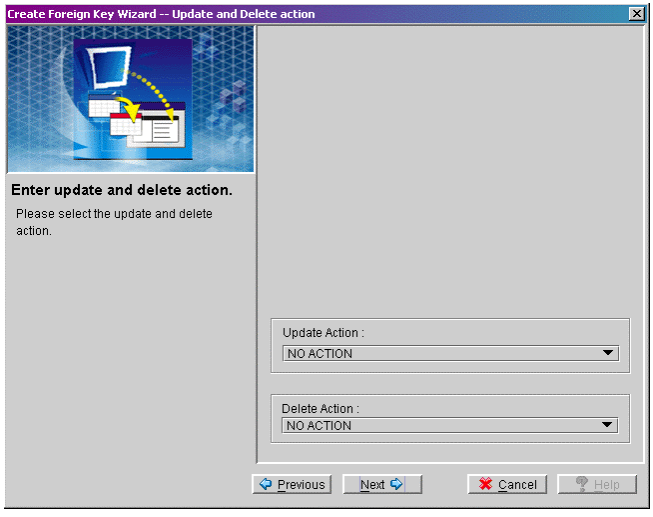


7. Select the name of the referenced table from the **Referenced Table** menu.

NOTE *Only tables containing primary keys or unique indexes that match the schema of the foreign key are shown*

8. All the selected table columns that are primary keys or unique indexes will be displayed.

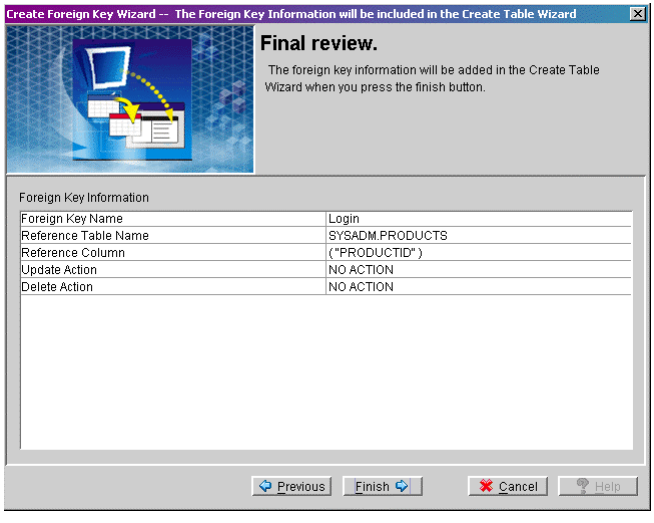
9. Click **Next**. The **Update action and Delete action** window will open.



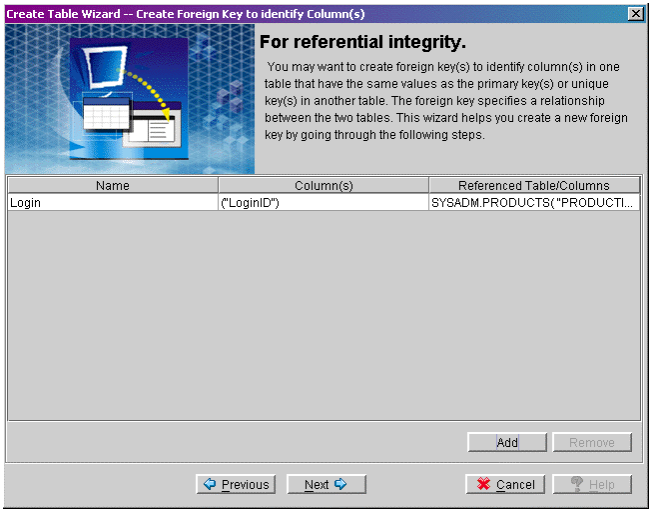
10. Choose a foreign key setting from the **Update Action** menu.

11. Choose a foreign key setting from the **Delete Action** menu.

12. Click **Next**. The **Final Review** window for foreign key creation will open.



13. Click **Finish**. The **Create Foreign Key to identify Column(s)** window reappears.



14. To delete a foreign key:

a) Select the foreign key you want to remove.

b) Click the **Remove** button.

15. Click **Next**. The **Advanced Information** window will open.

Setting Advanced Table Options

For the table you are creating, you can choose various table options that affect the performance of tables in handling data. The options follow:

Temporary / permanent tables: Temporary tables support fast data operations and can be used only by the creator. Temporary tables only exist during a single session, and DBMaker will automatically drop the temporary table when you disconnect from the database. Permanent tables on the other hand exist in the database until the user with the privilege to access and delete the table drops it. The default table type is **Permanent**.

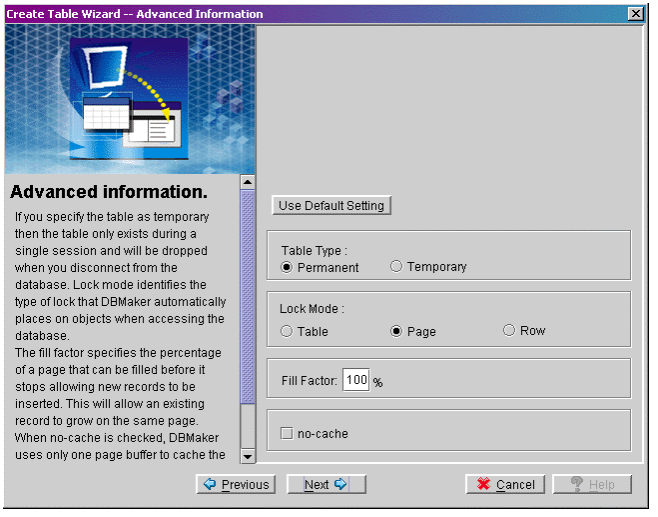
Lock mode: The lock mode of a table identifies the type of lock that DBMaker places on objects when accessing them. DBMaker supports three levels of lock mode: TABLE, PAGE, and ROW. The default lock mode is the **PAGE** lock mode. If the lock mode is set to a higher level (such as TABLE), the level of concurrency on database accesses will be lower, but the required lock resources (shared memory) will be smaller. If the lock mode is set to a lower level (such as ROW), the level of concurrency on database access will be higher, but the required lock resources (shared memory) will be larger.

Fill Factor: The fill factor specifies the percentage of a page that can be filled before it stops allowing new records to be filled. Lowering the fill factor value facilitates the growth of existing records in the page.

No-cache: The no-cache feature is useful when accessing large tables by table scan. Table scans on large tables can cause frequent disk I/O activity. This happens during a table scan on a table with a larger number of data pages than the number of page buffers, which causes all page buffers to be exhausted. If the no-cache option is specified when creating a table, DBMaker only uses one page buffer to cache the data retrieved from a table during a table scan. This prevents the page buffers from being exhausted by only one large table scan.

➲ **To set advanced table options:**

1. Follow the steps in the Create Table wizard until the **Advanced Information** window will open.



2. If you want the table being created to be a temporary table, select the **Temporary** option button.

3. If you wish to change the lock mode, choose a different lock mode from the default setting (page).

   To choose the Table Lock Mode, select the **Table** option button.

   To choose the Row Lock Mode, select the **Row** option button.

4. To change the Fill Factor from the default 100%, enter a percentage in the **Fill Factor** field.

5. Select the **no-cache** check box if you want to use only one page buffer to cache the data retrieved from the table.

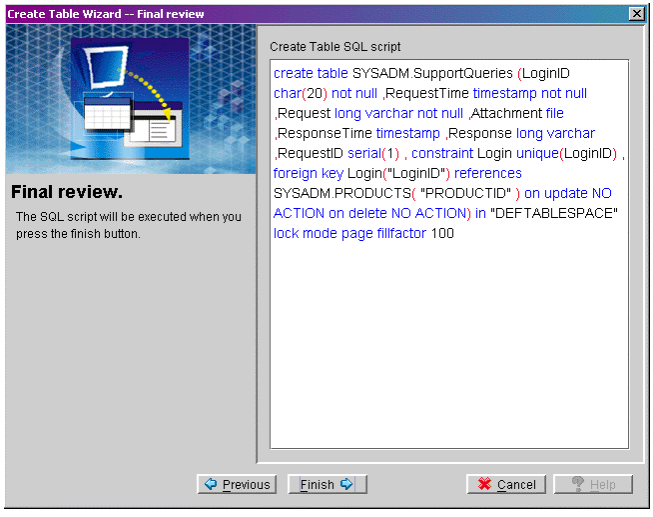
6. Click **Next**. The **Final review** window will appear.

Completing Table Creation

Once you have selected table options from the wizards, you are ready to complete the process of creating a table. The SQL statement that has been generated by the wizard will appear and may be edited by hand. Be aware of SQL99 grammar rules before editing any SQL. A full explanation of the CREATE TABLE statement and syntax may be found in the *SQL Command and Function Reference*.

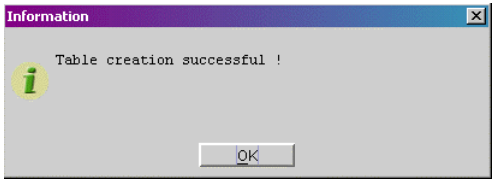
➲ **To complete table creation:**

1. Follow the steps in the Create Table wizard until the **Final Review** window will open.



2. The SQL statement may be modified by hand at this point. Simply use the mouse to place the cursor and insert new text.

3. Review the final SQL script and click the **Finish** button. A confirmation message will appear.



4. Click **OK**. The newly created table will appear in the tree under the **table** node.

4.2 Modifying Table Schema

Once a table is created, you can modify its structure by adding or deleting columns, changing the column order, or modifying schema objects. Some restrictions exist to modifying schema. The following schema objects may be altered:

Column Order

Name

Type

Null / Not Null

Primary Key(s)

Default Value

Column Constraints

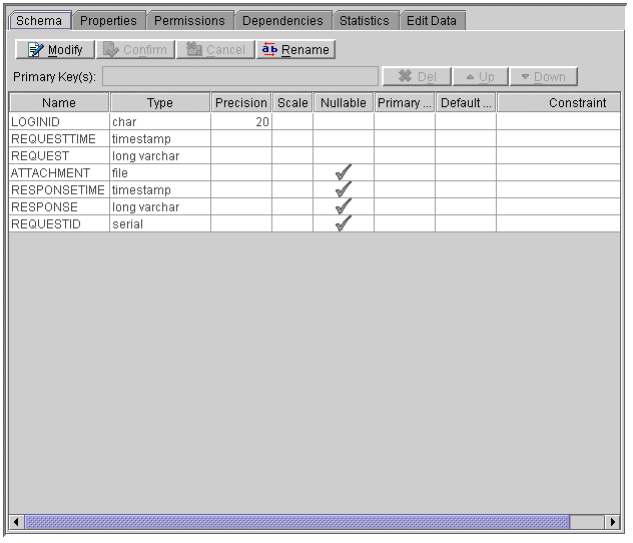
Adding columns

Deleting columns.

➲ **To modify a table's schema:**

1. Click the **Table** node in the tree.

2. From the table page, double-click the table you want to modify. The **Schema** page is displayed.



NOTE *You can also display the* **Schema** *page by opening the table node in the tree and clicking on a table in the tree.*

3. Click **Modify**. The **Del, Up, Down, Confirm** and **Cancel** buttons are activated.

Changing the Column Order

The order of columns in the table determines the table's schema. Changing the column order may be necessary to facilitate the use of foreign keys or table replication.

➲ **To change column order:**

1. Select the column you want to move.
2. Click the **Up** or **Down** buttons to move the column to the desired position.
3. Click **Confirm**.

Changing Column Names

Column names may be changed as long as the new name confirms to naming conventions. Database configuration determines whether column names are case-sensitive.

➲ **To change a column name:**

1. Select the column you want to change the name of.
2. Select the field under the **Name** heading.
3. Type a new name into the field.
4. Click **Confirm**.

Changing Column Type

Not all data types can be converted, for example, CHAR type data cannot be converted to INTEGER type data, DOUBLE type data cannot be converted into FLOAT type data. However, FLOAT type data can be converted into DOUBLE type data. Data conversion follows standard C programming rules.

➲ **To change a column's data type:**

1. Select the column you want to change the data type of.
2. Click the field under the **Type** heading, a drop down menu of available data types will appear.
3. Select the new data type from the drop down menu.
4. Click **Confirm**.

Changing Precision and Scale

The precision (length) of CHAR, DECIMAL and SERIAL type columns may be altered. Be aware that decreasing the scale of a column will result in concatenation at the end of the string. The scale of DECIMAL type data may also be altered.

➲ **To change a column's precision or scale:**

1. Select the column to change the data type.
2. Click the field under the **Precision** heading and enter a new value into the field.
3. Click the field under the **Scale** heading and enter a new value into the field.
4. Click **Confirm**.

Changing Column Null / Not Null

This setting determines whether a column may accept null values. A column can only be changed to Not Null if the table is empty or the GIVE keyword is used to input values into the empty records.

➲ **To make a column nullable/not nullable:**

1. Select the column you want to change.
2. Click the field under the **Nullable** heading. A check mark indicates that null values are allowed for the column.
3. Click **Confirm**.

Changing the Primary Key

Columns may only be selected as primary keys if the data contained in the column is unique.

➲ **To alter the primary keys:**

1. Select the column you want to modify
2. Select the field under the **Primary Key** heading. A key will appear in the field to indicate that it is a primary key. The column name will likewise appear at the end of the list in the **Primary Key(s)** text field at the top of the **Schema** page.
3. Clicking on the field when a key is displayed in the field removes it as a primary key, and likewise removes it from the list.

4. To change the order of the primary keys, remove all primary keys and then add them in the desired order. The order of the primary keys is indicated in the **Primary Key(s)** text field at the top of the **Schema** page.
5. Click **Confirm**.

Changing the Column Default Value

Default values are used to insert a value into the column in the case where the user has not input a value for that column in a record. Default values must be valid for the column data type.

➲ **To change or add a default value for the column:**

1. Select the column you want to change the default value of.
2. Enter a default value for the column that is valid for its data type in the field under the **Default Value** heading.
3. Click **Confirm**.

Adding Column Constraints

Column constraints must follow SQL syntax. See the section "*Constraint Syntax*" at the end of this chapter for more information

➲ **To change or add column constraints:**

1. Select the column you want to modify the constraint of.
2. Enter a new constraint for the column in the field under the **Constraint** heading.
3. Click **Confirm**.

Adding a Column

Tables without data may have columns added to them. Tables containing data or referencing foreign keys may or may not be able to have columns added to them. Foreign keys will be broken if the schema of the referenced table is changed. Refer to

section 4.5 "*Checking Dependencies*" for directions on how to check for referencing tables.

➲ **To add a column:**

1. Enter a column name into the blank field under the **Name** heading at the bottom of the list.
2. Select a data type from the drop down menu in the blank field under the **Type** heading.
3. Enter a precision and/or scale if the data type requires it.
4. Clicking on the blank field under the **Nullable** heading toggles if the column can have null values or not. A check mark indicates that null values are permitted.
5. Clicking on the blank field under the **Primary key** heading toggles if the column is a primary key or not. A key icon indicates that the column is a primary key.
6. Enter a default value into the blank field under the **Default Value** heading at the bottom of the list if one is desired.
7. Enter column constraints into the blank field under the **Constraints** heading at the bottom of the list.
8. Click **Confirm**.

Deleting a Column

Columns may be deleted from a table if they are no longer useful. Referencing foreign keys will need to be rebuilt after changing the column schema.

➲ **To delete a column:**

1. Select the column you want to delete.
2. Click **Delete**.
3. Click **Confirm**.

4.3 Setting Table Properties

You can modify the settings you added when you created a table, and view the table constraints. The following properties may be altered:

Lock Mode: The lock mode of a table identifies the type of lock that DBMaker places on objects when accessing the database. DBMaker supports three levels of lock mode: TABLE, PAGE, and ROW. The default lock mode is the **PAGE** lock mode. If the lock mode is set to a higher level (such as TABLE), the level of concurrency on database accesses will be lower, but the required lock resources (shared memory) will also be smaller. If the lock mode is set to a lower level (such as ROW), the level of concurrency on database accesses will be higher, but the required lock resources (shared memory) will be larger.

Fill Factor: The **Fill Factor** specifies the percentage of a page that can be filled before it stops allowing new records to be filled. The lower the **Fill Factor** value, the greater it facilitates the existing records in the page to grow.

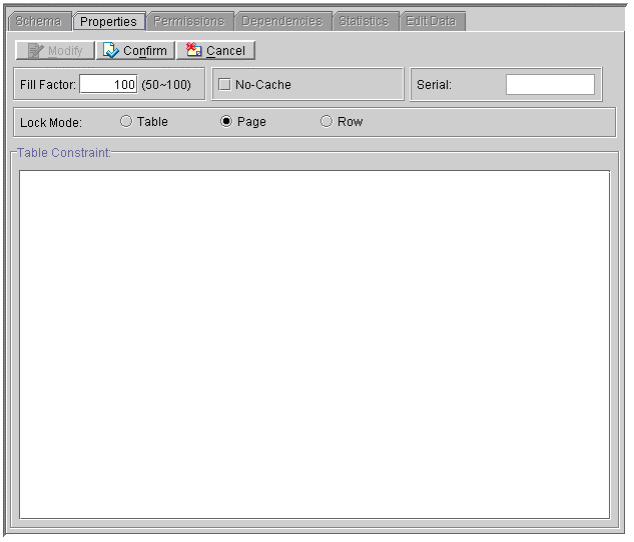
Cache / No-cache: The no-cache feature is useful when accessing large tables by table scan. Table scans on large tables can cause frequent disk I/O activity and can consume all available shared memory. This happens during a table scan on a table with a larger number of data pages than the number of page buffers, which causes all page buffers to be exhausted. Once the no-cache option is specified when creating a table, DBMaker only uses one page buffer to cache the data retrieved from a table during a table scan. This prevents the page buffers from being exhausted by one large table scan.

Serial: Resets the serial number for a serial type column in the table

➲ **To set table properties:**

1. Double-click the **Table** node in the tree.
2. Double-click the table you want to modify. The **Schema** page is displayed.

3. Select the **Properties** tab. The **Properties** page is displayed.



4. If you wish to change the lock mode, choose a different lock mode from the default setting.

To choose table lock mode, select the **Table** option button.

To choose row lock mode, select the **Row** option button.

5. To change the fill factor, enter a percentage in the **Fill Factor** field.

6. Select the **no-cache** check box if you want only one page buffer to cache the data retrieved from the table.

7. To reset the serial number for a serial type column in the table, enter the starting number of the column in the **Serial** box.

8. Click **Confirm**. The properties of the table are modified.

4.4 Granting/Revoking Table Privileges

To allow other users or groups to access tables in the database, you can grant privileges to them. You can also revoke privileges to deny groups and/or users access to tables. Privileges can be granted to specific functions (Insert, Update, Delete, Index, Reference) or to specific columns of the table.

Users with resource or DBA privilege cannot have their privileges revoked on objects they own. Only by changing the user's authority can these privileges be revoked.

The following list summarizes table privileges:

Select: Users have permission to select any records from the table.

Delete: Users have permission to delete any records from the table.

Insert: Users have permission to insert new records into the table or column.

Update: Users have permission to update any records in the table or column.

Alter: Users have permission to alter the schema of the table.

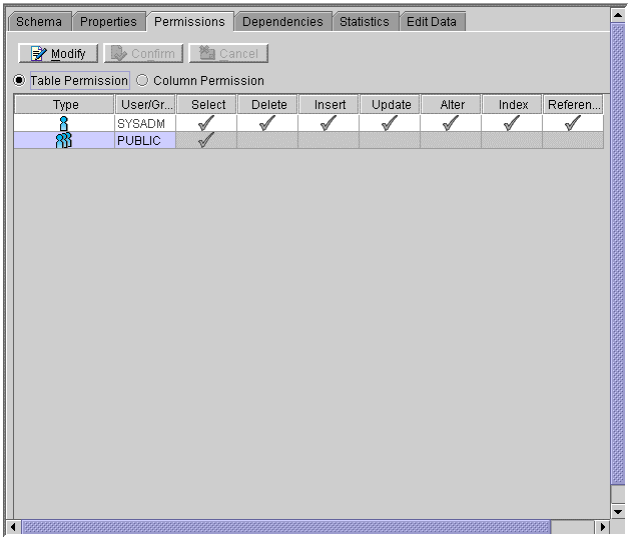
Index: Users have permission to create an index on the table.

Reference: Users have permission to create referential constraints on the table or column.

➲ **To grant/revoke privileges on tables:**

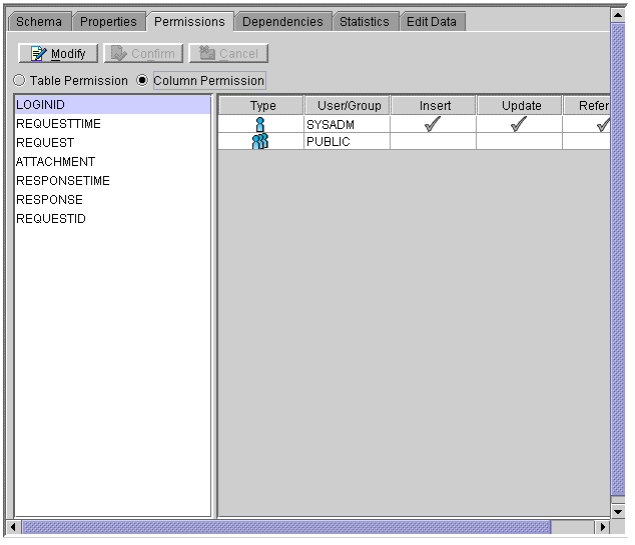
1. Select the table in which you want to add the privileges of which are to be granted to the other users/groups. The **Schema** page is displayed.

2. Click the **Permissions** tab at the top of the window. The **Permissions** page is displayed.



3. Click **Modify**. The **Confirm** and **Cancel** buttons are now active
4. Select one of the permissions in the cells for the user/group.
5. To revoke a privilege, click any of the checked columns.

6. To grant or revoke permission on columns, click the **Column Permissio**n option button. The window appears as below.



7. Select any of the following permissions in the cells for your user/group:

Insert: users have permission to insert new records into the column.

Update: users have permission to update any records in the column.

Reference: users have permission to create referential constraints on the column.

8. To revoke a privilege, click any of the checked columns.

9. Click **Confirm** to execute all changes.

4.5 Checking Dependencies and Statistics

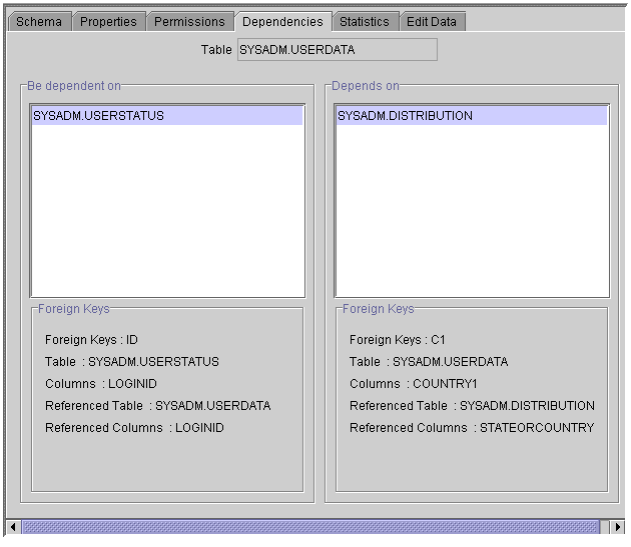
Before handling data in a table or modifying a table's schema, it is recommended that you check dependencies and refer to table statistics.

Checking Dependencies

You can check table names, owners, as well as foreign and primary keys in a table.

➲ **To check dependencies:**

1. Select the table that you want to check the dependencies of. The **Schema** page will appear.
2. Click the **Dependencies** tab at the top of the window. The names of tables that foreign keys identify as referencing the selected table will appear in the **Be dependent on** column. The names of tables that are referenced by the selected table will appear in the **Depends on** column.



3. Selecting a table from the **Be dependent on** list displays foreign key information on the bottom.

4. Selecting a table from the **Depends on** list displays foreign key information on the bottom.

5. Double clicking on the table name on either list to changes the selected table in the **Table** field at the top of the page.

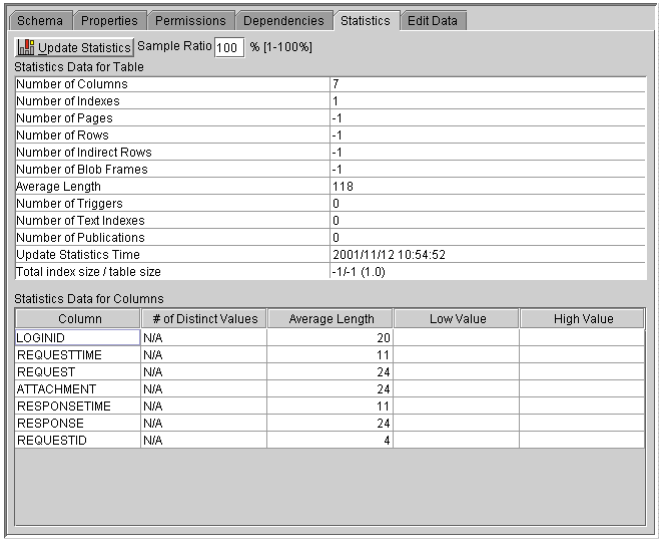
Checking and Updating Table Statistics

You can check statistics both for tables and columns. You can also alter the sample ratio and use the Update Statistics function.

➲ **To check and update table statistics:**

1. Select the table that you want to check the statistics of. The **Schema** page will appear.

2. Select the **Statistics** tab at the top of the window. The statistics will be displayed as shown below.



3. If you want to change the percentage of the data used to calculate updated statistics, select the **Sample Ratio** box and enter a sample ratio percentage.

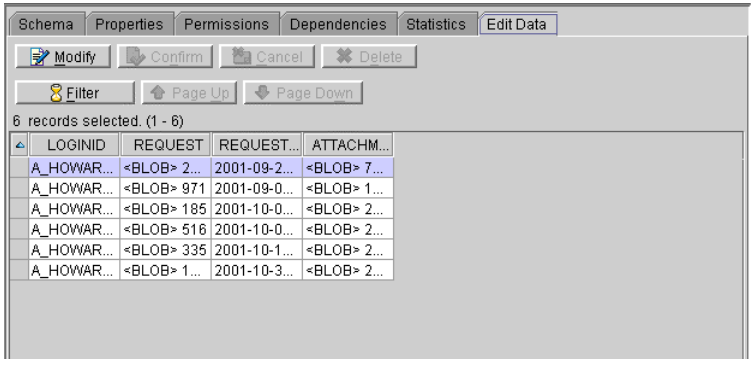
4. Click **Update Statistics**. The updated statistics will appear.

4.6 Editing Data in a Table

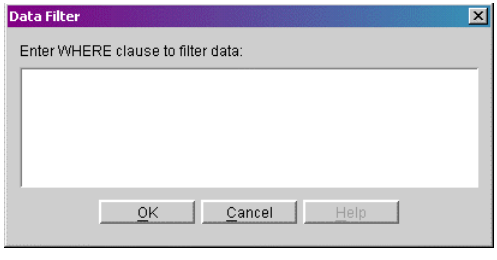
Once records are added to a table, you can update the existing records in the table. You must have object privilege for a table to edit it; refer to *Granting/Revoking Table Privileges* or *Granting and Removing Object Privileges* for information on how to manage user privileges on tables and columns.

➲ **To edit data in a table:**

1. Double-click the **Table** node in the tree.
2. Double-click the table you want to modify. The **Schema** page is displayed.
3. Click the **Edit Data** tab on top of the window. The **Edit Data** page is displayed.



4. To filter the data, click **Filter**. The **Data Filter** window will open where you can specify a WHERE condition.



5. Enter a condition by which the data is to be filtered and Click **OK**. The main window for the table is displayed.

NOTE *WHERE conditions must follow proper SQL syntax. See the section on Constraint Syntax at the end of this chapter for more information.*

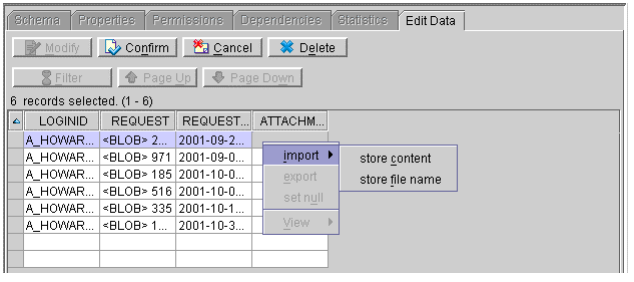
6. To update data,

a) Locate the data on the screen using the **Page Up** and **Page Down** buttons.

b) Click **Modify**.

c) Select the data on the screen.

d) Edit the data. If the column contains BLOB data, a pop-up menu will appear with the option to import, export, set null, or view. BLOB type columns allow content to be stored, FILE type columns give the option to store content (as a system file object) or store the file name (creating a user file object). Export allows the user to output BLOB or system file objects. Set null deletes the BLOB or file object. View opens the BLOB or file object in an application dependent on the view chosen.



e) Click **Confirm**. The data is edited in the selected table.

7. To delete data:

a) Locate the data on the screen using the **Page Up** and **Page Down** buttons.

b) Click **Modify**.

c) Select the data on the screen.

d) Click **Delete**.

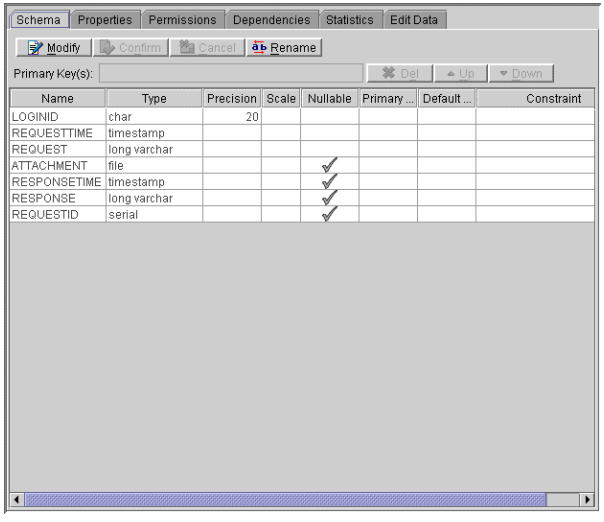
e) Click **Confirm**. The data is deleted from the selected table.

4.7 Renaming a Table

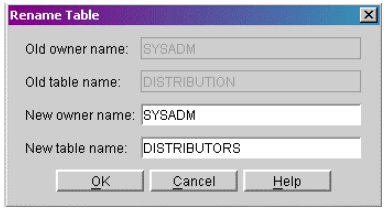
A table may be renamed any time after its creation. Be sure to check dependencies before renaming a table. Renaming a table will remove all dependencies on the table.

➲ **To rename a table:**

1. Expand the table node of the tree.
2. Select the table to change the name of from the tree. The table Schema page will appear.



3. Click **Rename**. The **Rename Table** window will open.



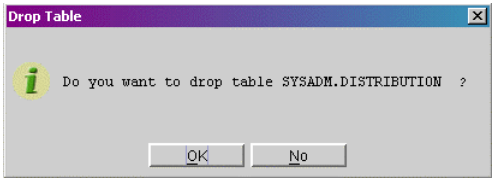
4. Enter a new owner name and table name into the appropriate fields and click **OK**.

4.8 Dropping a Table

You can drop a table that is no longer required in the database. Be aware of the impact that dropping a table may have on foreign keys that reference the table.

➲ **To drop a table:**

1. Select the table you want to delete from the tree.
2. Click **Drop**. Alternatively, right click the tree node of the table and select **Drop Table** from the popup menu. A window will open to confirm that you want to drop the selected table.



3. Click **OK** and the table is dropped.

4.9 Constraint Syntax

The following table lists comparators that can be used with the VALUE key word to create valid table, column, domain, or replication constraints. These comparators may also be used to create WHERE conditions for filtering data or creating views, or WHEN conditions for creating triggers.

You can combine simple conditions with the logical operators AND, OR, and NOT to form compound conditions. You can use the keyword AND to combine two search conditions that must both be true. You can use the keyword OR to combine two search conditions when one or the other (or both) must be true. Finally, you can use the keyword NOT to select rows where a search condition is false

COMPARATOR	DESCRIPTION
Relational Operators	These may be one of the following: >, >=, <=, <, =, and <>. The relational operator condition is satisfied when the expression on either side of the relational operator fulfills the relation set up by the operator.
BETWEEN	This comparison takes the form: x BETWEEN y AND z. The BETWEEN condition is satisfied when the value or expression to the left of the BETWEEN keyword lies in the range (denoted by the AND keyword) of the two expressions on the right of the keyword.
IN	This comparison takes the form: x IN (y, z, ...). The IN condition is satisfied when the value or expression to the left of the IN keyword is included in the list of values to the right of the keyword.
IS NULL	This takes the form: x IS NULL. The IS NULL condition is satisfied when the value or expression to the left of the IS NULL keywords is a NULL value.
IS NOT NULL	This takes the form: x IS NOT NULL. The IS NOT NULL condition is satisfied when the value or expression to the left of the IS NOT NULL keywords contains a value other than a NULL value.

COMPARATOR	DESCRIPTION
LIKE	This takes the form: x LIKE 'y' ESCAPE 'z'. The LIKE condition is satisfied when the string value or expression to the left of the LIKE keyword meets the criteria specified in the case-sensitive quoted string to the right of the keyword. You can use the percent symbol (%) and the underscore symbol (_) as wildcards in the quoted strings. The percent symbol matches zero or more characters, and the underscore symbol matches exactly one character. The ESCAPE clause is optional and allows you to define an escape character so you can include the percent and underscore symbols in the quoted string without having them interpreted as wildcards. To include a single-quote character in a quoted string, use two consecutive single-quotes.
MATCH	This takes the form: x NOT CASE MATCH 'y'. The MATCH condition is satisfied when the quoted string to the right of the MATCH keyword matches the entire string value or expression to the left of the keyword. The NOT and CASE keywords are optional. The NOT keyword inverts the search results, and the CASE keyword makes the search case-sensitive.
CONTAIN	This takes the form: x NOT CASE CONTAIN 'y'. The CONTAIN condition is satisfied when the quoted string to the right of the CONTAIN keyword matches any part of the string value or expression to the left of the keyword. The NOT and CASE keywords are optional. The NOT keyword inverts the search results, and the CASE keyword makes the search case-sensitive.

Table 4-1 Constraint Syntax Comparators

.

5 Working with Views

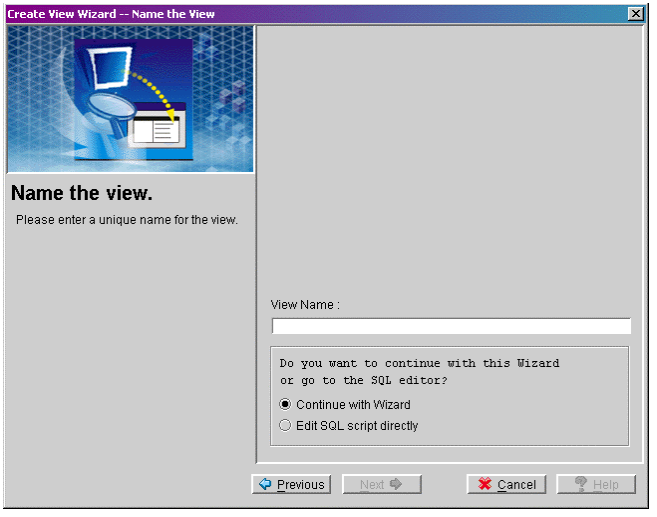
DBMaker provides the ability to define a virtual table, called a *view*. The view definition is stored in the database, but the actual data that you see in the view is not physically stored anywhere. Rather, the data is stored in the base tables from which the view's rows are derived. You can create views for flexible data queries on tables. You can use views to browse data in multiple tables. JDBA Tool allows you to create, update, and drop views.

5.1 Creating a View

Once data has been entered in the database tables, you can use the tables to create views.

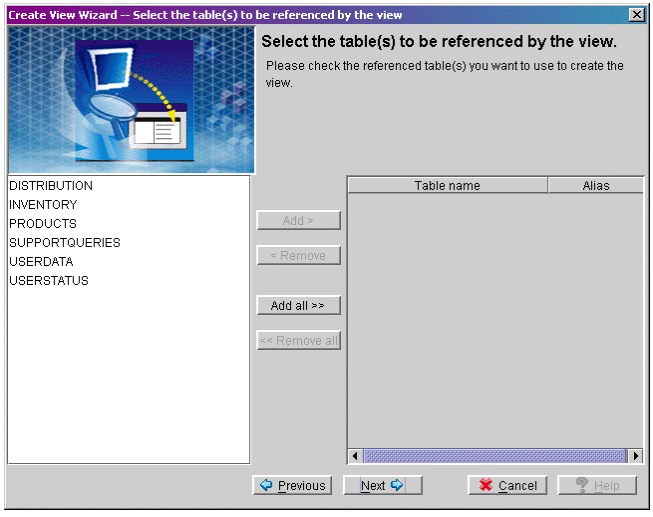
➲ **To create a view:**

1. Select the **View** object from the tree. The owner name, view name, and time of creation of all views on the database will be displayed on a table in the **View** page.
2. Click **Create**. The **Create View Wizard – Introduction** window is displayed.
3. Click **Next**. The **Name the View** window is opened.



4. Enter the name of the view in the **View Name** field.
5. Select the **Continue with Wizard** option button to continue with the wizard. Select the **Edit SQL script directly** option button to jump to the last step of the wizard (see step 13 below).

6. Click the **Next** button. The **Select the table(s) to be referenced by the view** dialog box is displayed.



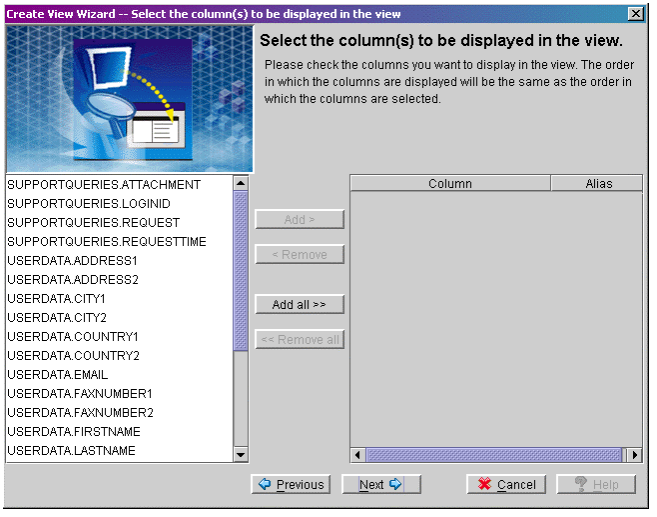
7. Select tables to use in the view by double clicking on the table names on the left. Alternatively, use the buttons in the middle to add or remove columns. Selected tables will appear on the right.

8. A table can be added to a view more than once. If this is done, an alias for the table must be given. The alias is a new name for the table that allows for columns in the table to be referenced more than once. To add an alias:

a) Select a table that has already been added to the view.

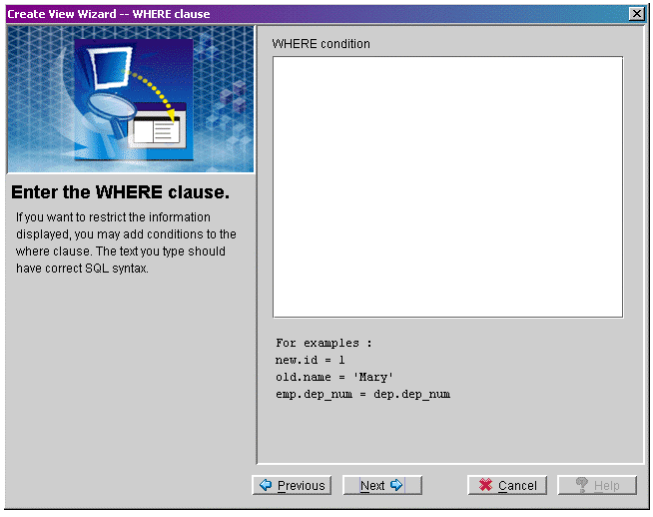
b) Enter an alias name into the field under the **Alias** heading. The default name appears as **TableAliasName#**.

9. Click **Next**. The **Select the column(s) to be displayed in the view** window will open.



10. Select columns to include in the view by double clicking on the column names on the left. Alternatively, use the buttons in the middle to add or remove columns. Selected columns will appear on the right. Columns added more than once must have an alias. Column aliases are created the same way as table aliases.

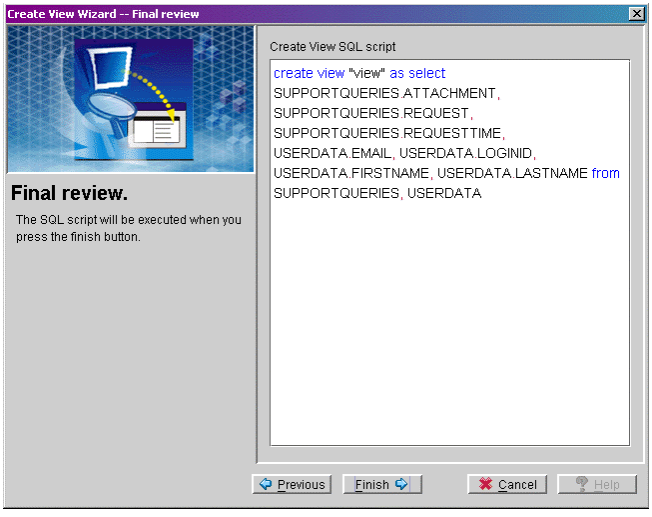
11. Click **Next**. The WHERE clause window will open.



12. Enter a condition by which the data in the view is to be filtered.

NOTE *WHERE statements must follow SQL syntax. See the section on Constraint Syntax at the end of chapter 4 for more information.*

13. Click **Next**. The **Final review** window will open.



14. If you wish to alter the view data after reviewing the SQL script, enter new script into the **Create View SQL script** field or click the **Previous** button to display previous screens in the wizard.

15. Click the **Finish** button after reviewing the SQL script.

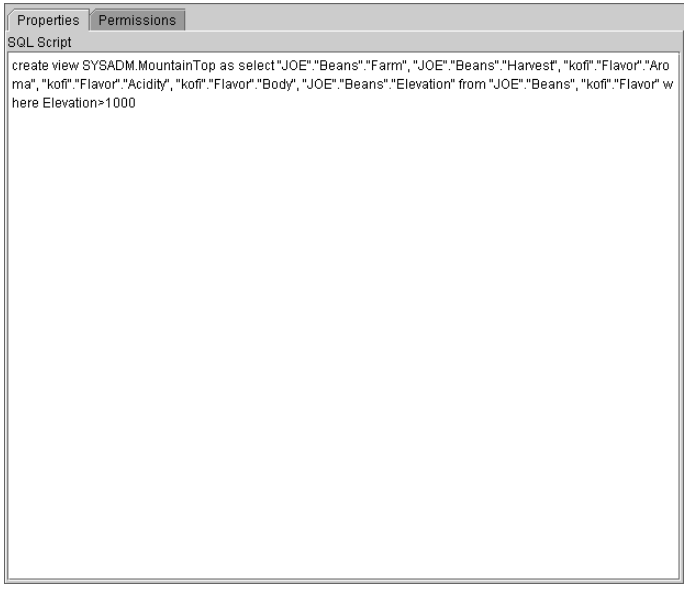
16. A conformation message appears. Click **OK**.

5.2 Checking View Properties

You can check the properties of a view including the tables and columns used as well as ownership.

➲ **To check view properties:**

1. Click the **View** object in the tree. This will display all the views in the database.
2. Select the view that you want to see the properties of. The **Properties** page will open.

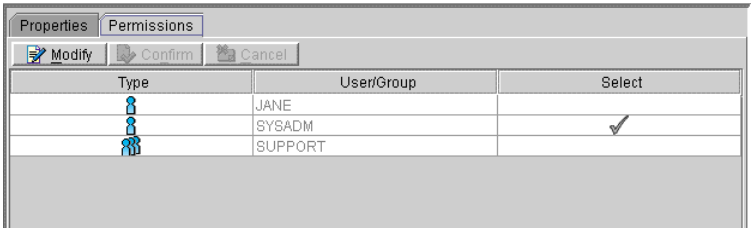


5.3 Granting View Privileges

You can allow other users/groups to access your view by granting them privileges. You can also revoke access to a view for a user/group at any point of time.

➲ **To grant view privileges:**

1. Select the **View** object in the tree.
2. Select the view that you want to change user privileges for. The **Properties** page appears.
3. Click the **Permissions** tab on the top of the window. The permission details will be displayed.



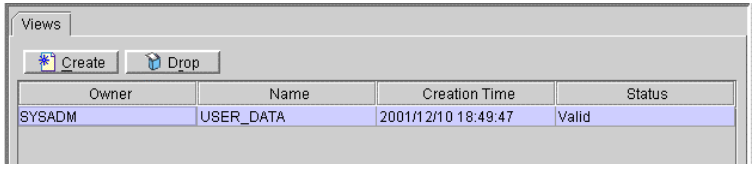
4. Click the **Modify** button.
5. Select the user/group that you want to grant or revoke select privilege to the view.
6. Click the **Select** column next to the selected User/Group. A check mark indicates that the user or group has permission on the view.
7. Click the **Confirm** button.

5.4 Dropping a View

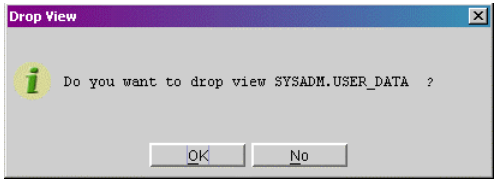
If the view is no longer required, it may be deleted by following the procedure given below.

➲ **To drop a view:**

1. Click the **View** object in the tree. This will display all the views in the database.
2. Select the view you intend to drop. The view is selected.



3. Click the **Drop** button at the top of the window. The **Drop View** dialog box is displayed.



NOTE *Alternatively, right click the view to be dropped from the tree, a pop-up menu will open. Select* ***Drop*** *from the pop-up menu.*

4. Click **OK**. The message clears. The remaining views in the database will be displayed.

6 Working with Indexes

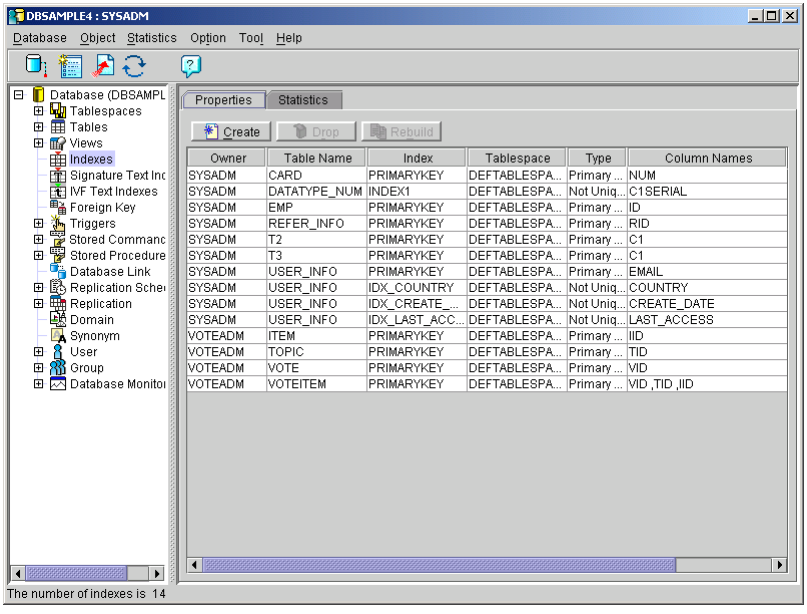
An index is used to facilitate random access of table rows; it is a sub-table of the table containing the accessing rows. The index table contains the key columns of its parent table. Indexes are particularly useful for tables that hold large amounts of data, or tables that are frequently accessed. An index can be *unique* or *non-unique*. In a unique index, no more than one row can have the same key value, with the exception that any number of rows may have NULL values.

6.1 Creating an Index

An index should be created when one or more rows in a table will be used in search queries. Indexing the rows will also allow sorting of data to be performed.

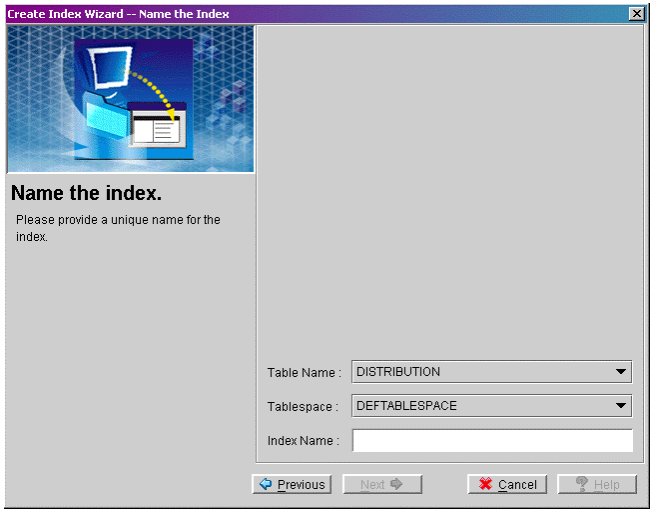
➲ **To create an index:**

1. Click the **Indexes** object in the tree. The indexes in the database are listed.



2. Click the **Create** button. The **Create Index Wizard** is displayed.

3. Click the **Next** button. The **Name the Index** window is displayed.

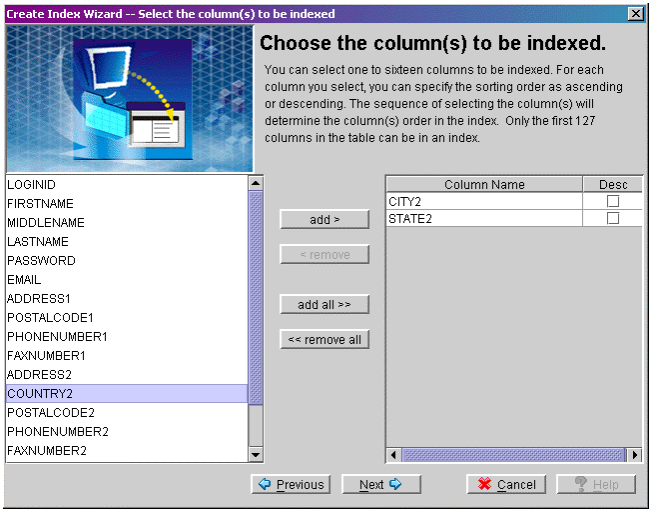


4. From the **Table Name** menu, select the table name that the index will refer to.

5. Select a tablespace to build the index in. Building an index on a different tablespace may result in better disk I/O, especially if the tablespaces reside on separate disks.

6. Enter a name for the Index in the **Index Name** field.

7. Click the **Next** button. This will open the **Select the column(s) to be indexed** window.



8. A list of available columns on the selected table appears on the left hand list box. Select the columns that you want to use in the index by double clicking on the column name or selecting the column and clicking the **Add** button. The **Add all** button can be used to add all columns to the index. All the columns selected for the creation of the index will be displayed in the list box to the right. Columns in the index can be likewise removed by double clicking on the column name or selecting the column and clicking the **Remove** button.

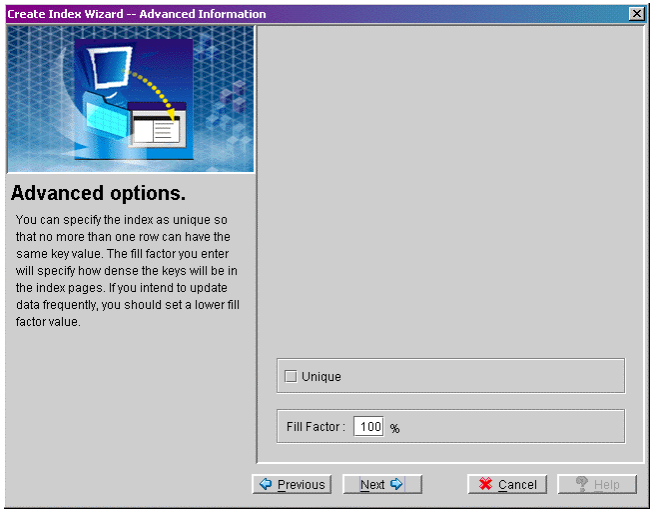
9. To change the sorting order to descending for a column:

a) Select a column from the list box to the right.

b) Click the cell in the **Desc** column for the selected column in the list box to the right. A check mark in the box indicates that the order for that column's index is descending

NOTE *By default, the sort order is ascending.*

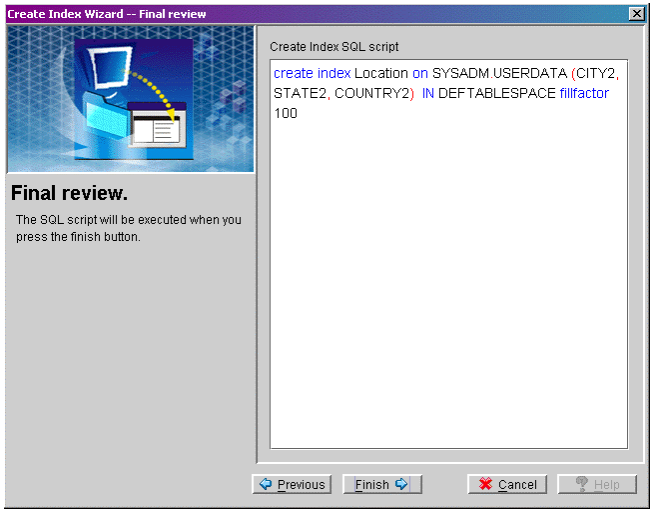
10. Click the **Next** button. The **Advanced Information** window appears.



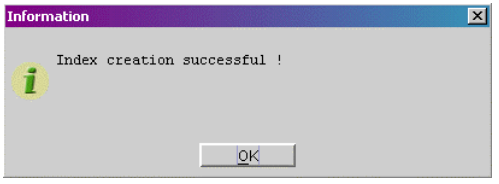
11. If you want the index to be unique, select the **Unique** option button.

12. To change the default fill factor, enter a percentage in the **Fill Factor** field.

13. Click the **Next** button. The **Final review** dialog box is displayed.



14. Click the **Finish** button after checking that the SQL script is correct. The **Information** dialog box appears.



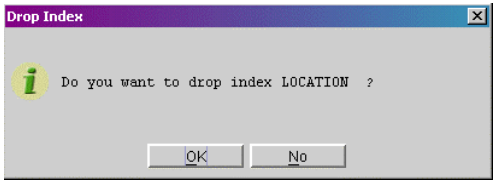
15. Click **OK**.

6.2 Dropping an Index

You can drop an index that is no longer required in a database.

➲ **To drop an Index:**

1. Click the **Index** object in the tree. The indexes in the database are listed.
2. Select the index to be dropped and click the **Drop** button. The **Drop Index** dialog box is displayed.



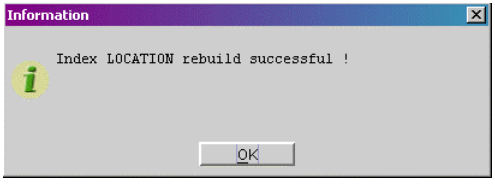
3. Click **OK**. All the remaining indexes in the database will be displayed on the **Properties** page.

6.3 Rebuilding an Index

If you have made a number of changes to the table, you should rebuild the index associated with that table. Through the rebuilding process, the old index is deleted and a new one is created.

➲ **To rebuild an index:**

1. Click the **Index** object in the tree. A list of all indexes in the database is displayed on the **Properties** page.
2. Select the Index that has to be rebuilt.
3. Click **Rebuild**. The **Information** message box is displayed.



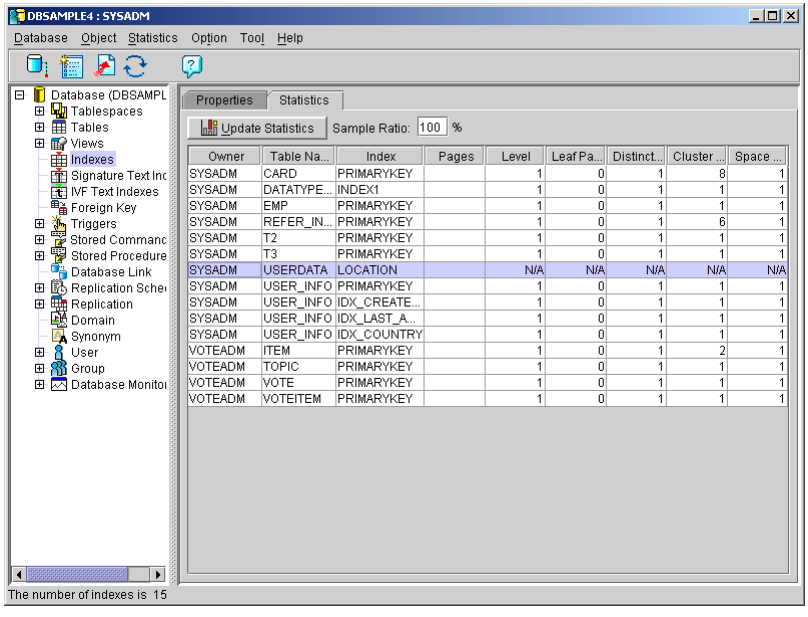
4. Click **OK**.

6.4 Updating Index Statistics

If your database is extremely large, it will take a lot of time to update the statistics values of all schema objects. An alternative method is to update statistics on only those specific schema objects that have been modified since the last update. You can also set the sampling rate, which is the percentage of the records used to calculate the statistics.

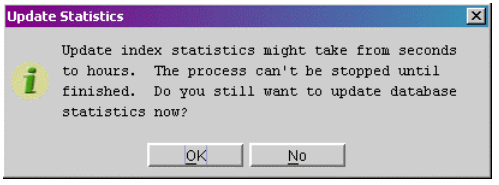
➲ **To update index statistics:**

1. Click the **Index** object in the tree. A list of all the indexes in the database is displayed.
2. Click the **Statistics** tab. The **Statistics** page is displayed.

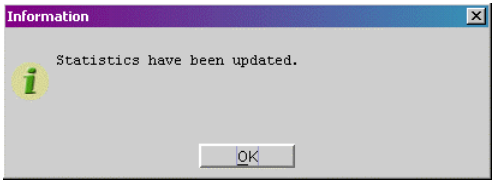


3. If you want to change the percentage of the data used to calculate updated statistics, select the **Sample Ratio** box and enter a percentage to sample.

4. Click the **Update Statistics** button. The following window is displayed.



5. Click **OK**.



6. Click OK.

7 Working with Signature Text Indexes

A text index is a mechanism that provides fast access to rows of a table containing columns with one or more words or phrases as text. Text indexes contain a representation of all the text found in the text columns on which they are based. However, the data is encoded and structured to make retrieval much faster than directly from the table. Once you create a text index on a table, its operation is transparent to users of the database. The DBMS uses the text index to improve full-text query performance whenever possible.

Signature text indexes are created within the same tablespace as the columns that the index is being created for. Signature text indexes are well suited for building on columns that contain a combined total of 200 or fewer MB of data. For larger text indexes, consider building an inverted file (IVF) text index. For detailed information on IVF text indexes, refer to *Working with IVF Text Indexes*.

7.1 Creating a Signature Text Index

Signature text indexes can be built on all character type columns, including CHAR, VARCHAR, LONG VARCHAR, NCHAR, NVARCHAR, NCLOB, and FILE types. A table can have multiple text indexes, and text indexes can be built on multiple columns.

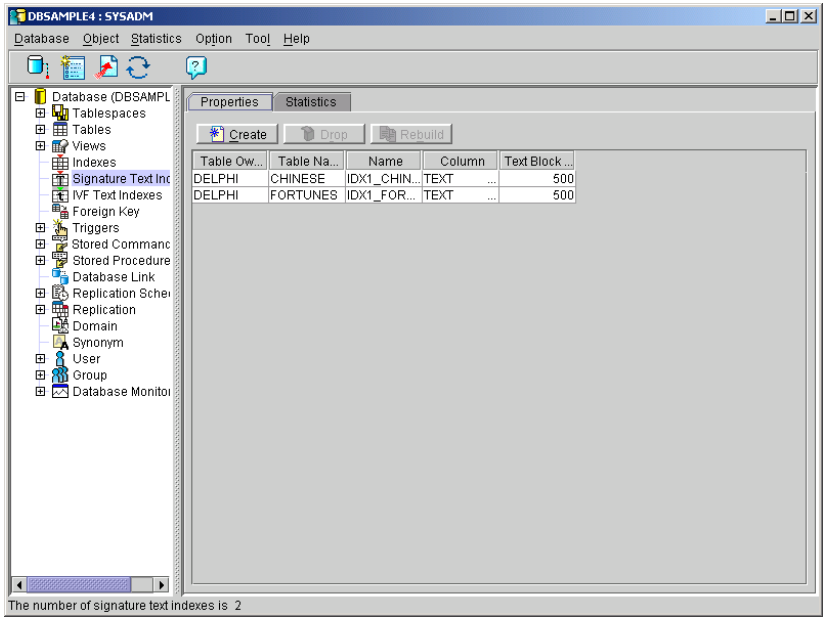
To ensure optimal performance for a signature text index, specify the total text size and the scale. The total text size is an approximation of the total number of megabytes of storage that are used by the columns on which the text index is to be built. The total text size can range from 10-200 MB, and the default total text size value is 32 MB.

The scale is the relative size of the text index to the total text size expressed as a percentage. Generally the larger the scale the better the query performance. The default scale is 50 percent.

Text indexes may be sorted by using the **order by** feature.

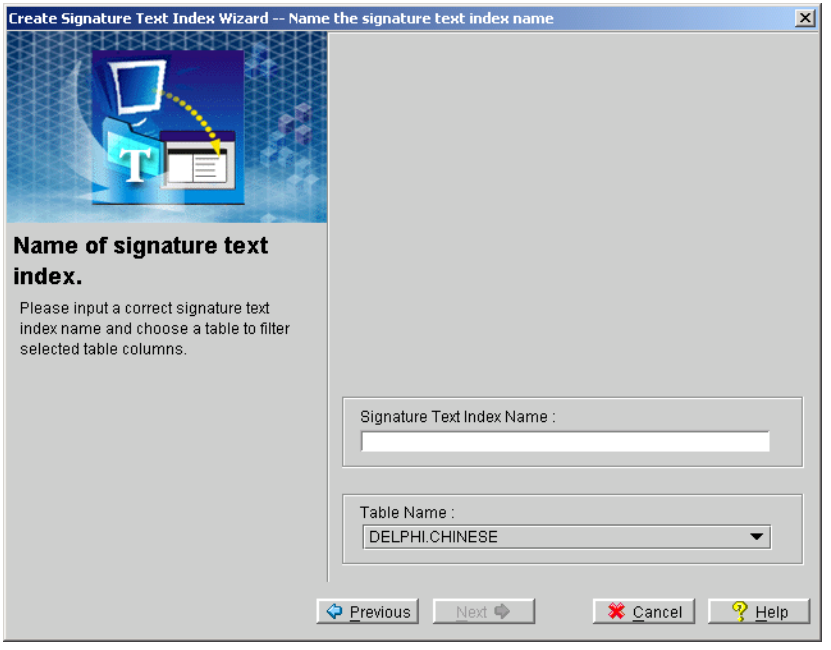
➲ **To create a Signature Text Index:**

1. Click the **Signature Text Index** object in the tree. The **Properties** page appears.



2. Click **Create**. The **Introduction** window for the **Create Signature Text Index Wizard** is displayed.

3. Click **Next**. The **Name the signature text index** window is displayed.

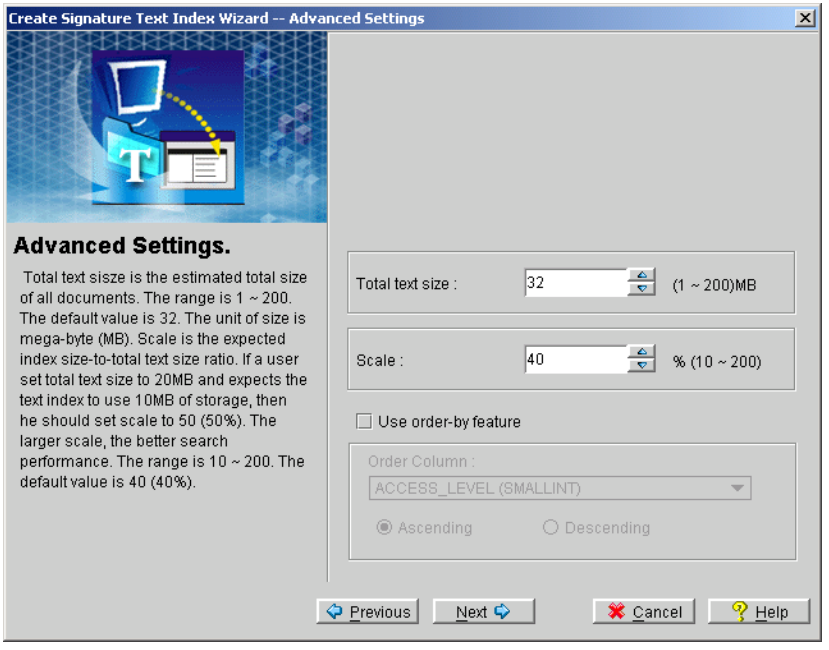


4. Select the name of the table from which the text index is to be created from the **Table Name** menu.

5. Enter the name of the text index in the **Text Index Name** field.

6. A list of available columns on the selected table appears on the left hand list box. Select the columns that you want to use in the signature text index by double clicking on the column name or selecting the column and clicking the **Add** button. The **Add all** button can be used to add all columns to the index. All the columns selected for the creation of the text index will be displayed in the list box to the right. Columns in the text index can be likewise removed by double clicking on the column name or selecting the column and clicking the **Remove** button.

7. Click **Next**. The **Advanced Settings** window will open.



8. To change the signature text index settings:

To change the total text size, enter a value into **Total Text Size** between 1 and 200.

To change the scale, enter a value into the **Scale** field between 10 and 200.

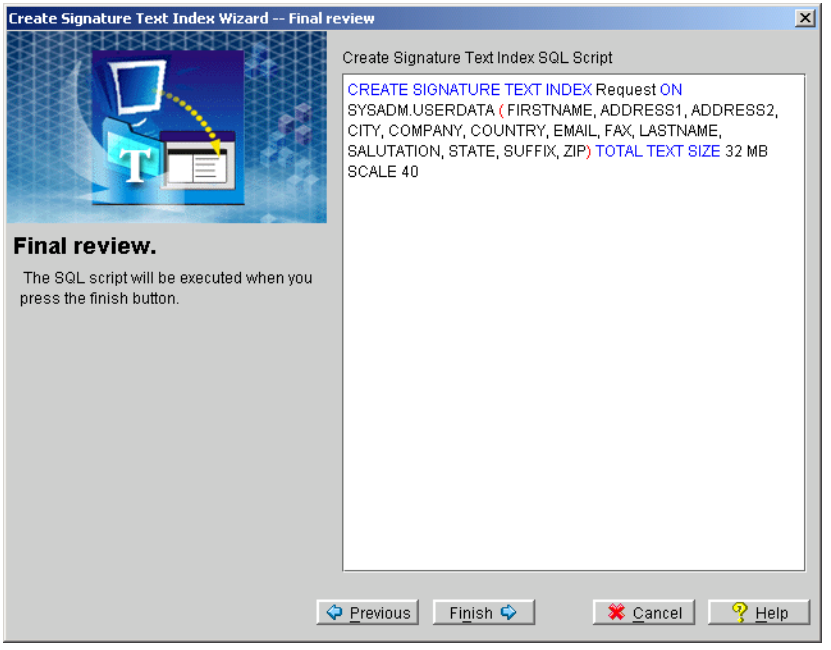
9. To present data as it relates to another column in the table select the **Use order-by feature** check box.

c) Select the column that the text index will follow the order of from the **Order Column** menu.

d) Select the **Ascending** option button to order data in the text index by the Order Column's ascending order.

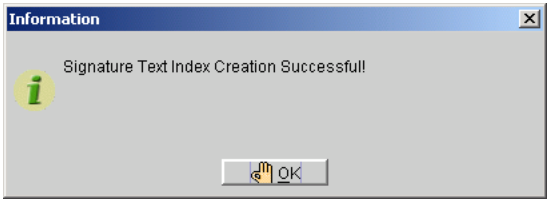
e) Select the **Descending** option button to order data in the text index by the Order Column's descending order.

10. Click **Next**. The **Final review** window will open.



11. To alter the SQL script, enter new script into the **Create Signature Text Index SQL script** field or click **Previous** to display the previous screens in the wizard.

12. To complete the view creation after reviewing the SQL script, click **Finish**. The **Information** dialog box is displayed.



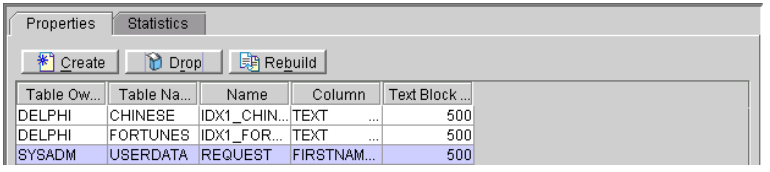
13. Click **OK**.

7.2 Dropping a Signature Text Index

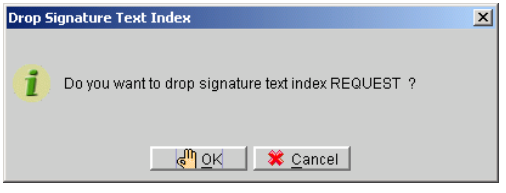
You can drop a text index that is no longer required.

➲ **To drop a signature text index:**

1. Click the **Signature Text Indexes** object in the tree. All the text indexes in the database will be displayed in the **Properties** page.
2. Select the **Signature Text Index** that has to be dropped by clicking on it.



3. Click **Drop**. A confirmation window is displayed to confirm that you want to drop the signature text index.



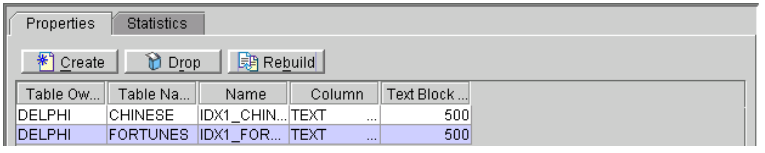
4. Click **OK** and the remaining signature text indexes in the database will be displayed.

7.3 Rebuilding a Signature Text Index

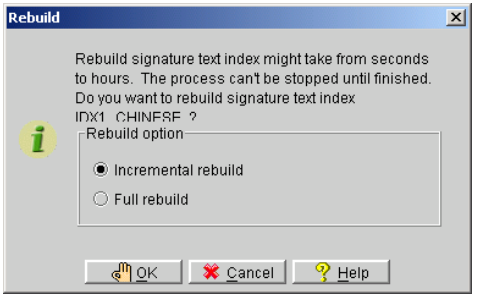
If a number of changes have been made to the table or to the column on which a text index was created, then the text index should be rebuilt. When a text index is rebuilt, it will be updated to include all changes to the table or column.

➲ **To rebuild a signature text index:**

1. Click the **Signature Text Indexes** object in the tree. The **Properties** page is displayed.
2. Select the signature text index that has to be rebuilt.



3. Click **Rebuild**. The **Rebuild** dialog will open.



4. Select an incremental rebuild or full rebuild.
5. Click **OK**. A message saying the signature text index has been rebuilt is displayed.

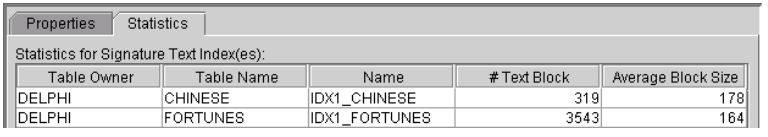
6. Click **OK**. The message clears.

7.4 Checking Signature Text Index Statistics

The statistics of a signature text index may periodically be checked. Statistics include the table owner and name, the text index name, the number of text blocks and the average block size.

➲ **To view text index statistics:**

1. Select **Signature Text Indexes** from the tree. The **Properties** page is displayed.
2. Click the **Statistics** tab. The statistics of the text indexes will be displayed. These include the names of the signature text indexes, the average block size of the indexes and the number of text blocks.



8 Working with IVF Text Indexes

Inverted file (IVF) text indexes are a type of text index built on files separate from the tablespace that the columns are stored in. As a result, query performance is significantly better for larger text indexes. You need to create a logical path for the IVF text index files to be stored in before creating an IVF text index. Logical paths are specified in the JConfiguration Tool or by modifying the DMConfig.ini file. Once the path is specified, DBMaker handles the creation of files for use in the IVF text index.

Since IVF text indexes are a part of the operating system's file system, they may be liable to fragmentation. Be sure to specify a logical path on a reserved partition and periodically defragment the partition to ensure optimal performance.

Once you create an IVF text index on a table, its operation is transparent to users of the database. The DBMS uses the IVF text index to improve full-text query performance whenever possible.

8.1 Creating an IVF Text Index

IVF Text indexes can be built on all character type columns, including CHAR, VARCHAR, LONG VARCHAR, NCHAR, NVARCHAR, NCLOB, and FILE types. A table can have multiple text indexes, but a text index can only be built on a column. If you need to perform full-text search for other columns, you should build an index for each column.

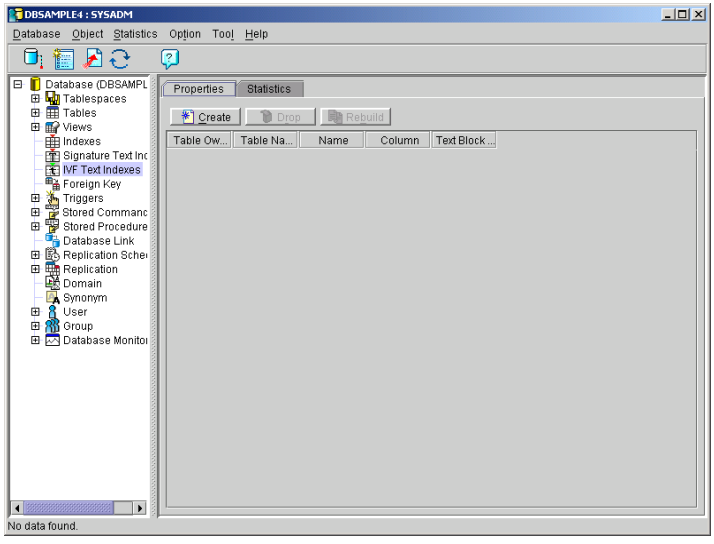
The three parameters used for IVF text indexes include the storage path, the total text size and the scale. The storage path is the location of the IVF text index in the file system. A path must be specified in the configuration file before using an IVF text index. For more information on the storage path for IVF text indexes, refer to the *JConfiguration Tool Reference*.

The total text size is an approximation of the total number of megabytes of storage that are used by the columns on which the text index is to be built. The total text size can range from 10-10,000 MB, and the default total text size value is 32 MB.

IVF text indexes may be sorted by using the **order by** feature.

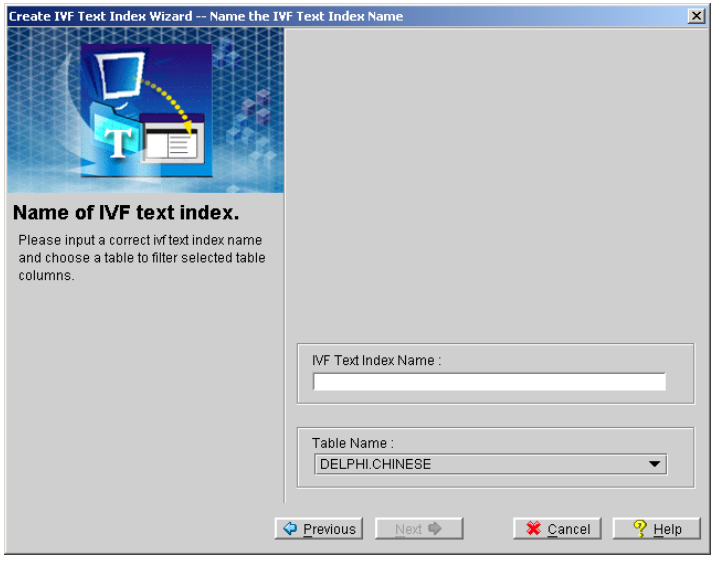
➲ **To create an IVF Text Index:**

1. Click the **IVF Text Index** object in the tree. The **Properties** page appears.



2. Click **Create**. The **Introduction** window for the **Create Text Index Wizard** is displayed.

3. Click **Next**. The **Name the Text Index** window is displayed.

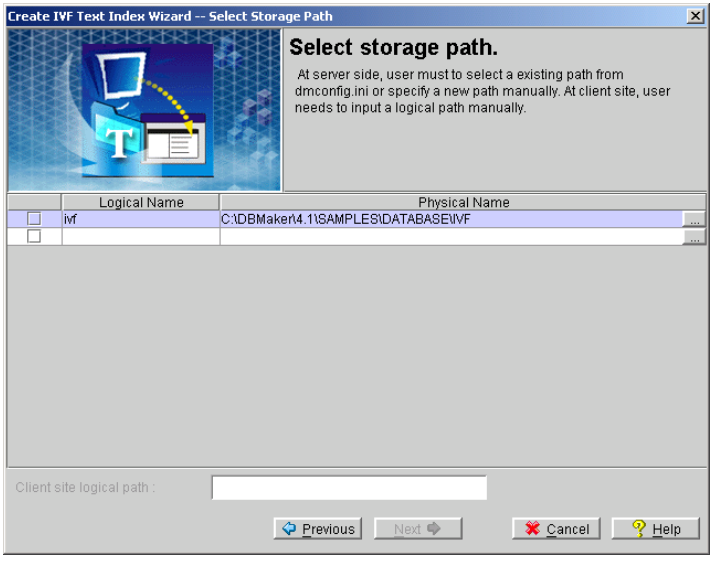


4. Select the name of the table from which the text index is to be created from the **Table Name** menu.

5. Enter the name of the text index in the **IVF Text Index Name** field.

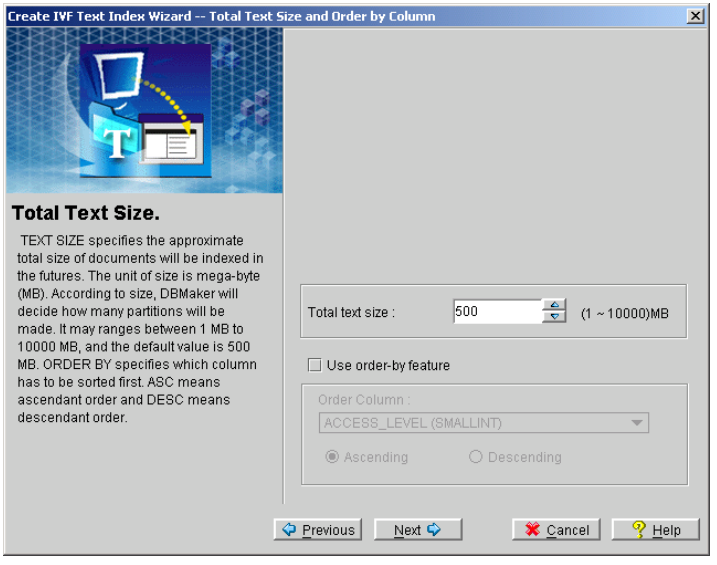
6. A list of available columns on the selected table appears on the left hand list box. Select the columns that you want to use in the IVF text index by double clicking on the column name or selecting the column and clicking the **Add** button. The **Add all** button can be used to add all columns to the index. All the columns selected for the creation of the text index will be displayed in the list box to the right. Columns in the text index can be likewise removed by double clicking on the column name or selecting the column and clicking the **Remove** button.

7. Click **Next**. The **Select Storage Path** window will open



8. Select a storage path for the IVF text index by selecting the check box in the row corresponding to the IVF text index path. The paths and logical names that appear correspond to those specified in the configuration file. If you have not yet specified a path for storing IVF text indexes, open the JConfiguration Tool and add a new **IVF Text Index Storage Path** on the **User Files** page. For detailed instructions on how to use the JConfiguration Tool, refer to the *JConfiguration Tool Reference*.

9. Click **Next**. The **Total Text Size and Order by Column** window will open.



10. To change the IVF text index settings:

To change the total text size, enter a value into **Total Text Size** between 1 and 10,000.

11. To present data as it relates to another column in the table select the **Use order-by feature** check box.

a) Select the column that the text index will follow the order of from the **Order Column** menu.

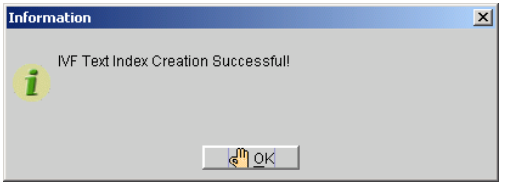
b) Select the **Ascending** option button to order data in the text index by the Order Column's ascending order.

c) Select the **Descending** option button to order data in the text index by the Order Column's descending order.

12. Click **Next**. The **Final review** window will open.

13. To alter the SQL script, enter new script into the **Create Text Index SQL script** field or click **Previous** to display the previous screens in the wizard.

14. To complete the view creation after reviewing the SQL script, click **Finish**. The **Information** dialog box is displayed.



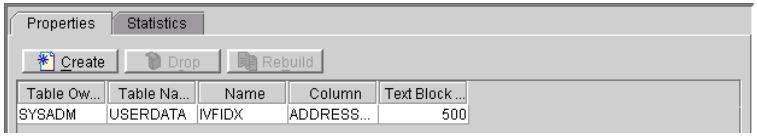
15. Click **OK**.

8.2 Dropping an IVF Text Index

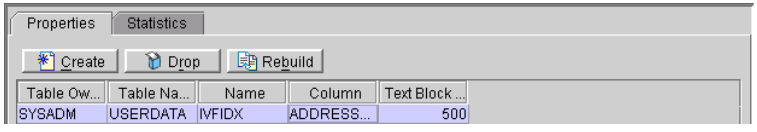
You can drop a text index that is no longer required.

➲ **To drop an IVF text index:**

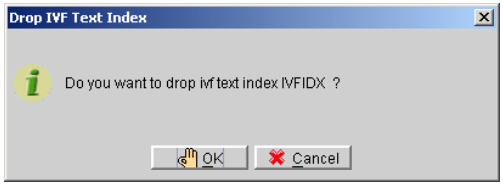
1. Click the **IVF Text Indexes** object in the tree. All the text indexes in the database will be displayed in the **Properties** page.



2. Select the **IVF Text Index** that has to be dropped by clicking on it.



3. Click **Drop**. A confirmation window is displayed to confirm that you want to drop the IVF text index.



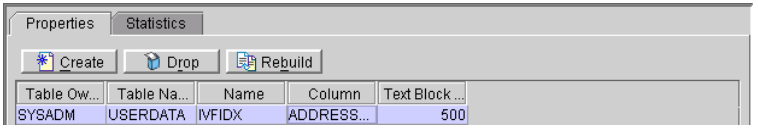
4. Click **OK** and the remaining IVF text indexes in the database will be displayed.

8.3 Rebuilding an IVF Text Index

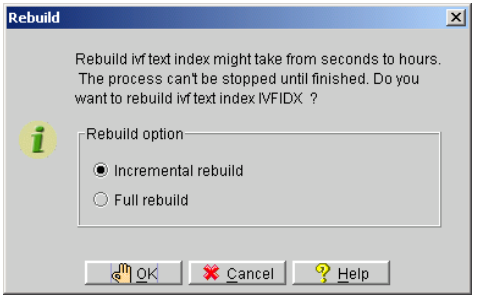
If a number of changes have been made to the table or to the column on which a text index was created, then the text index should be rebuilt. When an IVF text index is rebuilt, it will be updated to include all changes to the table or column.

➲ **To rebuild an IVF text index:**

1. Click the **Text Indexes** object in the tree. The **Properties** page is displayed.
2. Select the IVF text index that has to be rebuilt.



3. Click **Rebuild**. The **Rebuild** dialog will open.



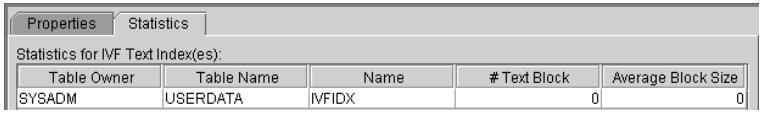
4. Select an incremental rebuild or full rebuild.
5. Click **OK**. A message saying the IVF text index has been rebuilt is displayed.
6. Click **OK**. The message clears.

8.4 Checking IVF Text Index Statistics

The statistics of an IVF text index may periodically be checked. Statistics include the table owner and name, the text index name, the number of text blocks and the average block size.

➲ **To view text index statistics:**

1. Select **IVF Text Indexes** from the tree. The **Properties** page is displayed.
2. Click the **Statistics** tab. The statistics of the text indexes will be displayed. These include the names of the text indexes, the average block size of the IVF text indexes and the number of text blocks.



9 Using Foreign Keys

A foreign key is the relationship that a column or combination of columns in a table has with the primary key or unique index from another table. A foreign key contains the same values and denotes the relationship between the two tables.

In this section, you will create a foreign key for a table. You will also drop a foreign key that is no longer required in the database.

9.1 Creating a Foreign Key

A foreign key is used to refer to another table by specifying the referencing and referenced columns. Both the referencing and referenced columns should be mapped to each other. The mapping columns should be the same type and length. The referenced columns should be NOT NULL, but the referencing columns can be NULL .The referenced columns should be a primary key or unique index.

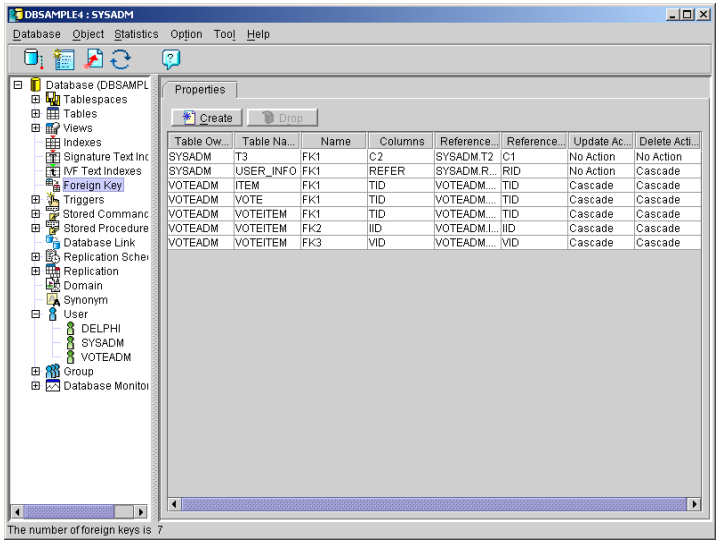
Adding a Foreign Key

Creating a foreign key consists of several phases. The first phase consists of selecting the table that contains the referencing columns, selecting the columns, and naming the foreign key.

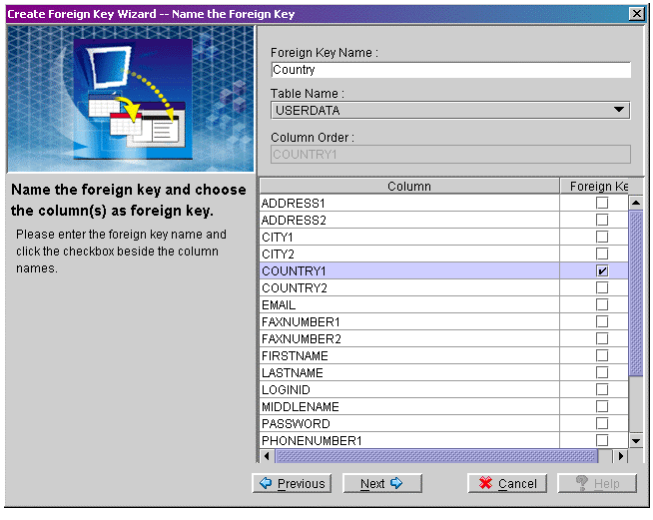
Foreign key columns in the referencing table must map exactly to the primary key columns or unique index columns of the referenced table. The number of columns that are used in the foreign key and the schema of those columns must be the same in both the referencing and referenced tables.

➲ **To add a foreign key:**

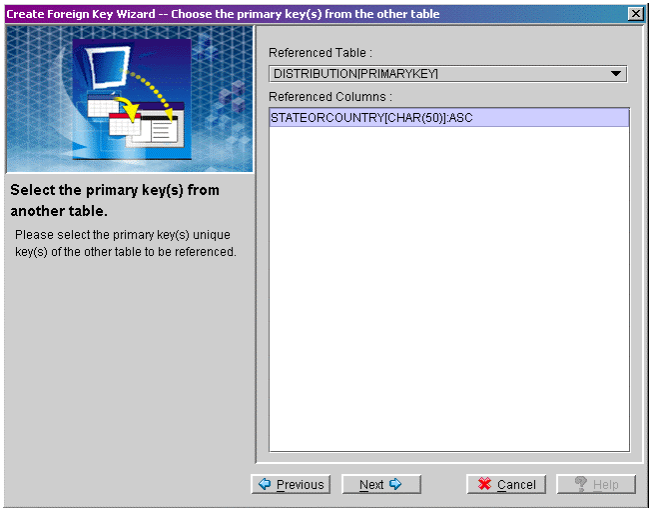
1. Select **Foreign Key** from the tree. The **Properties** page appears.



2. Click **Create**. The **Create Foreign Key – Introduction** window is displayed. Select **Next** at the bottom of the page, the **Name the Foreign Key** window is displayed.



3. Enter the foreign key name in the **Foreign Key Name** text field.
4. Select the referencing table name.
5. Click checkboxes beside the column names to indicate which columns to include in the foreign key.
6. Click **Next**. The **Choose the primary key(s) from the other table** window of the **Create Foreign Key Wizard** is displayed.

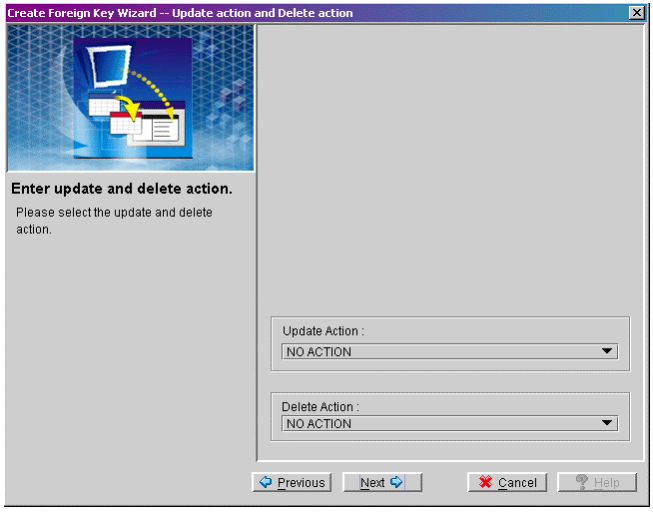


7. Select the name of the referenced table from the **Referenced Table** menu.

NOTE *Only tables containing primary keys or unique indexes that match the schema of the foreign key are shown.*

8. All the selected table columns that are primary keys or unique indexes will be displayed.

9. Click **Next**. The **Update action and Delete action** window will open.



Setting Foreign Key Options

You can choose a foreign key as well as select foreign key options. The following table shows the effect of available foreign key options.

Foreign Key Option	Description
Set Null	Records in the foreign key column are set to Null if the user updates / deletes the corresponding records in the primary key or unique index column
Cascade	The corresponding records in the referencing table are also updated or deleted.
Set Default	The foreign key record is set to a default value if the user updates / deletes corresponding records in the primary key or unique index column.

FOREIGN KEY OPTION	DESCRIPTION
No Action	There is no action on the foreign key column if the user updates / deletes corresponding records in the primary key or unique index column.

Table 9-1Foreign Key Options

➲ **To set foreign key options:**

1. Choose one of the following foreign key settings from the **Update Action** menu:

   No Action

   Cascade

   Set Default

   Set Null

   Upon update of the referenced table the chosen update action will occur.

2. Choose one of the following foreign key settings from the **Delete Action** menu:

   No Action

   Cascade

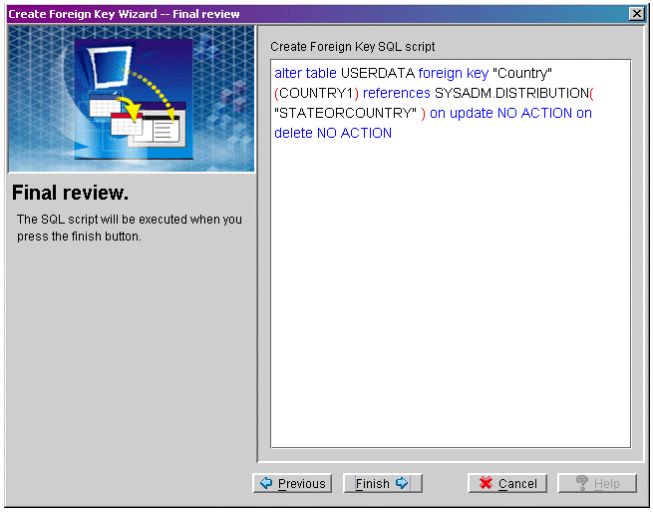
   Set Default

   Set Null

   Upon update of the referenced table the chosen delete action will occur.

NOTE *The default setting is **No Action**.*

3. Click **Next**. A window that allows the final review of all the SQL commands you provided to create the foreign key is displayed as shown below.



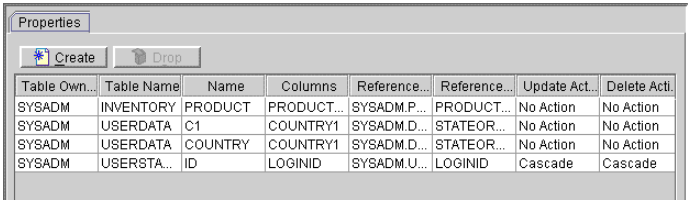
4. Click **Finish**. The **Properties** page will reappear with the new foreign key in the list.

9.2 Dropping a Foreign key

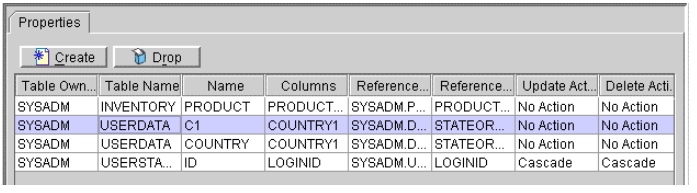
If a foreign key is no longer required, it can be dropped from the database.

➲ **To drop a Foreign key:**

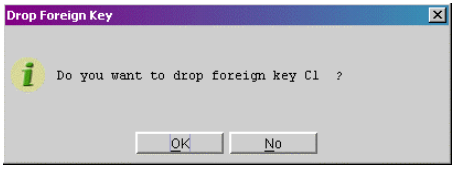
1. Select the **Foreign Key** object from the tree. The **Properties** page is displayed.



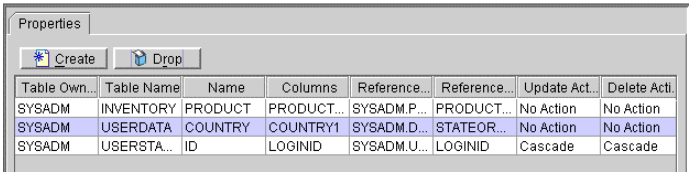
2. Select the foreign key that is to be dropped by clicking on it.



3. Click **Drop**. The **Drop Foreign Key** dialog box appears.



4. Click **OK**. The remaining foreign keys in the database will be displayed.



10 Using Triggers

Triggers are predefined commands that are automatically executed in response to specific events, regardless of which user or application program generated the events. Triggers allow you to customize your database in ways that may not be possible with standard SQL commands. You can use triggers to:

Implement business rules

Create an audit trail of database activities

Derive additional values from existing data

Replicate data across multiple tables

Perform security authorization procedures

Control data integrity

Define unconventional integrity constraints

In this section, you will create a trigger with various trigger options. Once this is done, you will drop a trigger from a database.

10.1 Creating a Trigger

To create a trigger on a table, you must be the owner of the table or have DBA security privileges on the table. You must also have the necessary object privileges on all objects referenced in the trigger definition to successfully create the trigger. Creating a trigger involves the following set of actions:

Assigning a trigger name and table.

Specifying trigger action time.

Indicating the referencing clause.

Entering the WHEN condition clause.

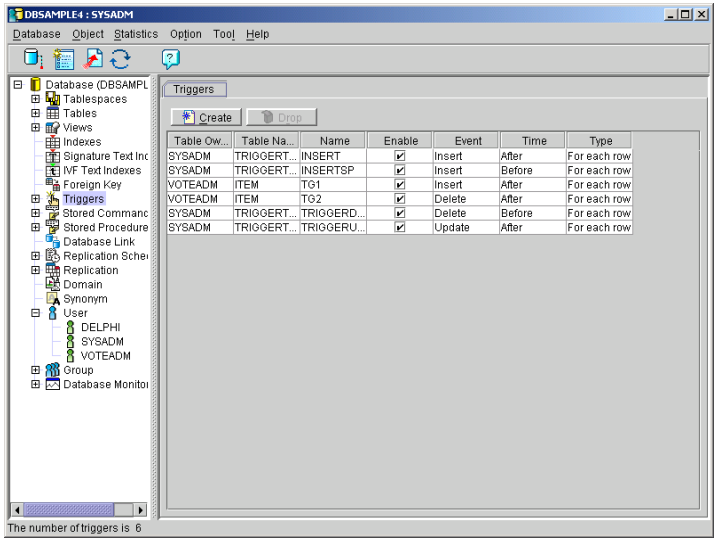
Entering SQL statements for the trigger action

Assigning a Trigger Name and Table

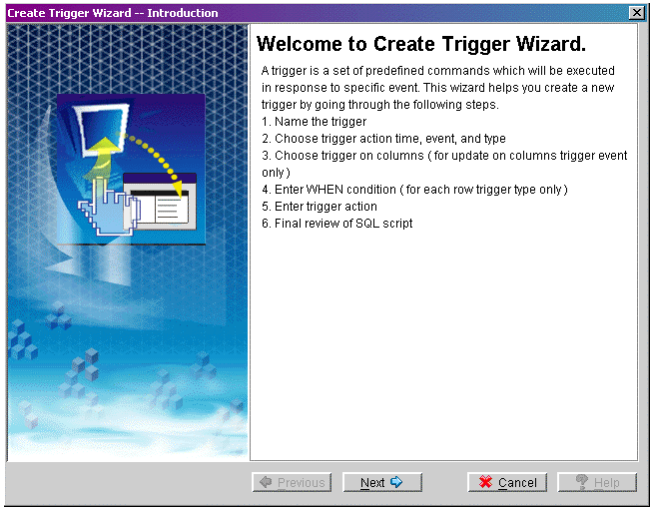
When creating a trigger you need to assign a trigger name and indicate the table in which the trigger is created.

➲ **To specify a trigger name and table:**

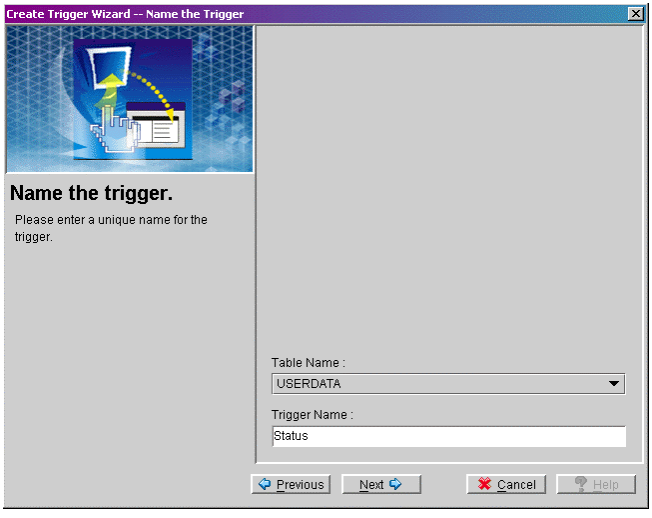
1. Click the **Trigger** object in the tree. The **Triggers** page is displayed.



2. Click **Create**. The **Introduction** window of the **Create Trigger Wizard** is displayed.



3. Click **Next**. The **Name the Trigger** window is displayed.
4. Select the table name in which the Trigger will be based on from the **Table Name** menu.
5. Enter the name of the trigger in the **Trigger Name** field.



6. Click **Next**. The **Choose the Trigger Action Time** window will open.

Specifying Trigger Action Settings

Once you have indicated the trigger name and the table in which the trigger is created, you can then specify the type of actions the trigger will perform. The settings are indicated as follows:

Action Time: The trigger action time specifies whether a trigger should fire before or after the SQL statement that activates the trigger. You can use the BEFORE and AFTER keywords to specify the trigger action time. The BEFORE keyword indicates the trigger action will fire before the trigger statement. The AFTER keyword indicates the trigger action will fire after the trigger statement. You can only specify one trigger time for each trigger.

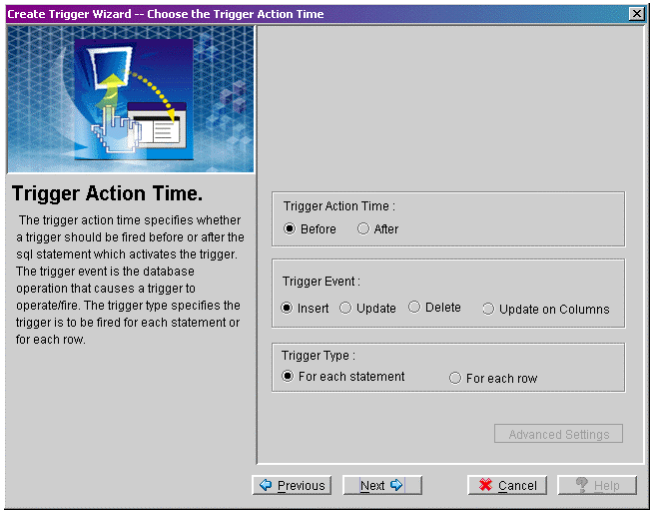
Trigger Event: The trigger event is the database operation that causes a trigger to execute (or fire). The trigger event may be an INSERT, UPDATE, or DELETE statement that operates on the trigger table. You can only specify one trigger event for each trigger.

Trigger Type: The trigger type specifies how many times the trigger will fire for each trigger event. There are two types of triggers: row triggers and statement triggers. The FOR EACH ROW keywords specify a row trigger, which fires a trigger action once for each row modified by the trigger event. The FOR EACH STATEMENT keyword specifies a statement trigger, which fires a trigger action once for each trigger event.

Referencing Clause. If the selected Trigger type is FOR EACH ROW, the advanced setting tab is highlighted. Click this tab to define the REFERENCING clause. The REFERENCING clause defines correlation names for the old and new values of a column. This is primarily used when you cannot use the default OLD and NEW names because of a conflict with a table with the same name.

➲ **To specify trigger action settings:**

1. From the **Name the Trigger** window of the **Create Trigger Wizard**, click **Next**. The **Choose the Trigger Action Time** window will open.



2. Select one of the following trigger action time options:

To set the trigger to fire after the SQL statement, click the **After** option button.

To set the trigger to fire before the SQL statement, click the **Before** option button.

3. Select one of the following trigger event options:

   To choose the INSERT command as the trigger event, choose the **Insert** option button in the **Trigger Event** group.

   To choose the UPDATE command as the trigger event, choose the **Update** option button in the **Trigger Event** group.

   To choose the DELETE command as the trigger event, choose the **Delete** option button in the **Trigger Event** group.

4. Select one of the following trigger type options from the **Trigger Type** field.

   Selecting the **For each row** option button sets the trigger statement to execute on each row modified by the trigger event.

   Selecting the **For each statement** option button sets the trigger statement to execute upon each instance of the trigger event.

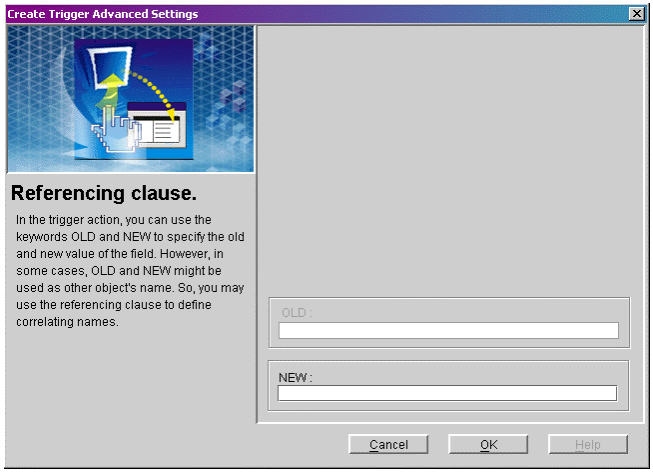
5. If the **For each statement** option button is selected, clicking **Next** will open the **Specify a WHEN condition** clause window. If the **For each row** option button is selected, clicking **Next** will open the **Create Trigger Advanced Settings** window.

Indicating the Referencing Clause

Once you have indicated trigger options, you can then specify the referencing clause of the trigger. You will need to select these options if you selected the FOR EACH ROW as the trigger type. The REFERENCING clause defines correlation names for the old and new values of a column. This is primarily used when you cannot use the default OLD and NEW names because a column has the same name.

➲ **To indicate the referencing clause:**

1. From the **Choose the Trigger Action Time** window of the **Create Trigger Wizard** click **Next**. The **Create Trigger Advanced Settings** window is displayed.



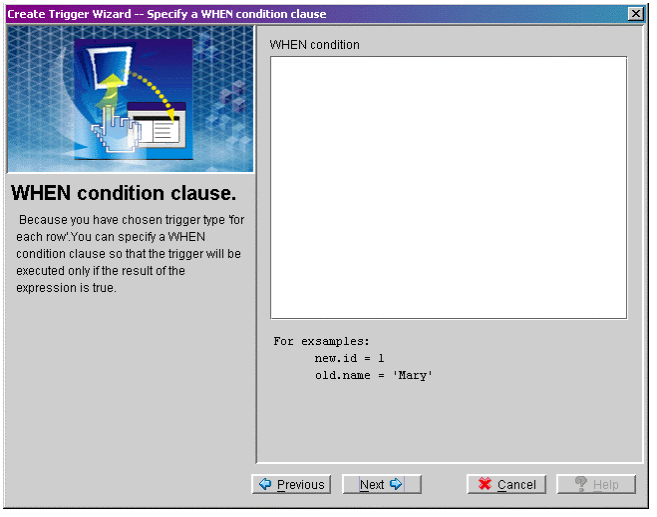
2. When you create a **For each row** type trigger,
3. Enter a substitute name to refer to the old value in the **OLD** field.
4. Enter a substitute name to refer to the new value in the **NEW** field.
5. Click **Next**. The **Specify a WHEN condition clause** window will open.

Entering the WHEN condition clause

Once the settings for the trigger action are complete, you may specify a WHEN clause to place constraints on the actions that will cause the trigger to fire. The WHEN statement must follow proper SQL syntax. See the section on *Constraint Syntax* at the end of Chapter 4 for more information.

➲ **To Specify a WHEN condition:**

1. From the **Choose the Trigger Action Time** window of the Create Trigger Wizard (if "**For each statement**" has been specified) or from the **Create Trigger Advanced Settings** window, click **Next**. The **Specify a WHEN condition clause** window is opened.



2. Enter the WHEN condition and click **Next**. The **Trigger Action** window will open..

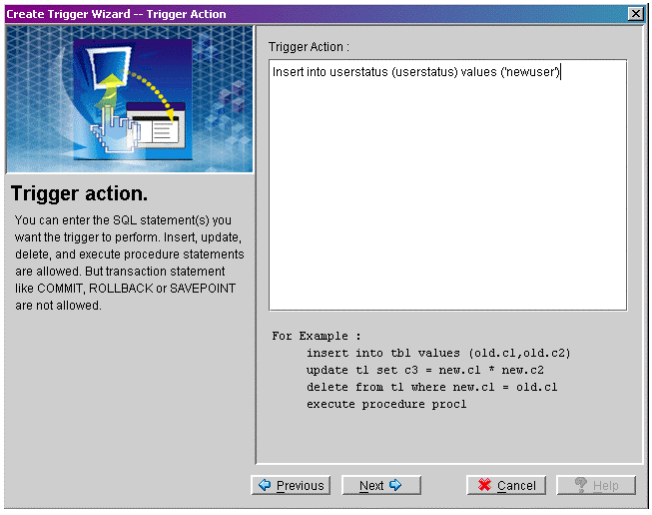
Entering SQL statements for the trigger action

Once the trigger event and its constraints have been defined, the trigger action is set. The trigger action is the command or set of commands that the trigger will carry out on all data that meet the constraints when the trigger is fired.

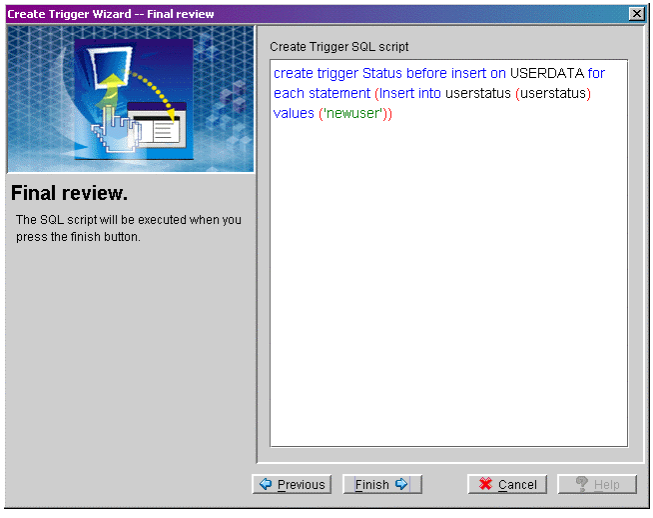
➲ **To Specify SQL statements for the trigger action:**

1. From the **Specify a WHEN condition clause**, click **Next**. The **Trigger Action** window will open.

2. Enter the SQL statements that are to be performed by the trigger.



3. Click **Next**. The **Final review** window will open.



4. Review the final SQL script. Click **Finish** if no more changes are to be made. A message saying the trigger is created successfully is displayed.

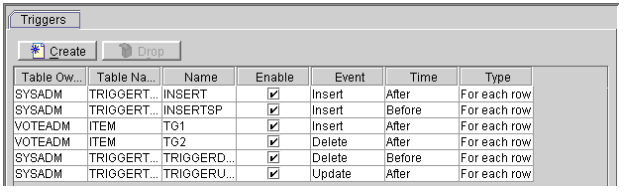
5. Click **OK**.

10.2 Dropping a Trigger

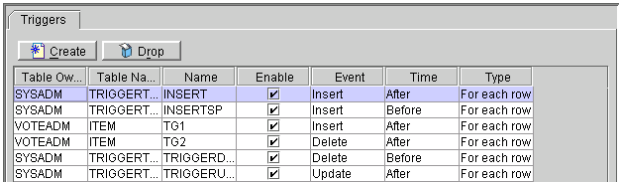
When a table's schema is altered and then a trigger on that table is fired, DBMaker will try to execute the trigger according to the new table definition. However, if the specified column in a triggering event or action is dropped, the trigger execution will fail and the triggering statement will fail too. In this situation, users need to drop the trigger or modify the trigger definition according to the new table schema. Upon deleting a table, the trigger that references that table will also be dropped.

➲ **To delete a trigger:**

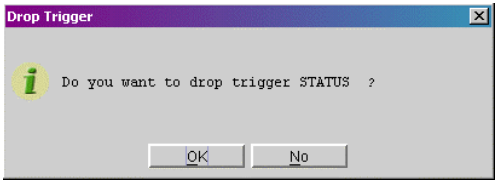
1. Click the object **Trigger** in the tree. All the triggers in the database will be displayed as shown.



2. Select the trigger that is to be dropped by clicking on it.



3. Click **Drop**. A confirmation window will open to confirm if the trigger is to be dropped.



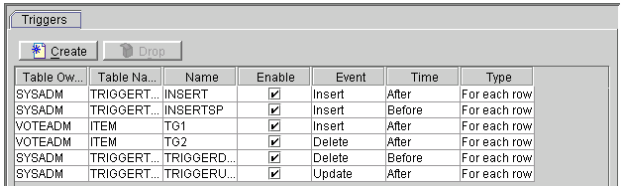
4. Click **OK**.

10.3 Modifying a Trigger

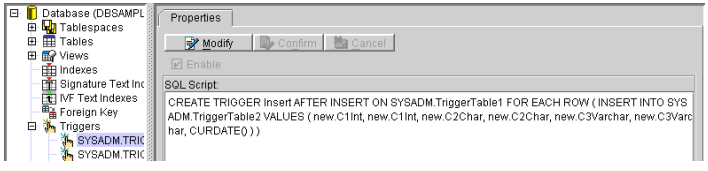
After creating a trigger, it can be modified or disabled. Changing the trigger's SQL code can modify the triggering event, trigger action, or trigger type.

➲ **To modify a trigger:**

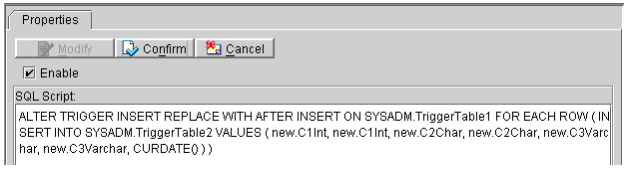
1. Click the object **Trigger** in the tree. All the triggers in the database will be displayed on the **Triggers** page.



2. Double click the trigger to be modified, or expand the Triggers node in the tree and select a trigger from the tree. The **Properties** page will appear.



3. Click the **Modify** button.



4. To disable the trigger, remove the check mark next to **Enable**.
5. To make changes to the SQL script, click in the appropriate place in the **SQL Script** field and edit the statement.
6. Click the **Confirm** button. The modified trigger will be displayed.

11 Working with Stored Commands

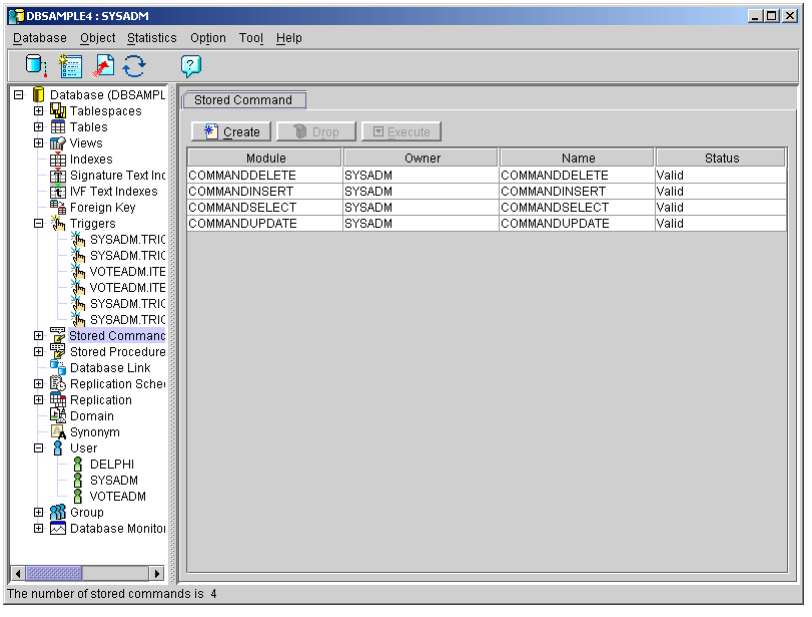
A stored command is a compiled SQL DML statement stored in the database. Since a stored command is precompiled in an executable form, you can execute the same command without repeatedly compiling and optimizing it. It is possible to create a stored command for any frequently used SQL statement.

11.1 Creating a Stored Command

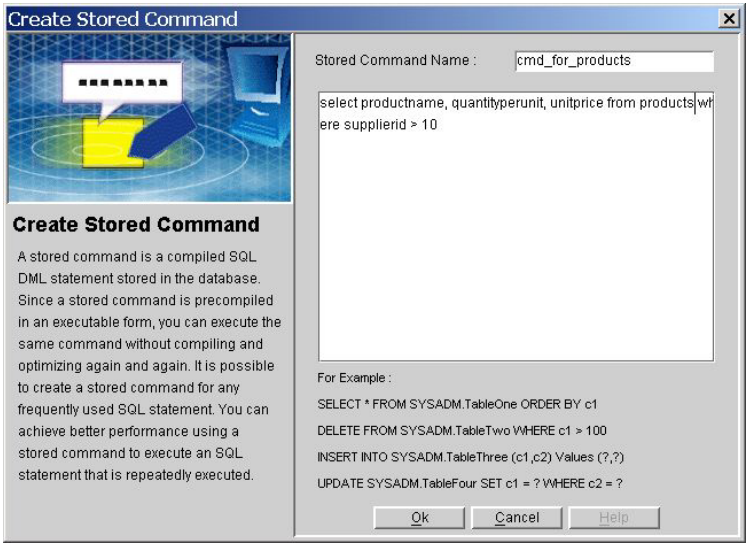
For better performance with frequently used statements, the creation of a stored command is recommended.

➲ **To Create a Stored Command:**

1. Click the object **Stored Command** in the tree. The **Stored Command** window will open.



2. Click **Create**. The **Create Stored Command** window will open.



3. Enter the stored command name into the **Stored Command Name** field.

4. Enter the SQL command.

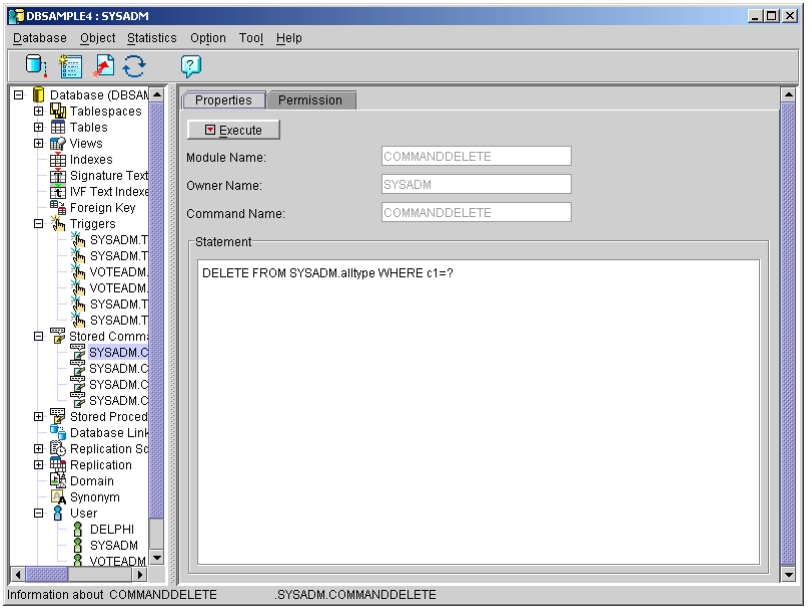
5. Click **OK**. The newly created stored command is displayed with the rest of the stored commands in the database.

11.2 Executing a Stored Command

After creating a stored command, you can execute it directly or in an application program. If you execute a stored command that has input parameters, JDBA tool will prompt you to provide its value when executing the stored command. When you execute a stored command, number of input parameters should be equal to number of input parameters in the stored command.

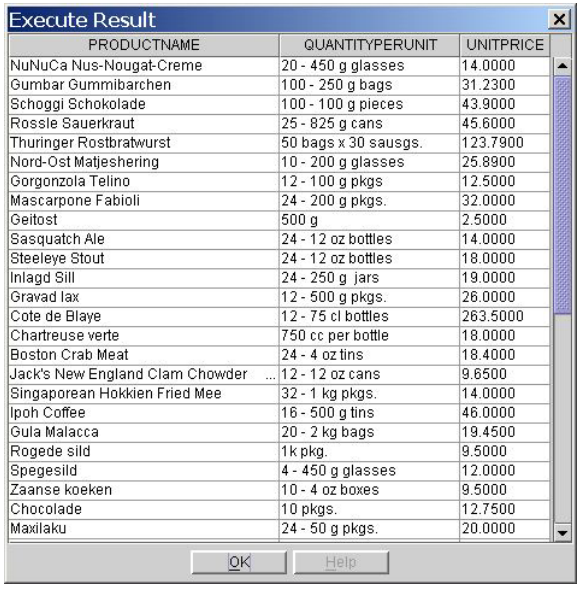
➲ **To Execute a Stored Command:**

1. Open the **Stored Command** node and then select the stored command that is to be executed. The stored command **Properties** page will open.



NOTE *Double clicking on the Stored Command node of the tree also displays the* ***Properties*** *window.*

2. Click **Execute**. The result of the executed stored command will be displayed.



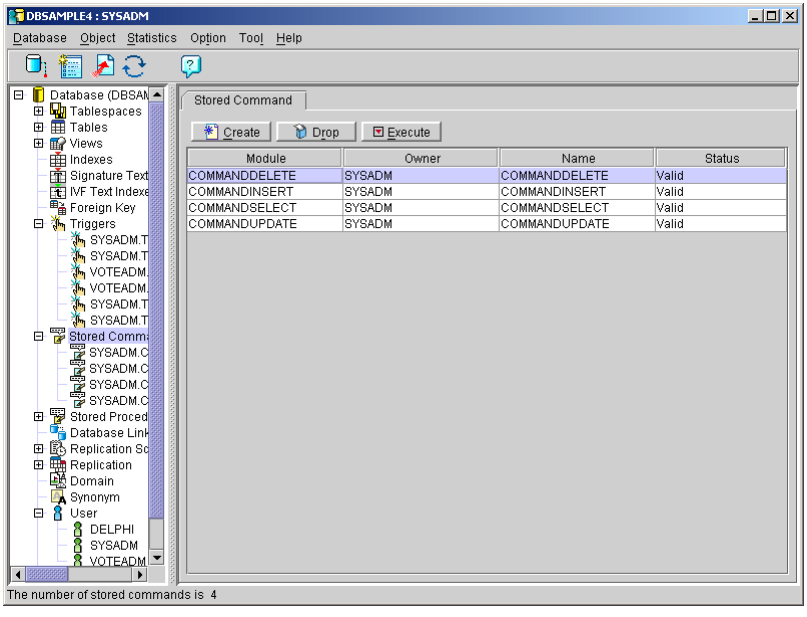
3. Click **OK**. This clears the result window.

11.3 Dropping a Stored Command

If a stored command is no longer required, it can be dropped.

➲ **To drop a stored command:**

1. Click the object **Stored Command** in the tree. The **Stored Command** page will open.
2. Select the stored command that is to be dropped by clicking on it. It will highlight blue.



3. Click **Drop**. The **Drop Stored Command** dialog box will open to confirm if the stored command is to be dropped.

4. Click **OK**. The remaining stored commands in the database will be displayed.

11.4 Granting Stored Command Privileges

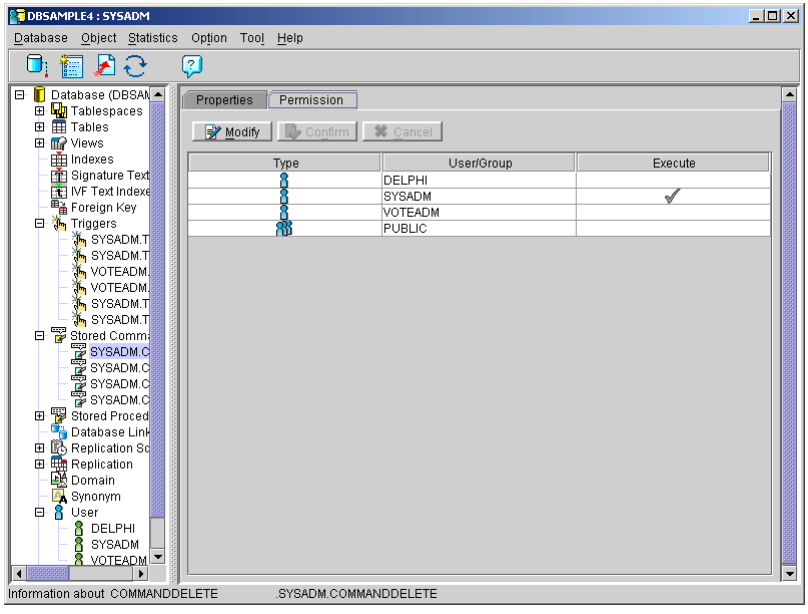
By default only the owner of a stored command or a user with DBA authority or higher have the execute privilege for the stored command. The owner or a user with DBA authority or higher may grant execute privilege to allow the stored command to be called by other users. The execute privilege can also be revoked from other users with resource authority by the owner of the stored command. A user with authority or higher has all privileges on all stored commands and can revoke those privileges even from the owner of the stored command (unless the owner has DBA authority or higher).

➲ **To grant or revoke privileges on a Stored Command:**

1. Open the **Stored Command** node and then select the stored command that is to be executed. The stored command **Properties** page will open.

   NOTE *Double clicking on the stored command node of the tree also displays the **Properties** window.*

2. Click the **Permission** tab. The stored command **Permission** page will open.



3. Select the User/Group to whom you want to grant or revoke the privilege.

4. Click **Modify**.

5. Click the **Execute** column beside the **User/Group** to grant or revoke the execute privilege. A check mark indicates that the user/group has the execute privilege.

6. Click **Confirm**.

12 Working with Stored Procedures

A stored procedure is a special kind of user-defined function that contains embedded SQL statements. This allows you to bypass repeated SQL compilation and optimization, increasing the performance of frequently repeated tasks. You can execute a stored procedure as a command in interactive SQL, or invoke it in application programs, trigger actions, or other stored procedures.

You can accomplish a wide range of objectives with stored procedures, including improving database performance, simplifying the writing of applications, and limiting or monitoring access to a database.

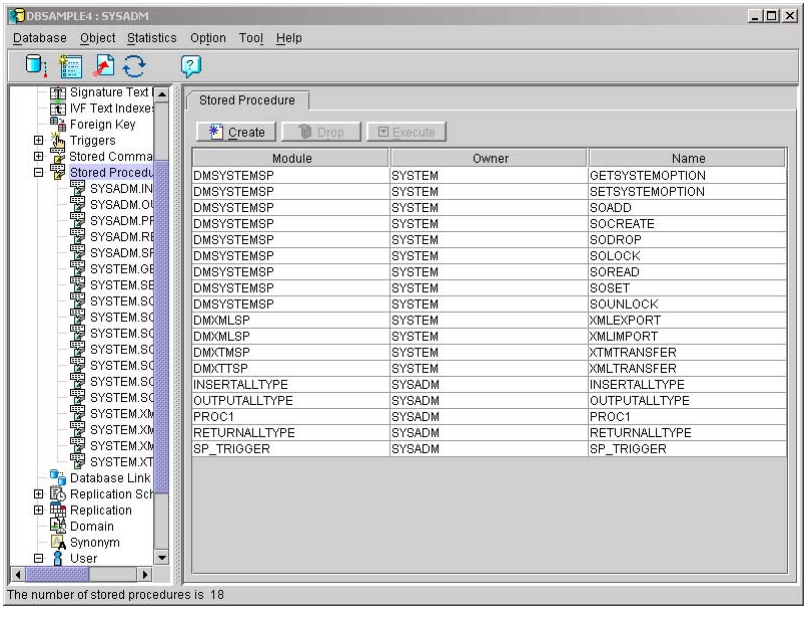
Because a stored procedure is stored in an executable object in the database, it is available to every application running on the database. Several applications can use the same stored procedure and hence the development time for an application is reduced.

12.1 Creating a Stored Procedure

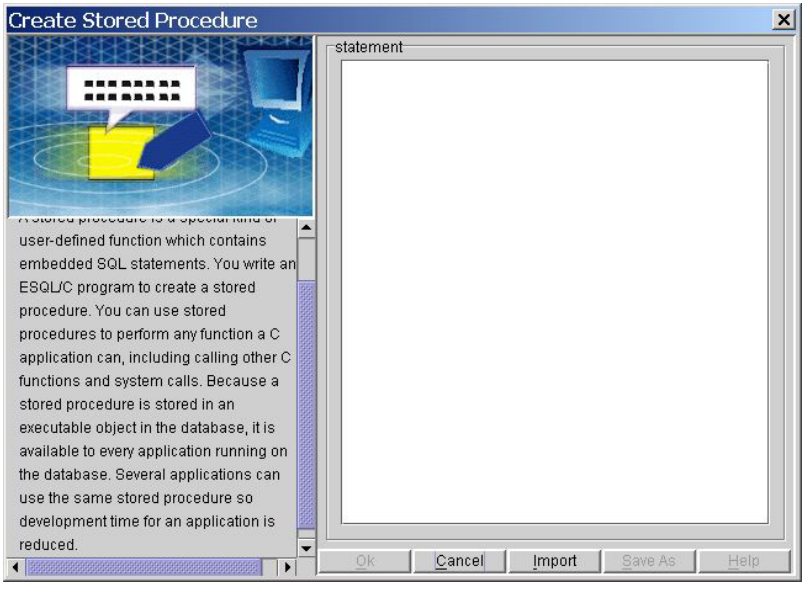
An ESQL/C program has to be written to create a stored procedure. You can use stored procedures to perform any function a C application can, including calling other C functions and system calls. A C compiler is needed for writing stored procedures. If you want to write your stored procedure under UNIX, generally a C compiler is already available and all you need to do is make sure it is included in your path. If your OS platform is Windows, you need to install Visual C++ version 4.2 or greater.

➲ **To Create a Stored Procedure:**

1. Click the object **Stored Procedure** in the tree. The **Stored Procedures** page will appear.

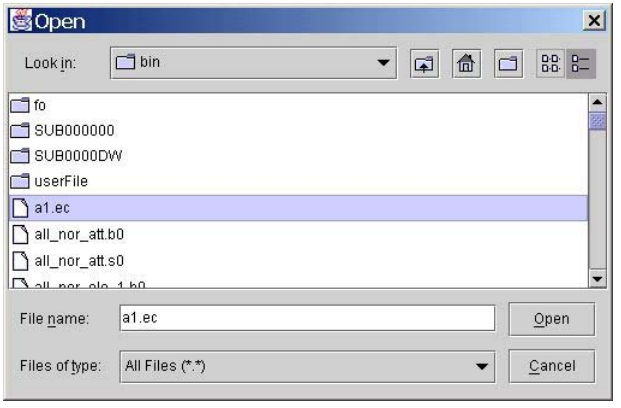


2. Click **Create**. The **Introduction** window of the **Create Stored Procedure** wizard will open.



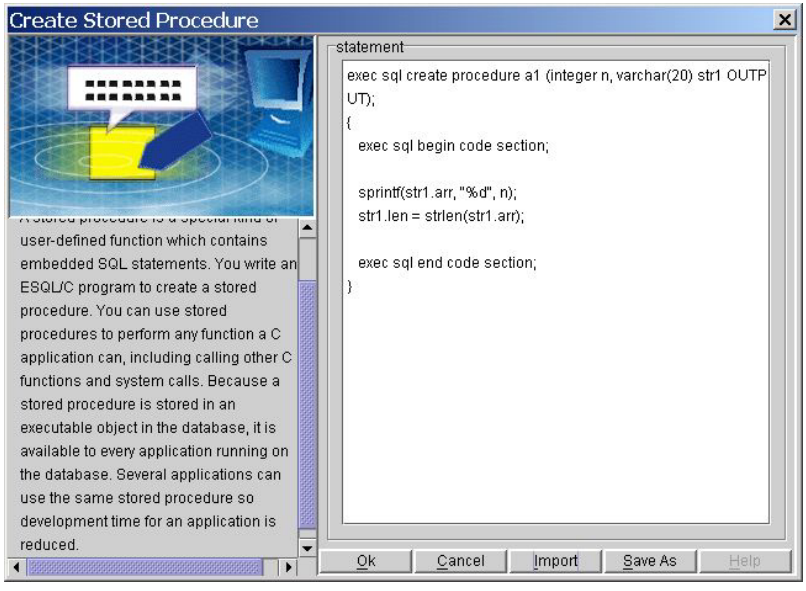
3. You can choose to type a stored procedure into the statement field, or import a stored procedure by clicking **Import**.

4. Clicking **Import** will open the **Open** window. Files can be imported from any source, including the SPDIR directory of other databases on the server or network drives. Select the desired file by typing in the path in the **File name** field, or browse through the directory tree until the correct path is found.



5. Click **Open** to open the file.

6. The **Create Stored Procedure** window will reappear as in the example below if the imported file contains properly formatted (ASCII) text, or if you choose to manually enter the code. Click **Save As** to store the stored procedure to another location, or click **OK** to compile and store the stored procedure in the database.



7. If the Stored Procedure compiles correctly, the following message will appear.

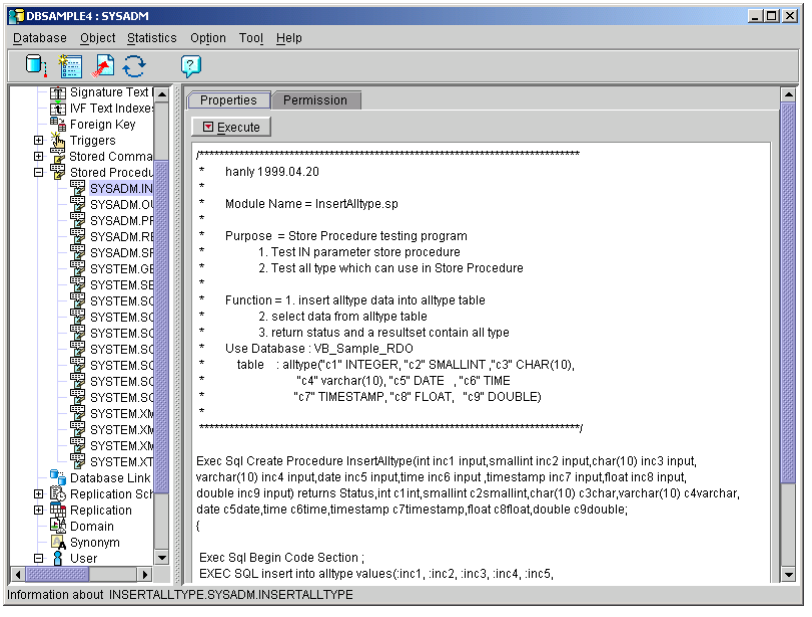
8. Click **OK**.

12.2 Executing a Stored Procedure

After creating a stored procedure, you can execute it directly or in an application program. If you execute a stored procedure that has input parameters, JDBA tool will prompt you to provide the parameter's value when executing the stored procedure. When you execute a stored procedure, number of input parameters should be equal to number of input parameters in the stored procedure.

➲ **To execute a stored procedure:**

1. Open the **Stored Procedure** node and then select the stored procedure that is to be executed. The stored procedure **Properties** page is opened.



NOTE *Double clicking a Stored Procedure on the right panel will display the same window.*

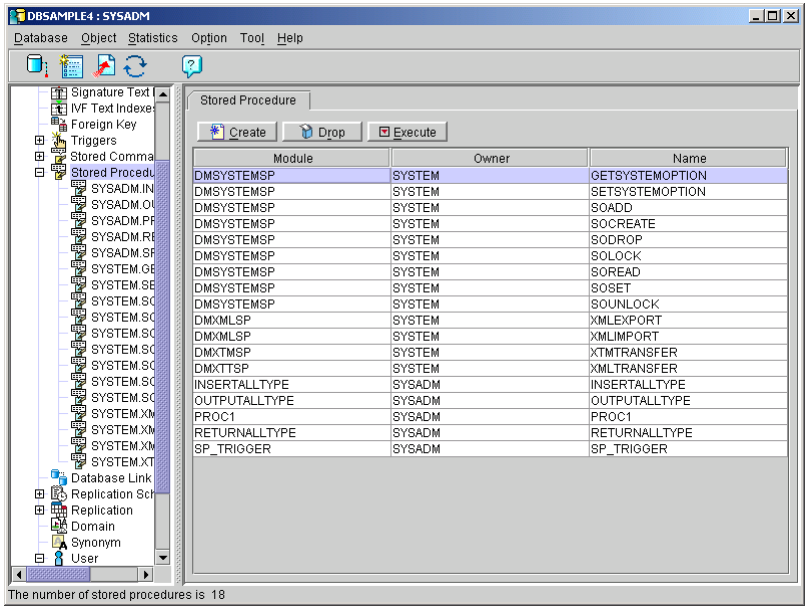
2. Click **Execute**. The result of the executed stored procedure is displayed.
3. Click **OK**. This clears the result window.

12.3 Dropping a Stored Procedure

If a stored procedure is no longer required, it can be dropped.

➲ **To drop a stored procedure:**

1. Click the **Stored Procedure** object in the tree. The stored procedure page with all the stored procedures in the database is displayed.



2. Select the Stored Procedure that is to be dropped.
3. Click **Drop**. A **Drop Stored Procedure** dialog box is displayed to confirm if the stored procedure is to be dropped.

4. Click **OK**.

12.4 Granting Privileges on Stored Procedures

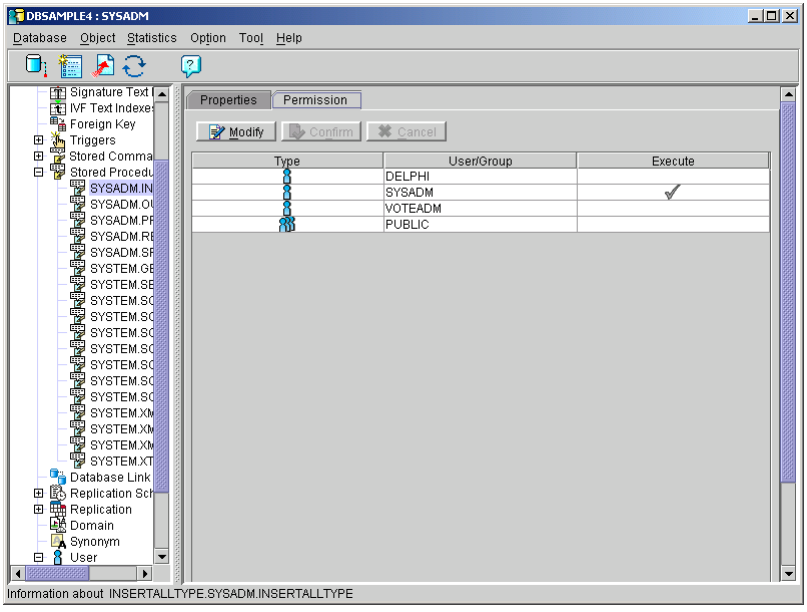
You must have the execute privilege for a stored procedure to execute it. In order to allow a stored procedure to be used by others, you can grant execute privilege on a stored procedure to other database users. If you do not wish others to execute a stored procedure, you can revoke execute privilege of the stored procedure from them. However, you can only grant or revoke the execute privilege of stored procedures. You have the execute privilege on all stored procedures in database if you are a DBA.

➲ **To grant or revoke privileges on a stored procedure:**

1. Open the **Stored Procedure** node and then select the stored procedure that is to be executed. The stored procedure **Properties** page is opened.

   NOTE *Double clicking on the Stored Procedure node from the tree also displays the **Properties** window.*

2. Click the **Permission** tab. The **Permission** page is displayed.



3. Select the User/Group to whom you want to grant or revoke the privilege.

4. Click **Modify**.

5. Click the **Execute** column beside the **User/Group** to grant the execute privilege. A check mark indicates that the user has the execute privilege. By clicking on it again, you can revoke the privilege.

6. Click **Confirm**.

13 Working with Database Links

DBMaker effectively links multiple client applications and multiple database servers. Each client has a direct connection to a single database server, which is known as the Coordinator Database to that client. Through the Coordinator Database, the client can connect to other remote databases, which are also known as Participant Databases. You can access a remote database by using database links defined in the Coordinator Database.

A database link creates a connection to a remote database, and contains the login information and password necessary for connecting to the remote database. This allows you to connect to a remote database with a different user name than you are using in the Coordinator Database, or to connect to a remote database where you do not have an account but can still connect using a public link. It also makes data in a distributed database environment location transparent. The link definition, which also contains the login information and password, is stored in the Coordinator Database.

Database links are also useful for the management of synchronous table replication, and can be used to define the connection between source and destination tables.

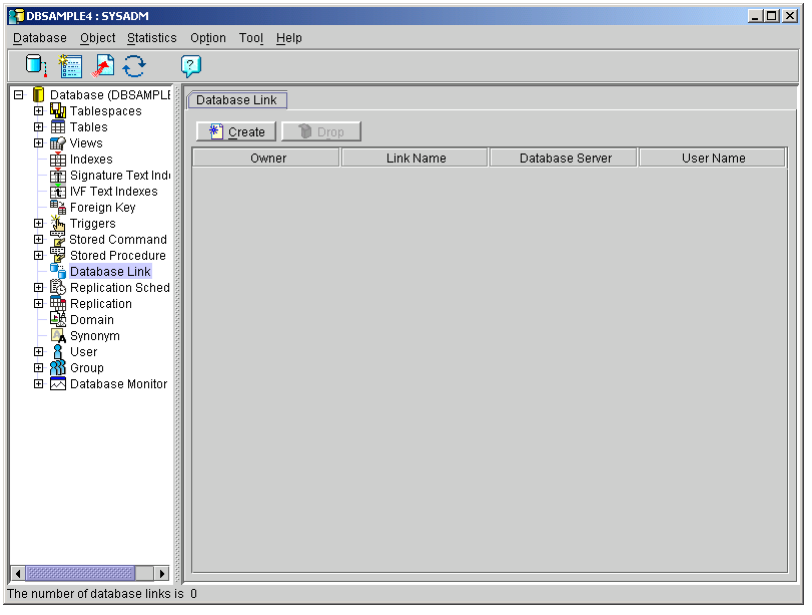
13.1 Creating a Database Link

Only users with DBA authority or higher can create public database links for all users of that database. Other users can create private database links for themselves. Multiple users may create private database links using the same name. If a private database link is created with a name identical to a public database link, the private database link will override the public database link.

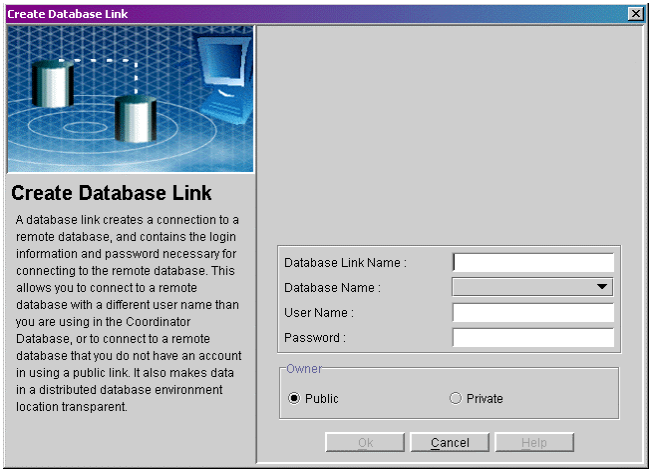
DBMaker will create a public link by default if you do not specify the type of link to create. If you do not specify the User Name and password, your current login name and password will be used by default.

➲ **To create a database Link:**

1. Select the object **Database Link** in the tree. A list of all database links in the database is displayed.



2. Click **Create**. The **Create Database Link** window is displayed.



3. Enter the database link name.

4. Select the database to be the coordinator database from the **Database Name** menu.

5. The default user name for the selected coordinator database is displayed in the **User Name** field. A new name can be entered.

6. Enter the password.

7. If the owner of the database link is private (exclusively for a single user), select the **Private** option button. If the database link is to be used by multiple users, select the **Public** option button. The default setting is **Public**.

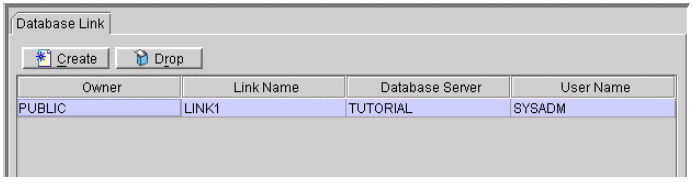
8. Click **OK**. The newly created database link can be seen in the list of all the database links.

13.2 Dropping a Database Link

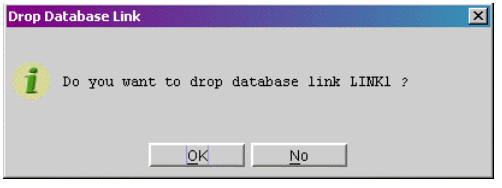
If the database link is no longer required, drop it.

➲ **To drop a database link:**

1. Select the object **Database Link** in the tree. A list of all database links in the database is displayed.
2. Select the database link that is to be dropped.



3. Click **Drop**. The **Drop Database Link** dialog box is displayed.



4. Click **OK**.

14 Working with Replication Schedule

Replication schedules are user-defined schedules that tell the replication daemon when to begin the process of updating replicated data to destination (subscriber) databases. Replication that is carried out by this process is called asynchronous table replication. 'Asynchronous' means the modification to the remote site is delayed. The time difference from source to destination database depends on the replication schedule. Each destination database must have its own replication schedule. Modifications to local tables are stored in replication logs, and are replicated to the destination table(s) according to the replication schedule. Using replication logs enables DBMaker to treat the local transaction and the remote transaction independently, allowing you to update local tables normally even if the remote connection is not available. This allows asynchronous table replications to tolerate network and remote database failures, since DBMaker will keep trying to replicate until any failures are corrected or until the user specified number of attempts to reconnect is reached.

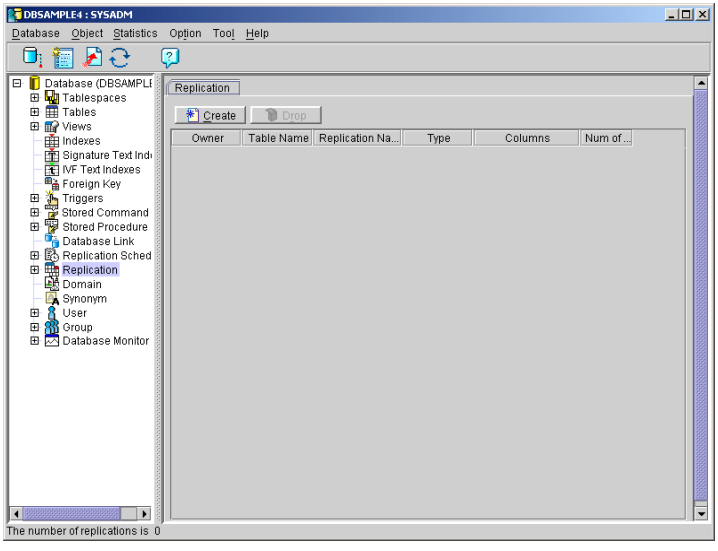
Asynchronous table replication depends on the establishment of a replication schedule. Likewise, developing a replication schedule without an asynchronous table replication that uses it is meaningless. A replication schedule should first be developed. Afterward, the asynchronous table replication can be successfully created.

14.1 Creating a Replication Schedule

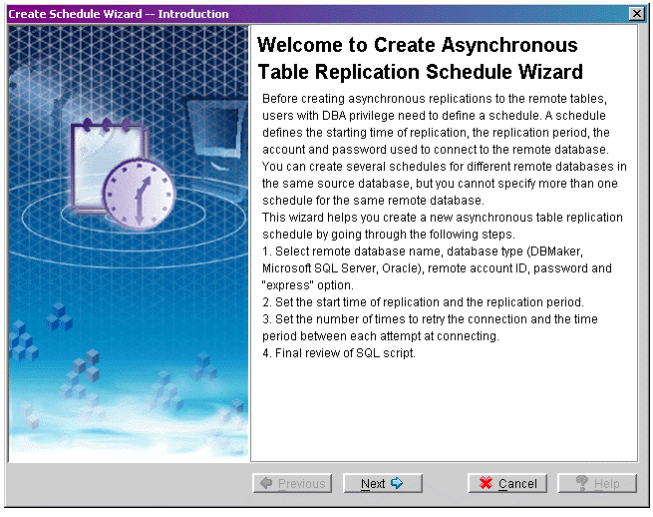
Before creating asynchronous replications to remote tables, a user with DBA privilege needs to define a *schedule*. A schedule defines the starting time of replication, the replicating period, and the account and password used to connect to the remote database. Several schedules for different remote databases in the same source database may be created, but no more than one schedule can be made for the same remote database.

➲ **To create a Replication Schedule:**

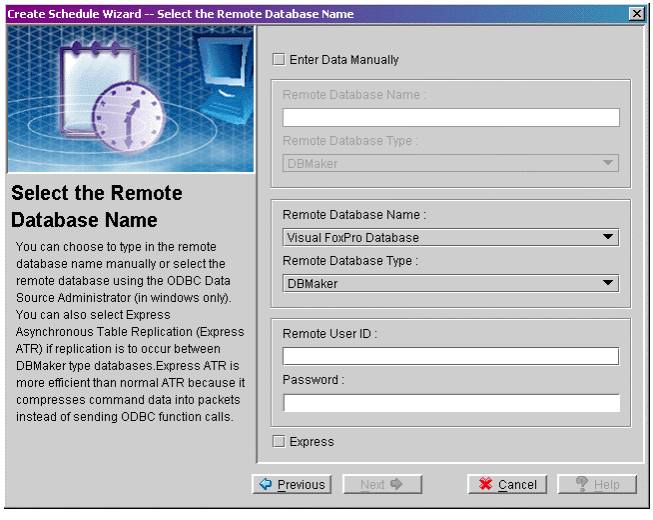
1. Select the object **Replication Schedule** from the tree. The following window is displayed.



2. Click **Create**. The **Create Schedule Wizard – Introduction** window is opened.



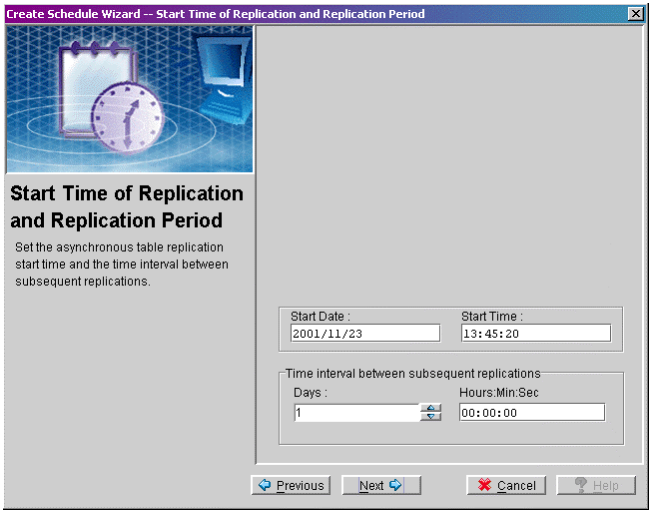
3. Click **Next**. The **Select the Remote Database Name** window is opened.



4. If you choose to enter the remote database name manually, click the **Enter Data Manually** check box and enter the **Remote Database Name**.

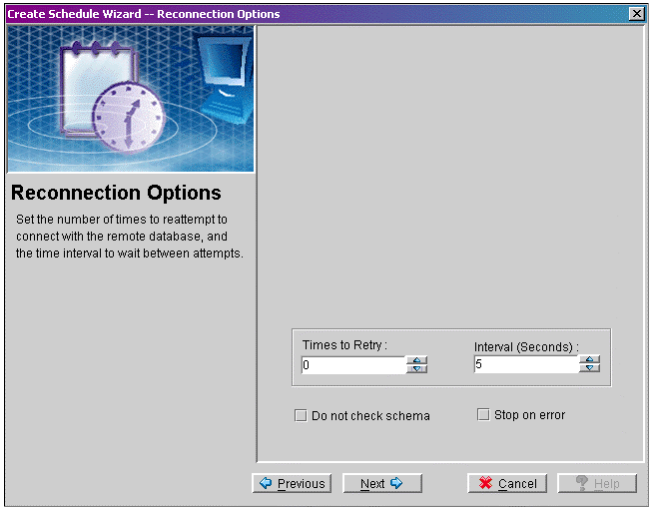
5. Select the **Remote Database Type** from the menu.

6. If you do not choose to enter data manually, you can select the **Remote Database Name** and the **Remote Database Type** from the respective menus.
7. Enter the **User ID** for the remote database
8. Enter the password
9. Click the **Express** check box if you want to enable express asynchronous table replication
10. Click **Next**. The **Start Time of Replication and Replication Period** window will open.



11. Enter the date on which the Asynchronous Table Replication is to be started in the **Start Date** box.
12. Enter the starting time of Asynchronous Table Replication in the **Start Time** box.
13. In the **Time interval between subsequent replications** field:
    a) Enter the number of days between which the Asynchronous Table Replication will be done in the **Days** combo box.
    b) Enter the interval of time in which the Asynchronous Table Replication will be done by entering the time interval in hours, minutes and seconds in the **Hours:Min:Sec** combo box. The total time between replications is equal to the sum of the **Days** and **Hours:Min:Sec** fields.

14. Click **Next**. The **Reconnection Options** window is displayed.



15. Enter the number of times the replication process has to be retried if any failure occurs and replication has not been completed in the **Times to Retry** combo box. Using the upward and downward arrows, you can increase or decrease the number.

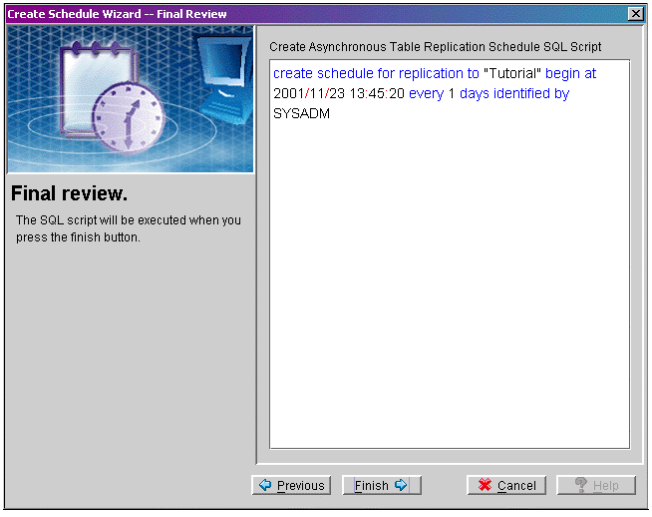
16. Enter the time duration the replication process has to wait before trying again in the **Interval** box. Using the upward and downward arrows, you can increase or decrease the number.

17. Click the **Do not check schema** check box if you do not want the Replication process to check the table schema before replicating.

18. Click the **Stop on error** check box if you want the Replication process to stop if any error is encountered.

NOTE *The Stop on error check box will be disabled if you want an Express Asynchronous Table Replication.*

19. Click **Next**. The following window is displayed.



20. The SQL script for replication schedule creation is displayed. You can modify it if required.

21. Click **Finish**. The **Information** dialog box is displayed as follows.

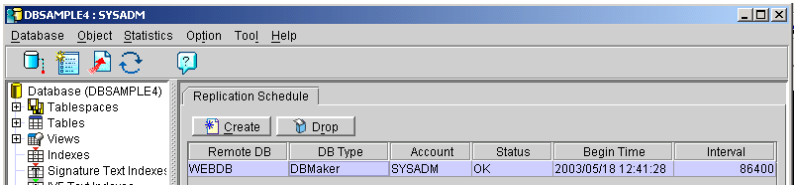
22. Click **OK** and the message will disappear.

14.2 Dropping a Replication Schedule

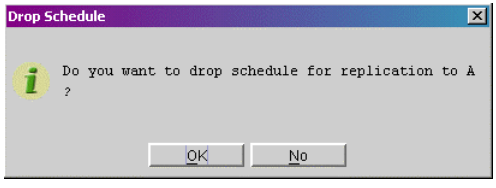
If you do not want to use a replication schedule any more it can be dropped.

➲ **To drop a replication schedule:**

1. Select the object **Replication Schedule** from the tree. The list of all the replication schedules in the database is displayed.
2. Select the replication schedule that is to be dropped. It will highlight blue.



3. Click **Drop**. A confirmation window will open.



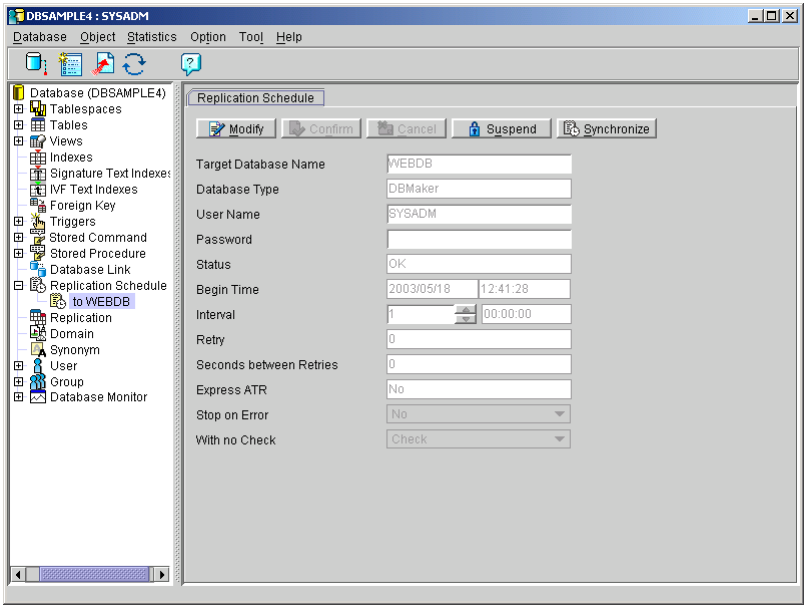
4. Click **OK**.

14.3 Modifying a Replication Schedule

If a replication schedule has to be changed, it can be modified by the following procedure.

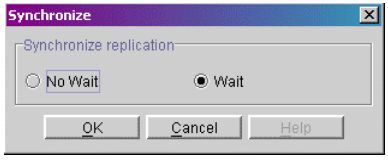
➲ **To modify a replication schedule:**

1. Open the node **Replication Schedule** in the tree and select the replication schedule that is to be modified. The following screen is displayed.



2. Click the **Modify** button. All the replication schedule information is displayed. The **Target Database** name and **Database Type** are disabled and they cannot be modified.
3. The **User Name** can be changed.
4. By clicking **Suspend**, you can change the status of the replication schedule to suspend. Once the status of the replication schedule is changed to suspend, you can click **Resume** to change the status to **OK**.

5. The **Begin Time** of the replication schedule can be modified.
6. The interval of time in which replication has to be done, the number of days and time interval can be modified.
7. The number of times to **Retry** and the interval of time between 2 consecutive retries can also be modified.
8. Replication is to be **Stopped on Error** can be changed to **Yes** or **No**.
9. The **With no Check** button is disabled; whether or not schema is to be checked before replication cannot be changed.
10. Click **Confirm**.
11. Click the **Synchronous** button to synchronize replication. A window is opened as shown.



12. Click the **Wait** option button or click the **No Wait** option.

    Wait – the user must wait for the completion of synchronization before he/she can do other operation.

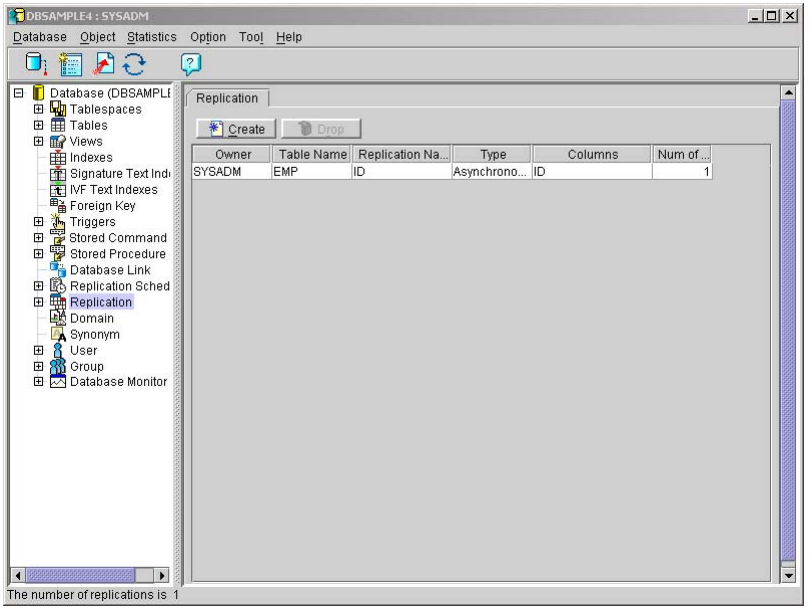
    No Wait – after triggering the synchronization process, the user can continue other processes instead of waiting for the completion of the synchronization process.
13. Click **OK**.

15 Managing Table Replication

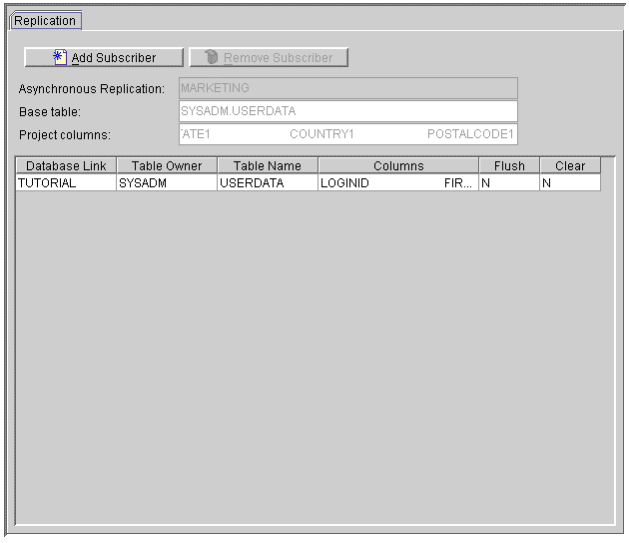
DBMaker allows data to be shared between databases by means of table replication. Table replication can take place simultaneously (synchronous table replication), or by means of the distributor daemon, which periodically updates data on the destination (target) database (Asynchronous Table Replication or ATR). JDBA Tool provides easy to use tools for creation and management of synchronous and asynchronous table replication.

➲ **To View Table Replications:**

1. Select **Replication** from the tree, the **Replication** page will display. You can view a list of replications by owner, source table name, replication name, replication type (synchronous or asynchronous), source columns, and the number of subscribers to the replication (the number of destination tables).



2. Selecting a replication from the tree or the list on the **Replication** page will open the page for that replication. You can view the link names, owners, table names, column names, and settings for each link to a destination table.



15.1 Synchronous Table Replication

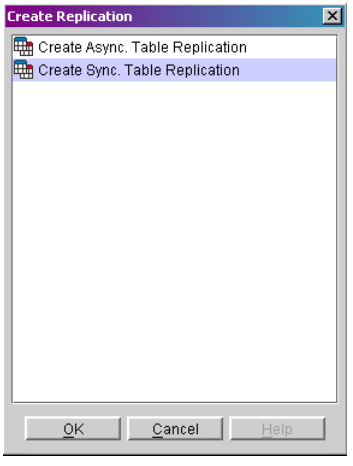
Table replication is used to ensure consistency of data across multiple databases. Synchronous means the modification to the remote site is immediate. The replication modifies the remote table at the same time it modifies the local table. Thus, after establishing a replication, any update on the source table will become a DDB (distributed database) action. This affects the local database's behavior; if the remote database server is cannot be contacted, an update on local database will fail.

Adding Synchronous Table Replication

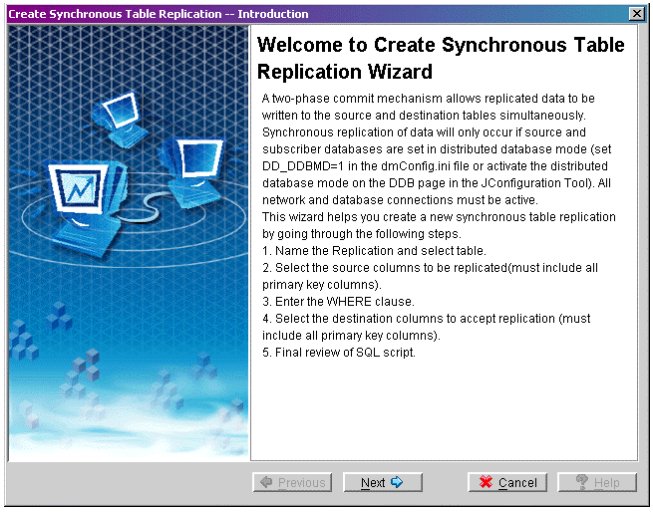
Synchronous table replication must be created for it to occur. Furthermore, distributed database mode must be enabled in all databases engaged in synchronous table replication. The schema of the destination columns must match the schema of the source columns. Primary key columns in the source table must be replicated.

➲ **To create a Synchronous Table Replication:**

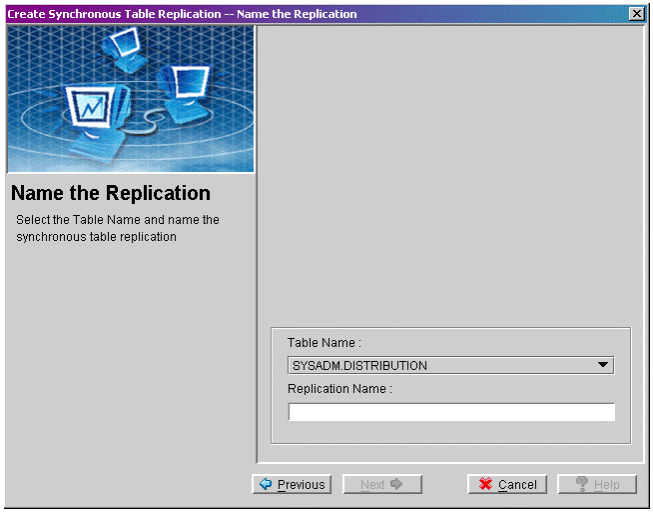
1. Select **Create** from the top of the **Replication** page. The **Create Replication** window will appear.



2. Select **Synchronous Table Replication** from the menu, the **Synchronous Table Replication Wizard** will begin.

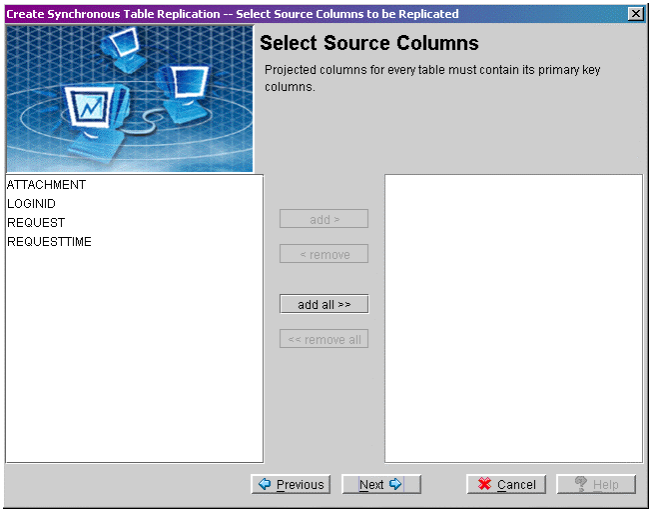


3. Click **Next**. The **Synchronous Table Replication Wizard** will appear.



4. Identify the source table to replicate and name the replication. Select the source table from the **Table Name** menu at the bottom of the window. Enter a name into the **Replication Name** field

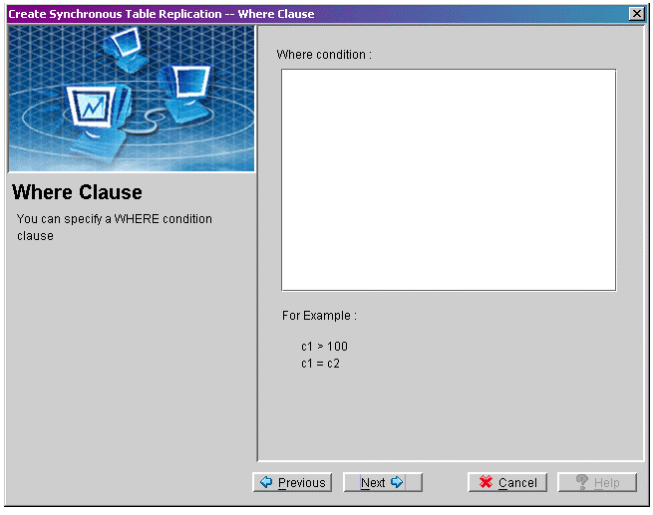
5. Click **Next**. The **Select Source Columns to be Replicated** window will appear.



6. Select source columns from the source table. A list of available columns on the selected table appears on the left hand list box. Select source columns by double clicking on the column name or selecting the column and clicking the **add** button. The **add all** button can be used to add all columns to the replication. All the selected source columns will be displayed in the list box to the right. Columns can be likewise removed from the replication by double clicking on the column name or selecting the column and clicking the **remove** button.

NOTE *You must select all primary key columns from the source table for replication.*

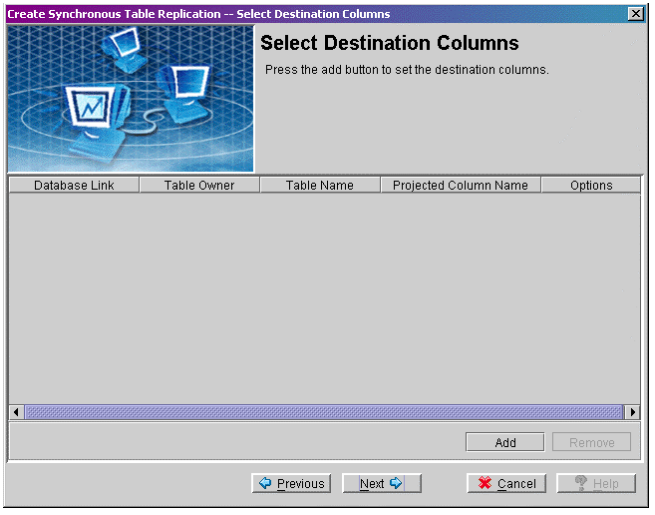
7. Click **Next**. The **Where Clause** window will open.



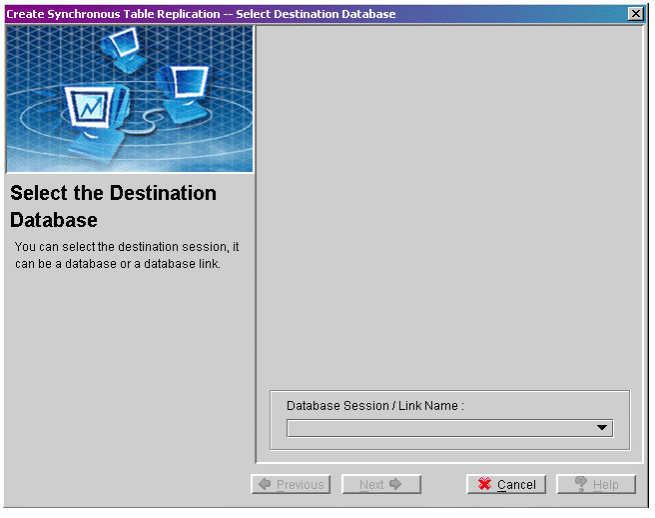
8. Add a WHERE condition clause to the SQL script, if desired. This is useful if you want to place constraints on the data that is to be replicated.

NOTE *The WHERE clause must follow proper SQL syntax. Refer to the section on Constraint Syntax at the end of chapter 4 for more information.*

9. Click **Next**. The **Select Destination Columns** window will appear.

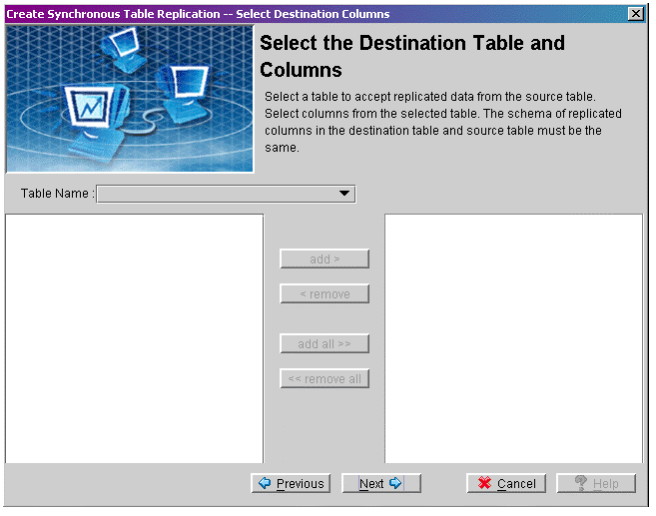


10. Add destination tables to the replication. Click **Add**. The **Select Destination Database** window will open.



11. You can choose to enter the destination database information from the **Database Session / Link Name** menu at the bottom of the page.

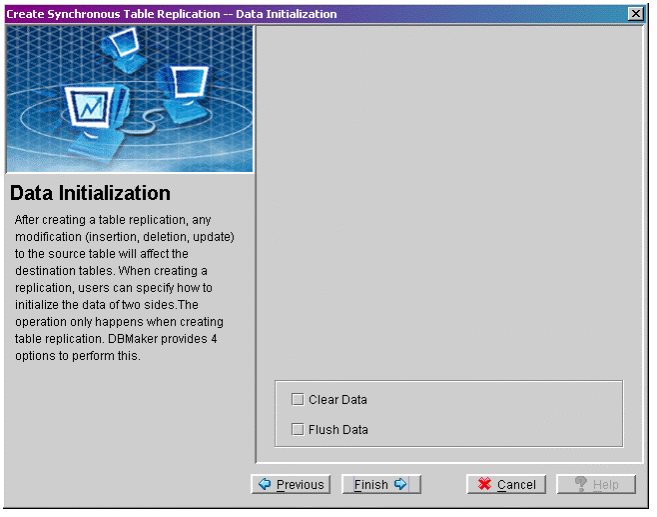
12. Click **Next**. The **Select the Destination Table and Columns** window will open.



13. Select a destination table from the **Table Name** menu. A list of available columns on the selected table will appear on the left hand list box.

14. Select destination columns by double clicking on the column name or selecting the column and clicking **add** button. The **add all** button can be used to add all columns. All the selected destination columns will be displayed in the list box to the right. Columns can be likewise removed from the replication by double clicking on the column name or selecting the column and clicking the **remove** button.

15. Select **Next**. The **Data Initialization** window will appear.

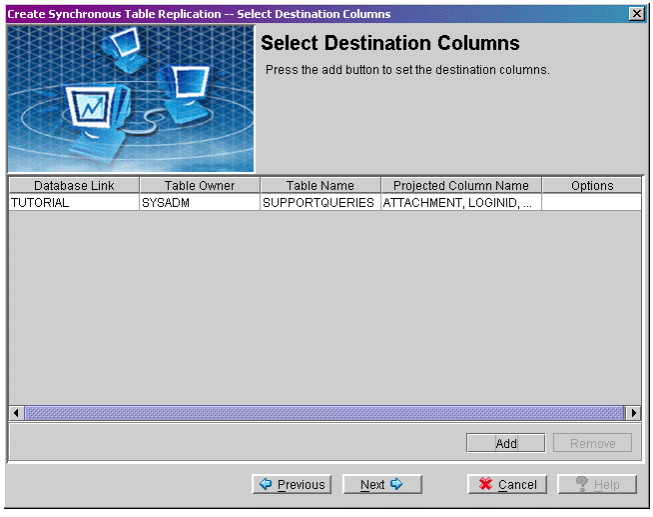


16. Select the desired options by checking the check box next to the appropriate option.

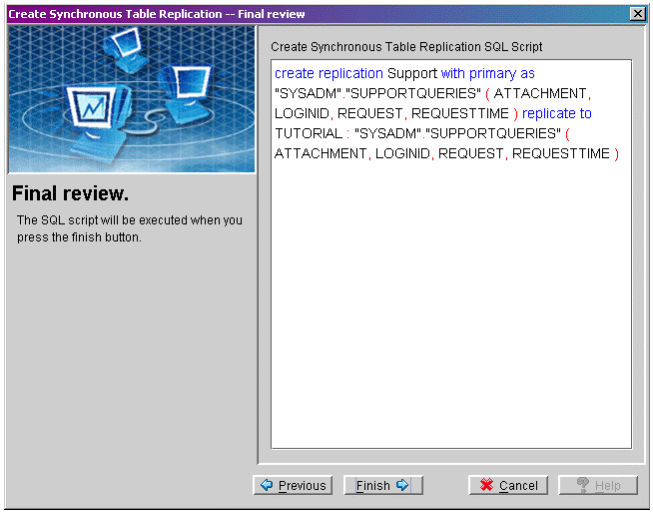
Clear Data: all data in the destination table is deleted when the replication is created.

Flush Data: copies all data that matches search criteria into the destination table.

17. Click **Next**. The **Select Destination Columns** window will reappear.

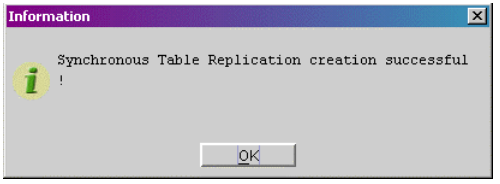


18. Click **Add** and repeat steps 7 – 13 if you want to add more columns, otherwise click **Next**.



19. Review the SQL script. You can edit the script manually by placing the cursor in the text field or clicking on **Previous** to return to prior windows. Click

Finish when you are satisfied with the SQL statement. The **Information** dialog box appears



20. Click **OK**

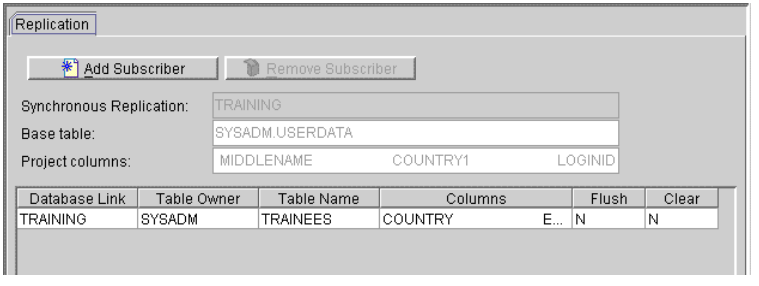
Adding Subscribers to a Synchronous Replication

After a replication has been created, other subscribers may be added to the replication. Other tables in the same database, or other databases can be subscribers. Subscriber destination tables must meet all requirements of synchronous table replication: the schema of source and destination columns must match, and the databases must have an active link

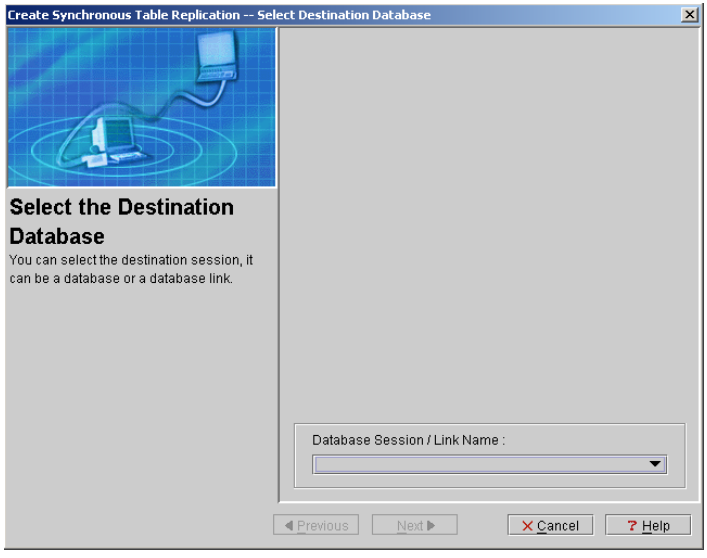
➲ **To add subscribers to a synchronous replication:**

1. Select the **Replication** node from the tree. A list of replications appears on the **Replication** page to the right.

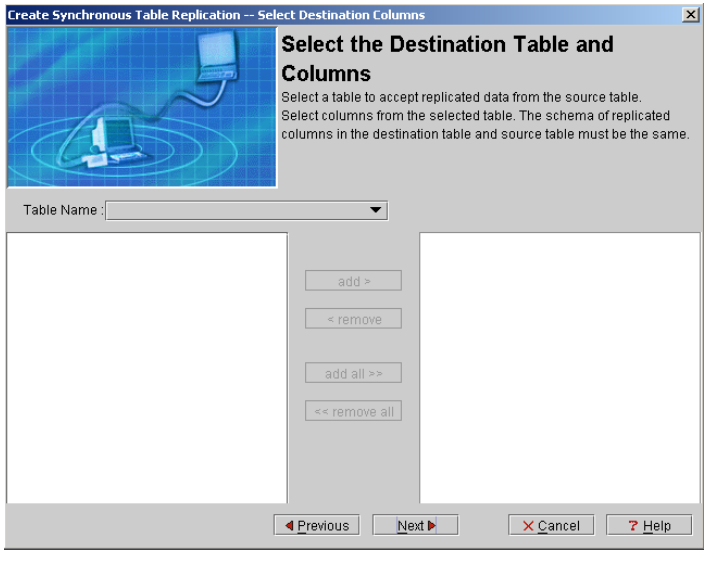
2. Select a replication from the list. The replication page will show subscriber information about the selected database in a table. Furthermore, the buttons **Add Subscriber** and **Remove Subscriber** will appear at the top of the page.



3. Click **Add Subscriber**. The **Select Destination Database** window will open.



4. Choose a database session from the **Database Session / Link Name** menu at the bottom of the page. Click **Next**.



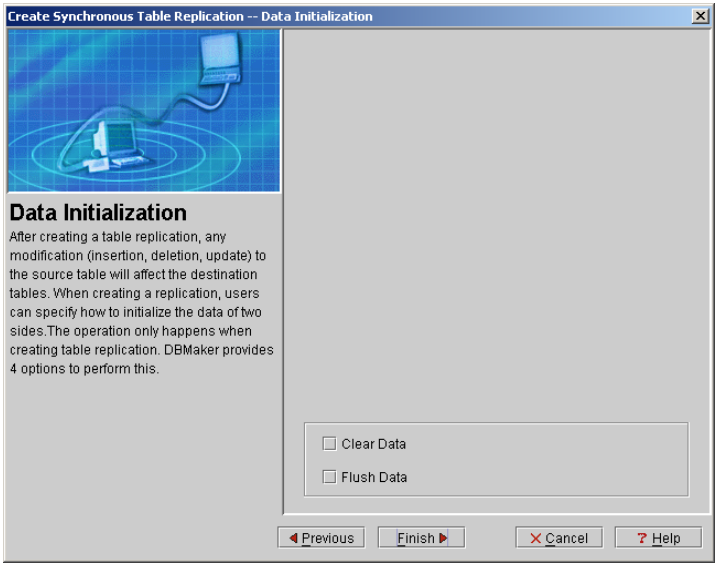
5. The **Select Destination Columns** window will prompt you to select destination columns and a destination table.

6. Select a destination table from the **Table Name** menu. A list of available columns on the selected table will appear on the left hand list box.

7. Select destination columns by double clicking on the column name or selecting the column and clicking the **add** button. The **add all** button can be used to add all columns. All the selected destination columns will be displayed in the list box to the right. Columns can be likewise removed from the replication by double clicking on the column name or selecting the column and clicking the **remove** button.

NOTE *The schema of the destination columns must match the schema of the source columns.*

8. Select **Next** after the desired columns have been selected.

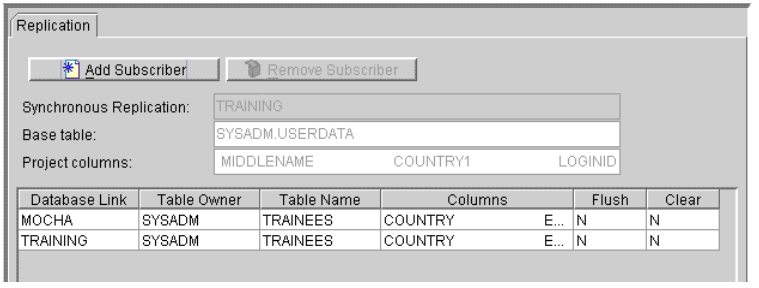


9. The **Data Initialization** window offers you two options. Select the desired options by checking the check box next to the appropriate option.

Clear Data: all data in the destination table is deleted when the replication is created.

Flush Data: copies all data that matches a search criteria into the destination table.

10. Select **Finish**. The new replication will be added to the list of subscriptions.



15.2 Asynchronous Table Replication

'Asynchronous' means that the modification to the remote site will be delayed. It is significantly slower than synchronous table replication. The time difference from source to destination database depends on a user-defined schedule. This is the replication schedule. Modifications to local tables are stored in replication logs, and are replicated to the remote table according to the predefined schedule. Using replication logs enables DBMaker to treat the local transaction and the remote transaction independently, allowing you to update local tables normally even if the remote connection is not available. This allows asynchronous table replications to tolerate network and remote database failures, since DBMaker will keep trying to replicate until any failures are corrected or until the user specified number of attempts to reconnect is reached.

Asynchronous table replication uses ODBC function calls to communicate with remote databases. This might give poor performance in a Wide Area Network (WAN) environment. To achieve better performance in a WAN, DBMaker provides another mechanism named Express Asynchronous table replication whereby SQL Commands and related data applied to the source table are packed and sent to the remote database. Since other DBMS packages do not support this protocol, Express Asynchronous Table Replication cannot work with heterogeneous replication. It also does not support the STOP ON ERROR option.

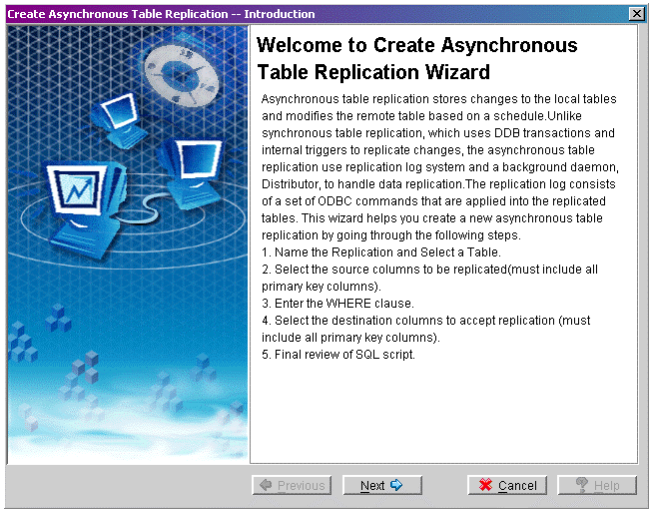
Adding Asynchronous Table Replication

A replication schedule must exist for asynchronous table replication to occur, and all associated settings for the replication schedule set. Furthermore, the schema of the destination columns must match the schema of the source columns and primary key columns in the source table must be replicated.

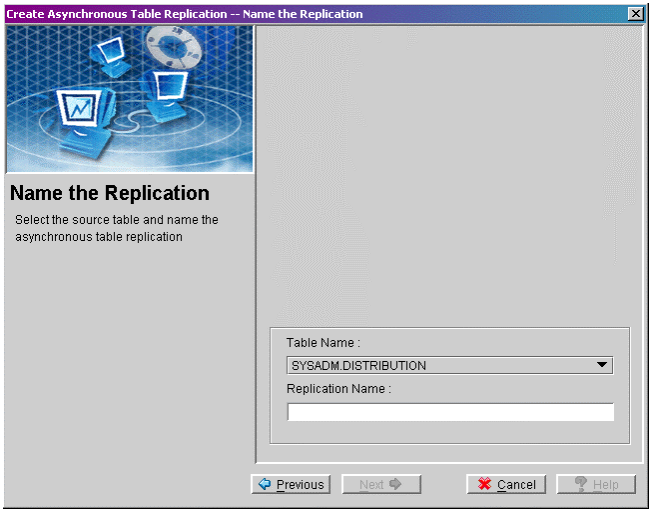
➲ **To create an Asynchronous Table Replication:**

1. Select **Create** from the top of the **Replication** page. The **Create Replication** window will appear.

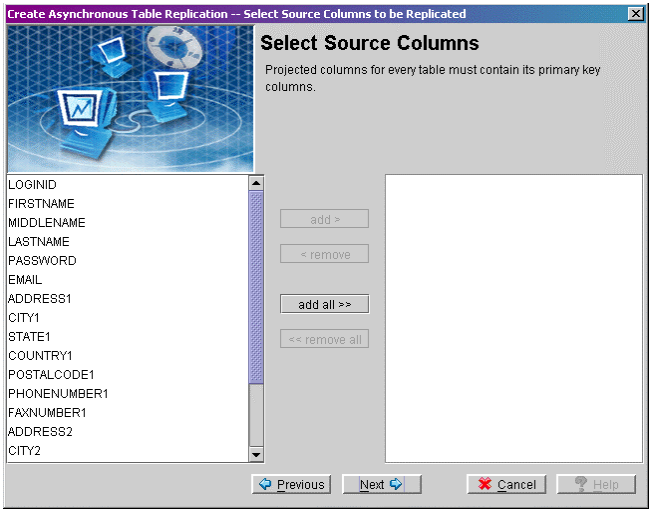
2. Select **Asynchronous Table Replication** from the menu, the **Create Asynchronous Table Replication Wizard** will begin.



3. Select **Next**. The **Name the Replication** window will appear.

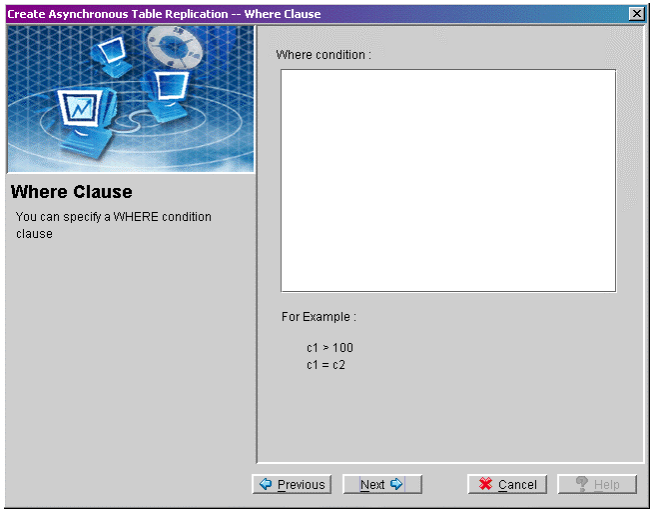


4. The **Asynchronous Table Replication Wizard** will prompt you to identify the source table to replicate and name the replication. Select the source table from the **Table Name** menu at the bottom of the window. Enter a name into the **Replication Name** field and Click **Next**.



5. The **Select Source Columns to be Replicated** window will prompt you to select source columns from the source table. A list of available columns on the selected table appears on the left hand list box. Select source columns by double clicking on the column name or selecting the column and clicking **add**. The **add all** button can be used to add all columns to the replication. All the selected source columns will be displayed in the list box to the right. Columns can be likewise removed from the replication by double clicking on the column name or selecting the column and clicking **remove**.

6. Click **Next** after the desired columns have been selected.

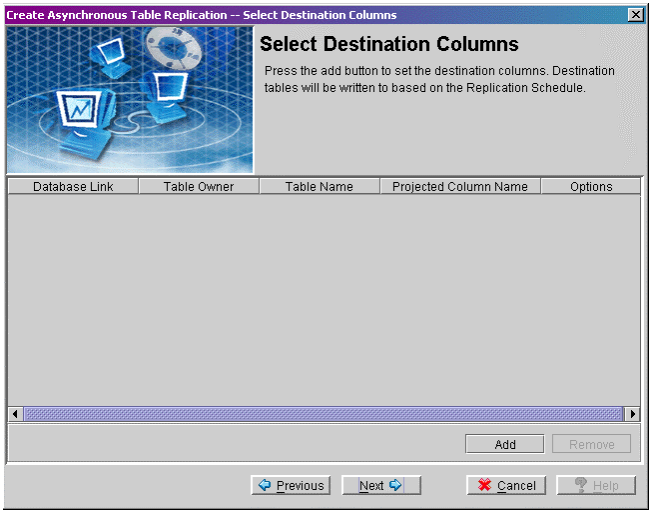


NOTE *You must select all primary key columns from the source table for replication.*

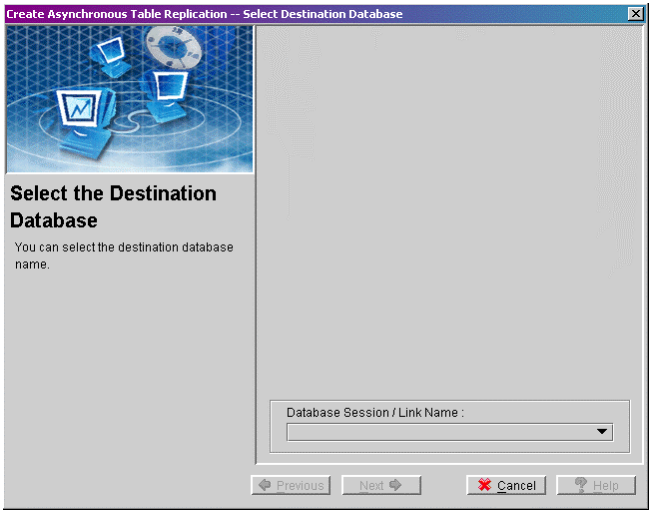
7. The **Where Clause** window allows you to add a WHERE condition clause to the SQL script. This is useful if you want to place constraints on the data that is to be replicated. Click **Next** to continue.

NOTE *The WHERE clause must follow proper SQL syntax. Refer to the section on Constraint Syntax at the end of chapter 4 for more information.*

8. The **Asynchronous Table Replication Wizard** will next prompt you to add destination columns to the replication.



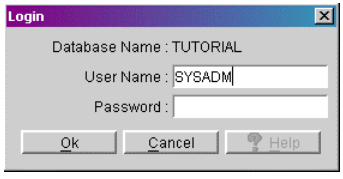
9. Click **Add** at the bottom of the **Select Destination Columns** window. The **Select Destination Database** window will open.



10. You can choose to enter the destination database information manually or from the **Database Session / Link Name** menu at the bottom of the page. To

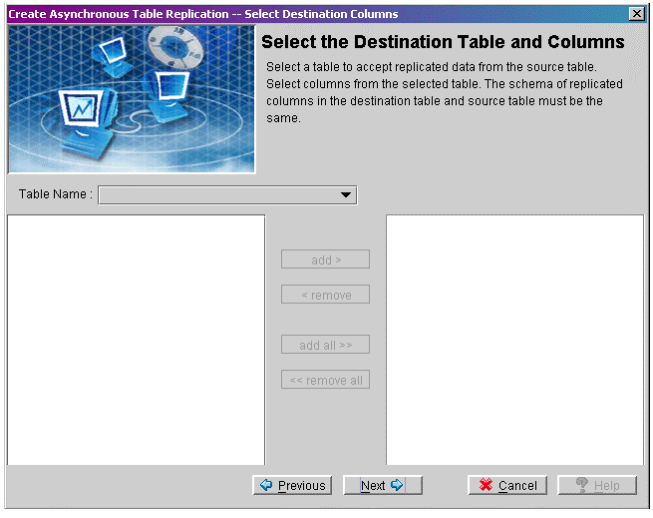
enter the information manually, select the **Enter Data Manually** check box and provide the required information in the fields below.

11. Click **Next**.



12. JDBA will prompt you to log on to the destination database. Enter a User Name and Password for a user with DBA authority or SYSADM.

13. Click **OK**, the **Select Destination Columns** window will open.



14. The **Select Destination Columns** window will prompt you to select destination columns and a destination table.

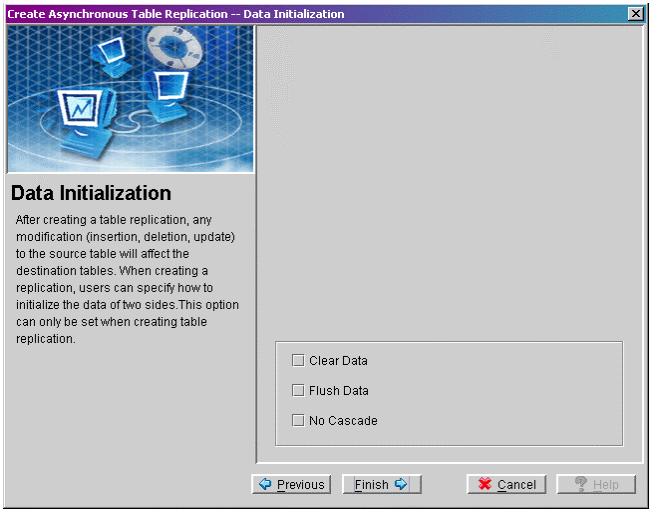
15. Select a destination table from the **Table Name** menu. A list of available columns on the selected table will appear on the left hand list box.

16. Select destination columns by double clicking on the column name or selecting the column and clicking the **add** button. The **add all** button can be used to add all columns. All the selected destination columns will be displayed in the list box to the right. Columns can be likewise removed from the replication by

double clicking on the column name or selecting the column and clicking the **remove** button.

NOTE *The schema of the destination columns must match the schema of the source columns.*

17. Click **Next** after the desired columns have been selected.



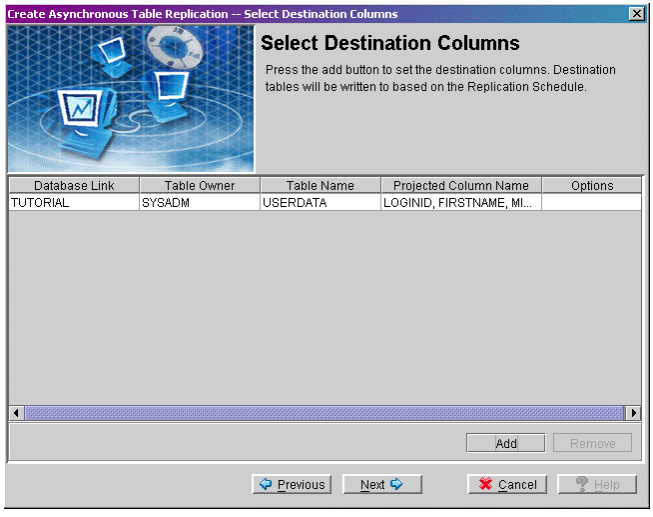
18. The **Data Initialization** window offers you three options. Select the desired options by checking the check box next to the appropriate option.

Clear Data: all data in the destination table is deleted when the replication is created.

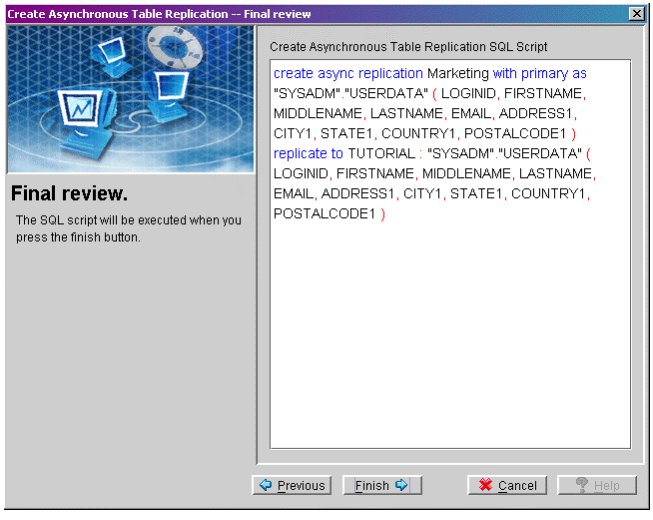
Flush Data: copies all data that matches search criteria into the destination table.

No Cascade: prevents replicated data from being replicated from the destination tables. If this option is not selected, data can be replicated from the source database to the destination databases down to other databases that are subscribers to the destination databases.

19. Click **Next**. The **Select Destination Columns** window will reappear.

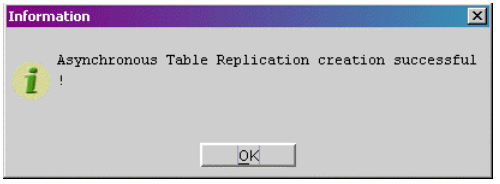


20. Click **Add** and repeat steps 7 – 13 if you want to add more columns, otherwise click **Next**.



21. Review the SQL script. You can edit the script manually by placing the cursor in the text field or click **Previous** to return to prior windows. Click **Finish**

when you are satisfied with the SQL statement. The **Information** dialog box appears.



22. Click **OK**.

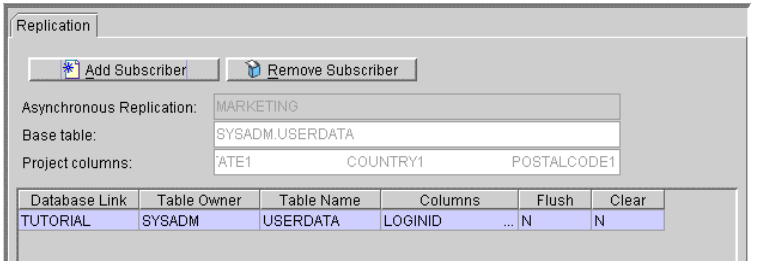
Adding Subscribers to an Asynchronous Replication

After a replication has been created, other subscribers may be added to the replication. Other tables in the same database, or other databases can be subscribers. Subscriber destination tables must meet all requirements of asynchronous table replication: the schema of source and destination columns must match, and a replication schedule must exist.

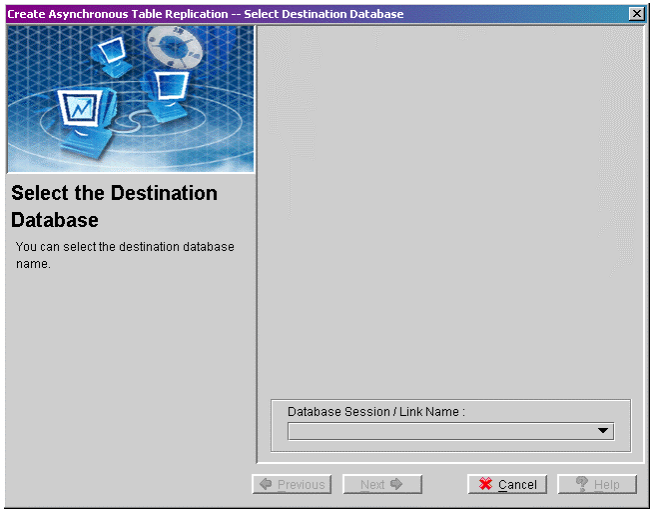
➲ **To add subscriber tables to an Asynchronous Replication:**

1. Select the **Replication** node from the tree. A list of replications appears on the **Replication** page to the right.

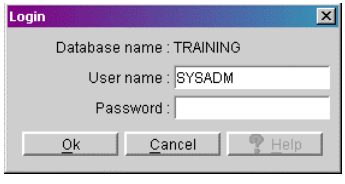
2. Select a replication from the list. The replication page will show subscriber information about the selected database in a table. Furthermore, the buttons **Add Subscriber** and **Remove Subscriber** will appear at the top of the page.



3. Click **Add Subscriber**. The **Select Destination Database** wizard window will open.



4. Select the destination database from the **Database Session / Link Name** menu. Click **Next**.



5. JDBA will prompt you to log on to the destination database. Enter a User Name and Password for a user with DBA authority or SYSADM. Click **OK**, the **Select Destination Columns** window will open.

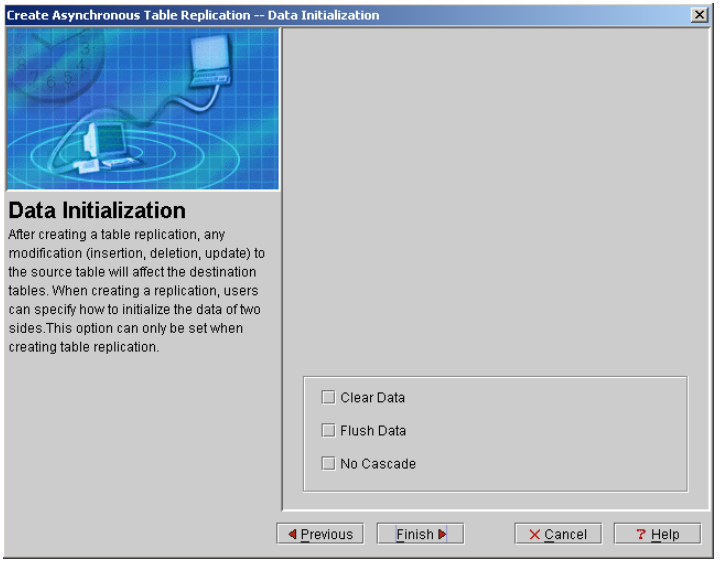
6. The **Select Destination Columns** window will prompt you to select destination columns and a destination table.

7. Select a destination table from the **Table Name** menu. A list of available columns on the selected table will appear on the left hand list box.

8. Select destination columns by double clicking on the column name or selecting the column and clicking the **add** button. The **add all** button can be used to add all columns. All the selected destination columns will be displayed in the list box to the right. Columns can be likewise removed from the replication by double clicking on the column name or selecting the column and clicking the **remove** button.

NOTE *The schema of the destination columns must match the schema of the source columns.*

9. Select **Next** after the desired columns have been selected.



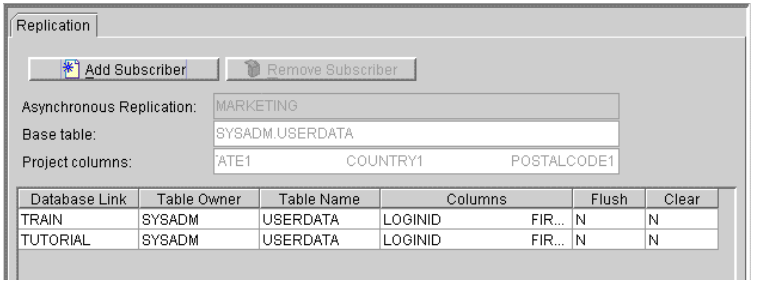
10. The **Data Initialization** window offers you two options. Select the desired options by checking the check box next to the appropriate option.

Clear Data: all data in the destination table is deleted when the replication is created.

Flush Data: copies all data that matches search criteria into the destination table.

No Cascade: prevents replicated data from being replicated from the destination tables. If this option is not selected, data can be replicated from the source database to the destination databases down to other databases that are subscribers to the destination databases.

11. Select **Finish**. The new database link will appear in the table on the **Replication** page.

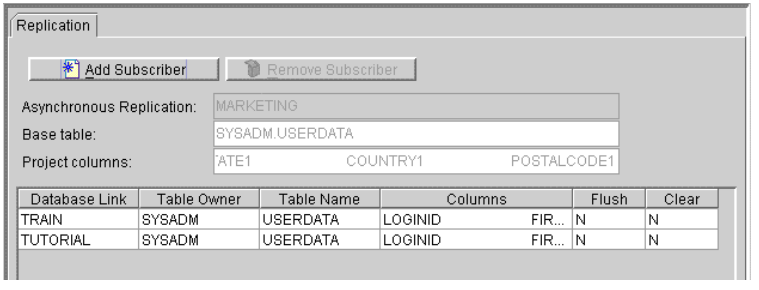


15.3 Dropping a Subscriber

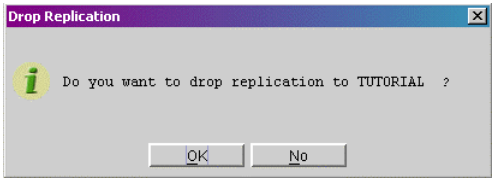
Subscribers may be dropped when the data no longer needs to be replicated. Subscribers should be dropped whenever the subscriber's destination columns or databases are deleted.

➲ **To drop a subscriber:**

1. Select the **Replication** node from the tree. The tree will expand, showing a list of replications below the **Replication** node on the tree.
2. Select a replication from the tree. The replication page will show subscriber information about the selected database in a table. Furthermore, the buttons **Add Subscriber** and **Remove Subscriber** will appear at the top of the page.



3. Select a database link to drop a subscription (it will highlight blue).
4. Click **Remove Subscriber**. The **Information** dialog box will appear asking conformation.



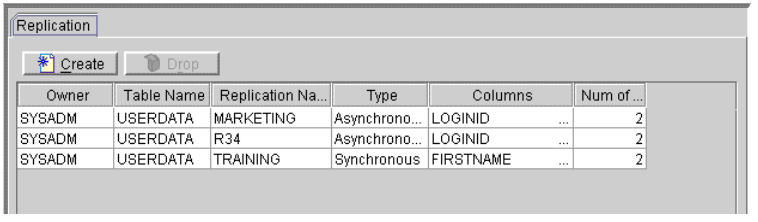
5. Click **OK**.

15.4 Dropping a Replication

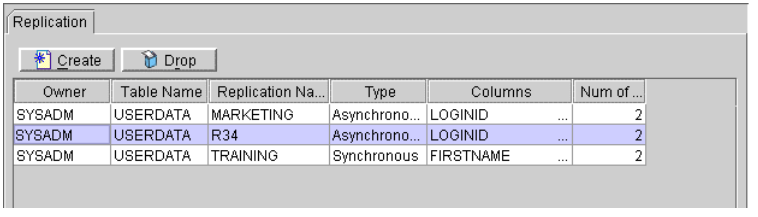
A replication may be dropped if there are no subscribers an no further need to replicate the data.

➲ **To drop a replication:**

1. Select the **Replication** node from the tree. A list of replications appears on the **Replication** page to the right.



2. Select a replication from the list (it will highlight blue). The **Drop** button at the top of the page will activate.



3. Click **Drop**. The **Information** dialog box will appear asking conformation.

4. Click **OK**.

16 Working with Domains

A domain is a type of integrity constraint you can use when defining a column. Domains specify the data type of the column, and may also specify a default value and a value constraint. When you define a column using a domain, the column inherits the properties of the domain (data type, default value, and value constraint) without requiring you to explicitly specify them.

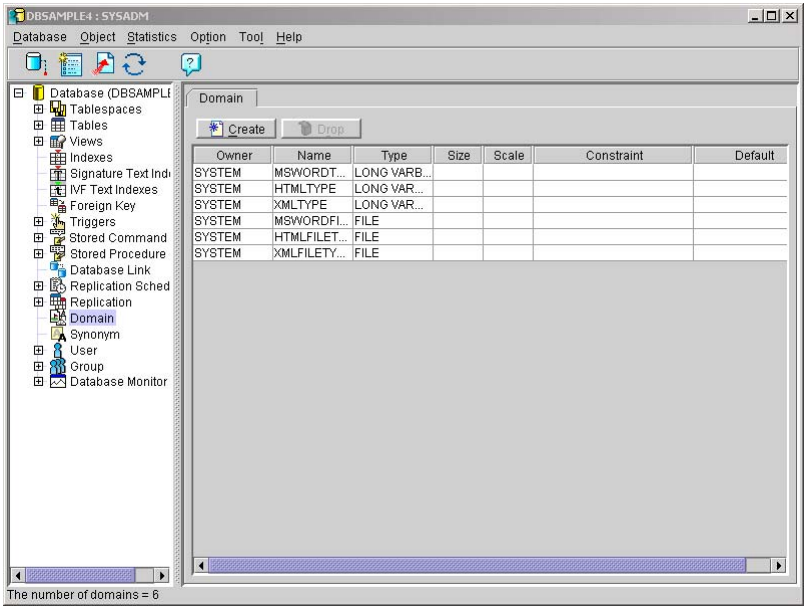
Specifying the default value and value constraint using domains achieves the same results as specifying them in a standard column definition. If you specify a default value for a column, it will override the default value specified in a domain. If you specify any value constraints in the column definition, they will be used in addition to the value constraints specified in the domain. If you do define a column using a domain and specify additional value constraints, you should ensure the additional value constraints do not conflict with those defined in the domain.

16.1 Creating a Domain

Defining a domain name, an optional default value and an optional constraint creates a domain. Any data type supported by DBMaker except the SERIAL type can be used in domains.

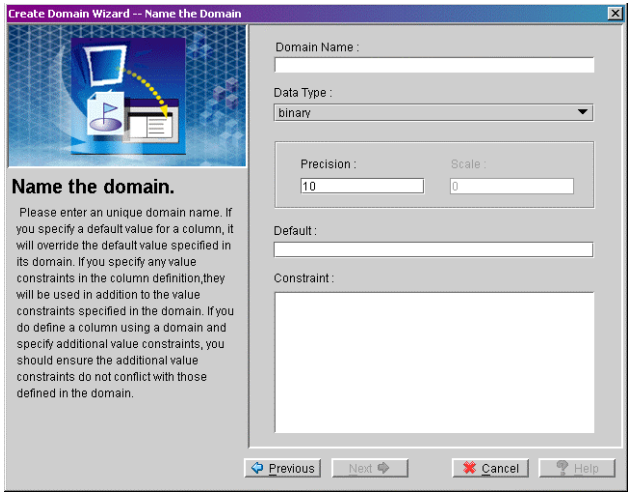
➲ **To create a Domain:**

1. Select the object **Domain** from the tree. The following window is displayed.



2. Click **Create**. The **Create Domain Wizard** window is opened.

3. Click **Next**.

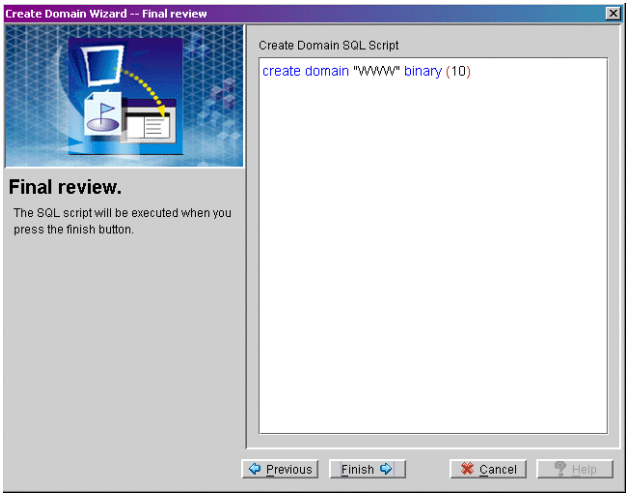


4. Enter the **Domain Name**.

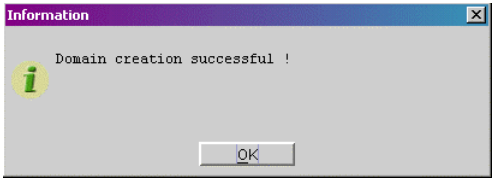
5. Select a data type from the **Data Type** menu.

6. Enter the **Precision** and **Scale** information corresponding to the selected Data Type.

7. Enter the **Default** value.
8. Enter the **Constraint** of the column. Constraints must follow proper SQL syntax. See the section *Constraint Syntax* at the end of Chapter 4
9. Click **Next** .



10. Check the SQL script and make changes if required.
11. Click **Finish**. The **Information** dialog box appears.



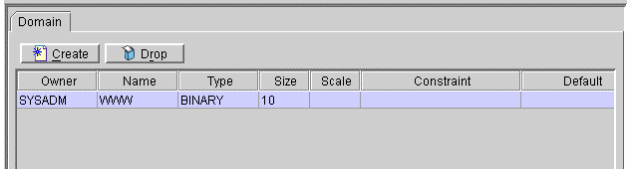
12. Click **OK**.

16.2 Dropping a Domain

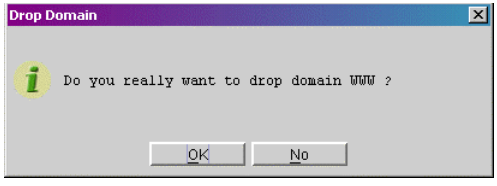
If a domain is no longer required, it can be dropped.

➲ **To drop a Domain:**

1. Select the object **Domain** from the tree. The list of all the Domains in the database is displayed.
2. Select the Domain that is to be dropped as shown below.



3. Click **Drop**. A confirmation window is shown to confirm if you want to drop the Domain.



4. Click **OK** and the confirmation window disappears.

17 Working with Synonyms

A *synonym* is an alias for any table or view. Since a synonym is simply an alias, it requires no storage other than its definition in the system catalog.

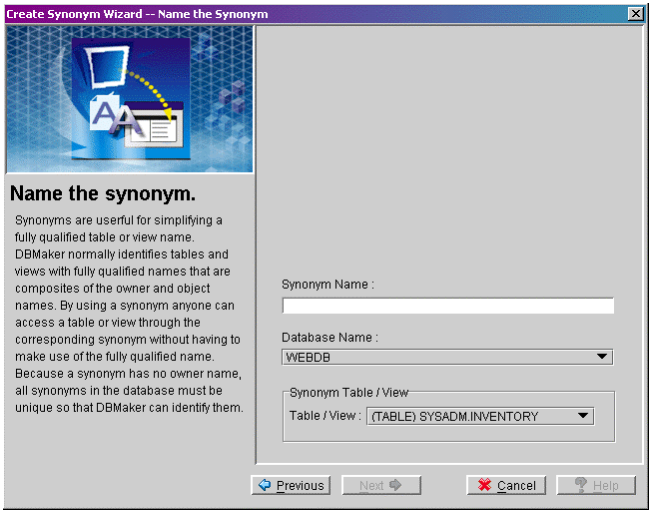
Synonyms are useful for simplifying a fully qualified table or view name. DBMaker normally identifies tables and views with fully qualified names that are composites of the owner and object names. By using a synonym anyone can access a table or view through the corresponding synonym without having to make use of the fully qualified name. Since a synonym has no owner name, all synonyms in the database must be unique for DBMaker to identify it.

17.1 Creating a Synonym

You can create a synonym for a table or view.

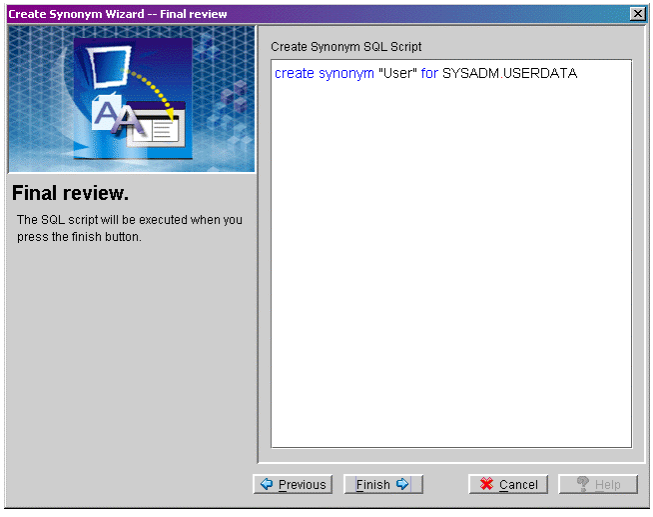
➲ **To create a synonym for a table or view:**

1. Select the object **Synonym** from the tree and click the **Create** button.
2. The **Create Synonym Wizard - Introduction** window will open.
3. Click **Next**. The **Name the Synonym** window will open.



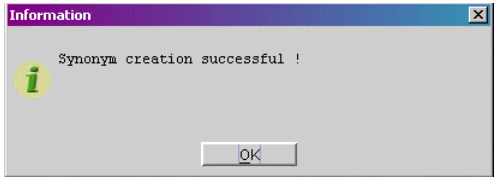
4. Enter the **Synonym Name**.
5. Select the **Table/View** name from the menu.

6. Click **Next**. The **Final Review** window will open.



7. Check the SQL script and make changes if required. Click **Finish**.

8. The **Information** dialog box will open.



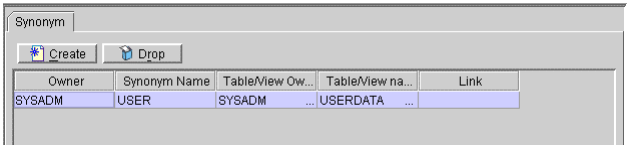
9. Click **OK**.

17.2 Dropping a Synonym

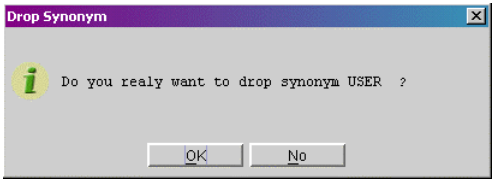
If a synonym is no longer required, it can be dropped.

➲ **To drop a Synonym:**

1. Select the object **Synonym** from the tree and select the Synonym from the list.



2. Click **Drop**. The **Drop Synonym** window will open to confirm if you want to drop the window.



3. Click **OK**.

18 Managing Users

DBMaker supports multiple users and allows database privileges to be granted or revoked to those users. JDBA tool provides an easy to use interface for creating new users, changing user passwords, and adding or revoking user privileges. User names have a maximum length of thirty-two characters, passwords have a maximum length of sixteen characters, and both may contain letters, numbers, the underscore character, and the symbols $ and #. The first character may not be a number. You must be logged onto the database as SYSADM to be able to create a new user or change database authority for any user.

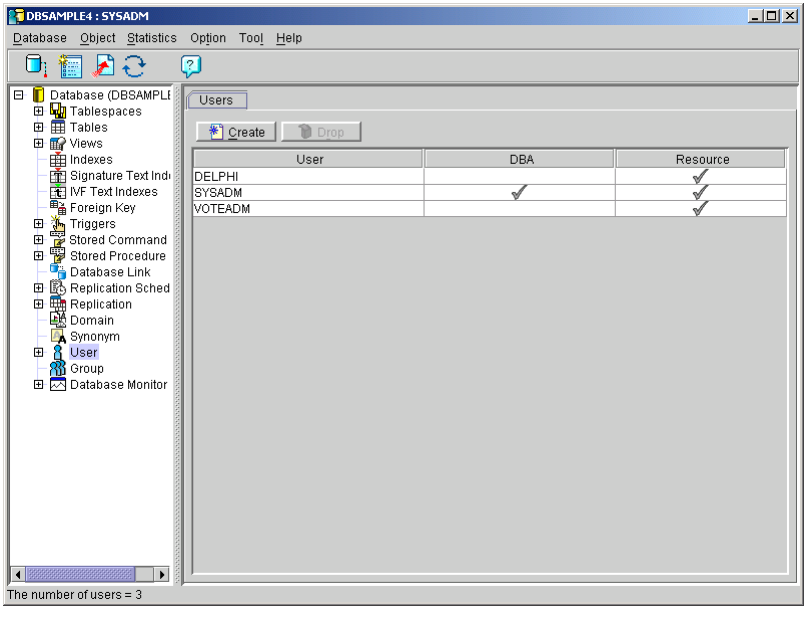
Object privileges include the SELECT, INSERT, DELETE, UPDATE, INDEX, ALTER, and REFERENCE privileges for tables and views, and the EXECUTE privilege for stored commands and stored procedures. Object privileges may be granted or revoked by the user that created the object, or by a user with DBA authority or higher.

18.1 Creating a New User Account

New accounts should always be created when a specific access profile is required for an individual. It is also useful to create user accounts for all individuals accessing the database to facilitate user management. User names have a maximum length of thirty-two characters, passwords have a maximum length of sixteen characters, and both may contain letters, numbers, the underscore character, and the symbols $ and #. The first character may not be a number.

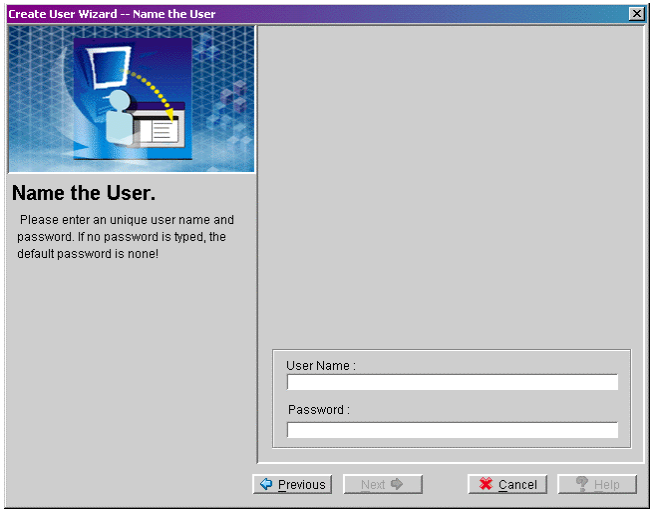
➲ **To create a new user account:**

1. Open the **User** node on the object tree. The **Users** window will open.



2. Click **Create** from the top of the **Users** page. The **Create User Wizard** will start.

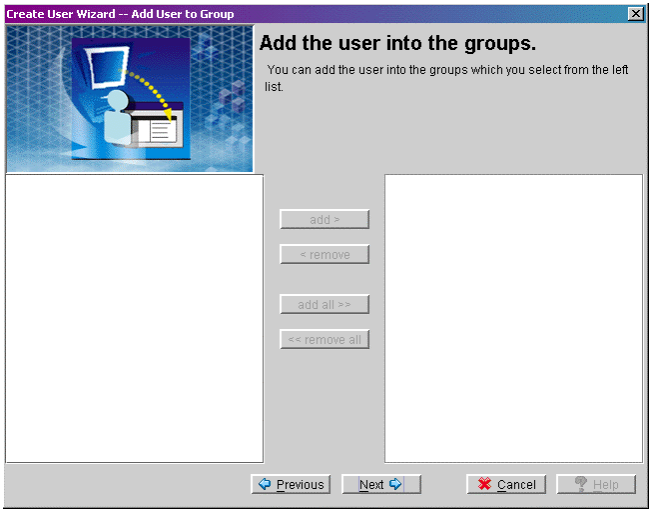
3. Click **Next**. The **Name the User** window will open.



4. Enter a name into the **User Name** field, and a password in the **Password** field.

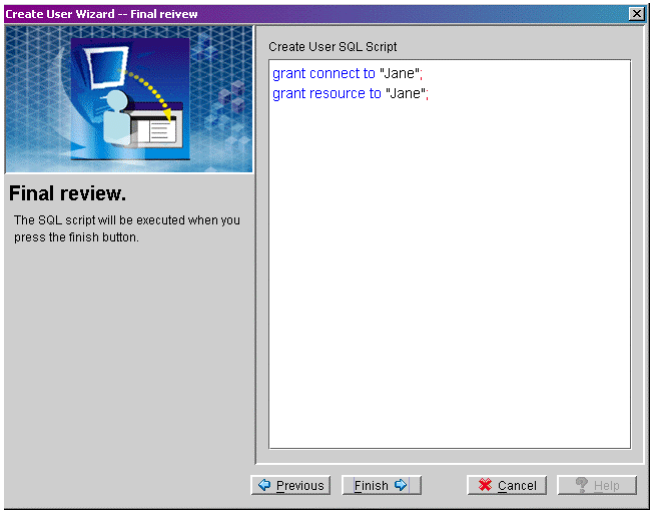
5. Click **Next**. The **Grant DBA or Resource to User** window will open.

6. Select which type of database authority the new user will have. If no authority level is selected, the new user is granted connect authority. Click **Next**, the following window will open.

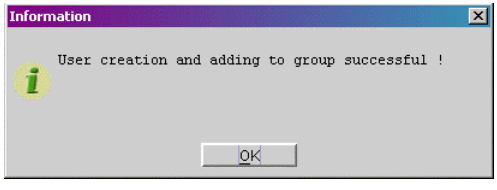


7. A list of available groups appears on the left list box. Select a group by double clicking on the group name or selecting the group(s) and clicking the **add** button. The **add all** button can be used to select all groups to add the user into. All the selected groups will be displayed in the list box to the right. Groups can be, likewise removed by double clicking on the group name, or by selecting the name and clicking the **remove** button.

8. Click **Next**, the **Final review** page will appear.



9. Review the SQL script that has been created. You can add to the SQL script in the **Create User SQL Script** field if necessary by clicking on the field. Click **Finish** to execute the SQL command. The **Information** dialog box will appear.



10. Click **OK**.

18.2 Changing Security Level

There are two security mechanisms in DBMaker: user authority level and object privilege. Authority determines the level of access and control the user has to grant or revoke object privileges, as well as the ability to create new objects. Object privilege determines the level of access and control an individual user has over database objects, such as tables, views, stored procedures, and stored commands.

Setting Authority Level

There are three levels of authority: CONNECT, RESOURCE, and DBA.

Any users that are created but have not yet been granted RESOURCE or DBA authority will have CONNECT authority by default. A user with CONNECT authority may create temporary tables in a database, or perform queries on any data to which they have explicitly been granted permission.

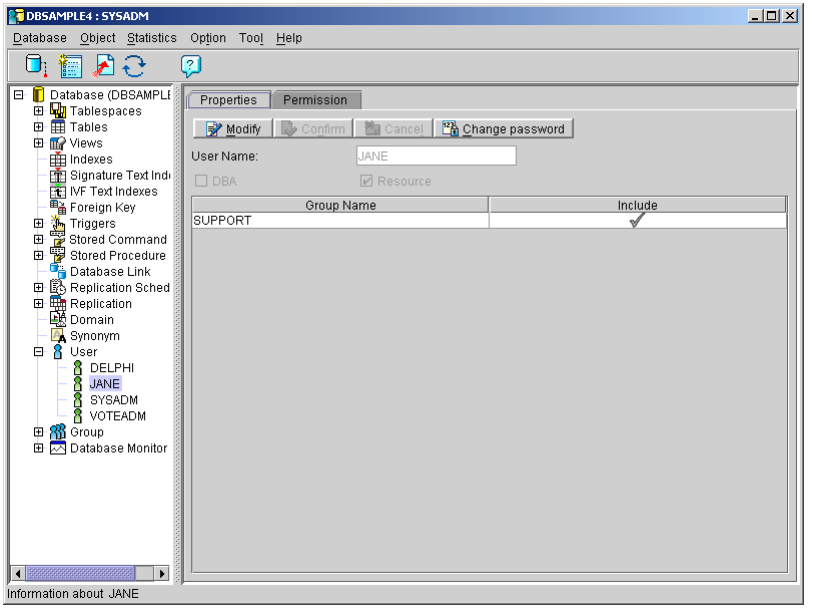
A user that is created and granted RESOURCE authority can create, alter, and drop tables, domains, and indexes. As the owner of any objects they create, users with RESOURCE authority may grant and revoke object privileges to other users and create synonyms and views for any objects they own.

DBA authority has all same capabilities as RESOURCE authority, but may additionally create tablespaces and files. Users with DBA authority can also grant or revoke object privileges for schema objects owned by other users (except for system schema objects).

➲ **To set a user's authority level:**

1. Open the **User** node on the object tree. The **Users** page will open.

2. The current authority level of the user will be displayed on the users table. Double click the user name in the tree or on the users table. The **Properties** page will be displayed.



3. Select **Modify** to change the user's database authority.

4. A check mark next to the type of database authority indicates the user has that privilege. Clicking the check box toggles the authority of the user. Click **Confirm** to save the changes, or **Cancel** to discard them.

Granting and Removing Object Privileges

Object privileges control which database objects a user can access and the actions they can perform. There are seven object privileges: SELECT, INSERT, DELETE, UPDATE, INDEX, ALTER, and REFERENCE.

SELECT privilege allows you to select the data in a database object. This privilege applies to the entire object and cannot be granted on specific columns.

INSERT privilege allows you to insert new data into a database object. This privilege can also be restricted to specific columns.

DELETE privilege allows you to delete data from a database object. This privilege applies to the entire object and cannot be granted on specific columns.

UPDATE privilege allows you to update data in a database object. This privilege can also be restricted to specific columns.

INDEX privilege allows you to create an index on a database object. This privilege applies to the entire object and cannot be granted on specific columns.

ALTER privilege allows you to alter the schema of a database object. This privilege applies to the entire object and cannot be granted on specific columns.

REFERENCE privilege allows you to create referential constraints (such as foreign keys) on a database object. This privilege can also be restricted to specific columns.

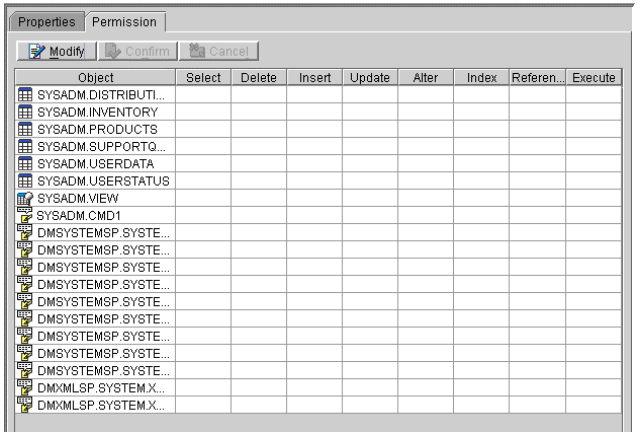
EXECUTE privilege allows the user to use a stored procedure or command.

The user who creates a schema object is the owner of that object. The owner and any user with DBA or SYSADM security privileges automatically have all object privileges on that object. System catalog tables belong to a special virtual user called SYSTEM. All users including the SYSADM have only SELECT privilege on system catalog tables. You cannot grant additional object privileges on the system catalog tables.

You cannot grant privileges on specific columns and on the entire database object at the same time. If you wish to do so you must use the command twice, once to grant privileges on specific columns, and once to grant privileges on the entire table. It is possible to grant object privileges to all users simultaneously by granting the privileges to PUBLIC. All current and future users will then have those privileges on the database object.

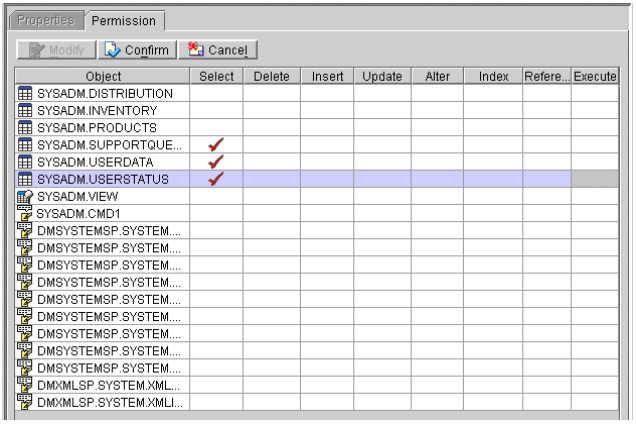
➲ **To Grant or Remove Object Privileges:**

1. Open the **User** node on the object tree. The **Users** page will open.
2. Double click the user name in the tree or on the users table. The **Properties** page will open.
3. Select the **Permission** tab at the top of the page. The **Permission** page will appear.



NOTE *Users with DBA privilege have permission on all objects in the database.*

4. Click **Modify** to alter the user's object privilege. Choose which type of privilege to grant to the user on each object. Click the column and row corresponding to the object privilege to be granted and the object to grant the privilege to. Privileges can be likewise removed. A check mark will appear for each privilege selected as in the table below.



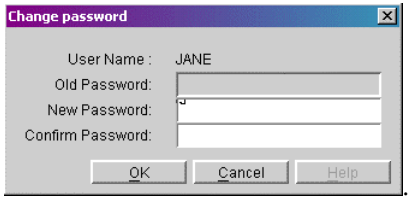
5. Click **Confirm** to save the changes, or **Cancel** to discard them.

18.3 Changing a User's Password

Passwords may be changed for a user if they have forgotten their old password or are requesting a new one. User names have a maximum length of thirty-two characters, passwords have a maximum length of sixteen characters, and both may contain letters, numbers, the underscore character, and the symbols $ and #. The first character may not be a number..

➲ **To change a user's password:**

1. Open the **User** node on the object tree. The **Users** page will open.
2. Double click the user name in the tree or on the users table. The **Properties** page will be displayed.
3. Select the **Change password** button to open the **Change Password** dialog box. Enter a new password and confirm it by reentering it in the **Confirm Password** field.



4. Click **OK** to change the password.

18.4 Deleting a User Account

User accounts should be deleted when the account is no longer active.

➲ **To remove a user account from the database:**

1. Open the **User** node on the object tree. The **Users** page will open.
2. Select the user account to be deleted from the table by clicking once (the user account will highlight blue).
3. Click **Drop**. A window will appear asking confirmation.
4. Click **OK** to remove the user from the database.

19 Managing Groups

Groups simplify the management of object privileges in a database with a large number of users. You can use a group to collect several users and/or groups. Any object privileges you grant to the group are automatically granted to all members of the group.

Members added to a group after object privileges have been granted gain those object privileges in addition to object privileges that have been granted to them directly. Members removed from a group lose object privileges that have been granted to the group, but retain any privileges that have been granted to them directly.

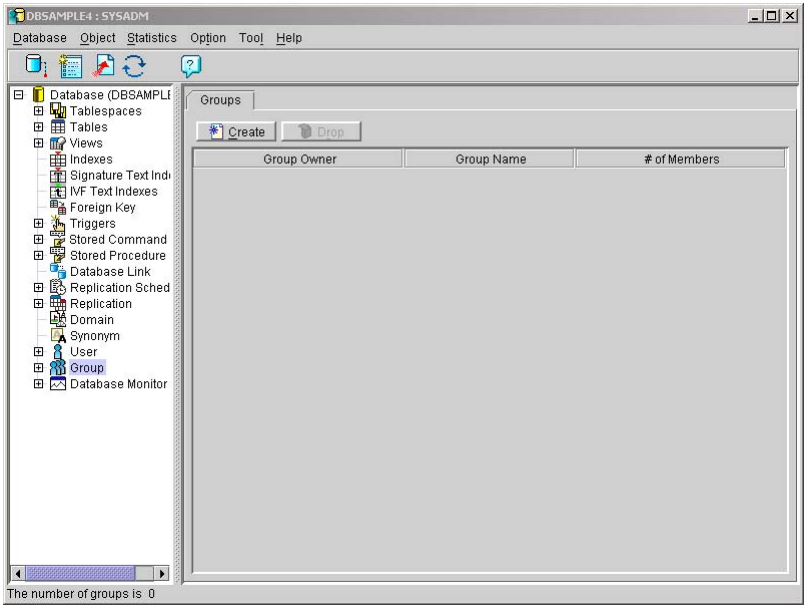
You can also specify a group name in place of a user name, as long as the group you are trying to add does not contain the group you are adding it to. User and group names have a maximum length of eight characters, and may contain letters, numbers, the underscore character, and the symbols $ and #. The first character may not be a number.

19.1 Creating a New Group

New group accounts should always be created when a specific access profile is required for a group of individuals. It is also useful to create group accounts that represent the structure of your organization, business, etc.

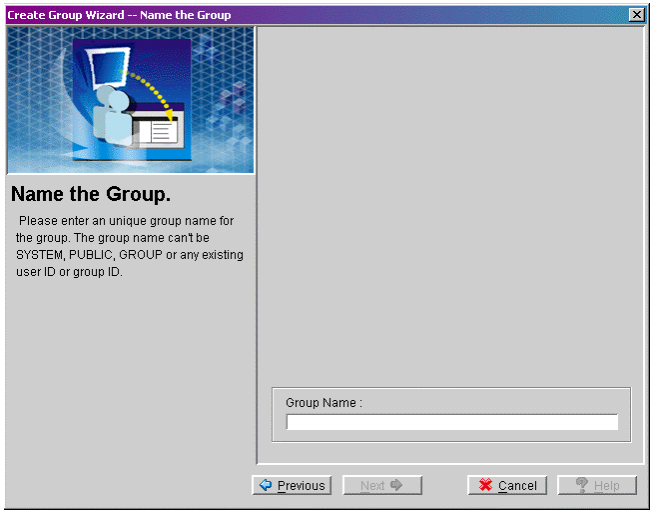
➲ **To create a new group:**

1. Select the **Group** node from the object tree. The **Groups** window will appear displaying all existing groups, the group owner, and the number of members in the group.

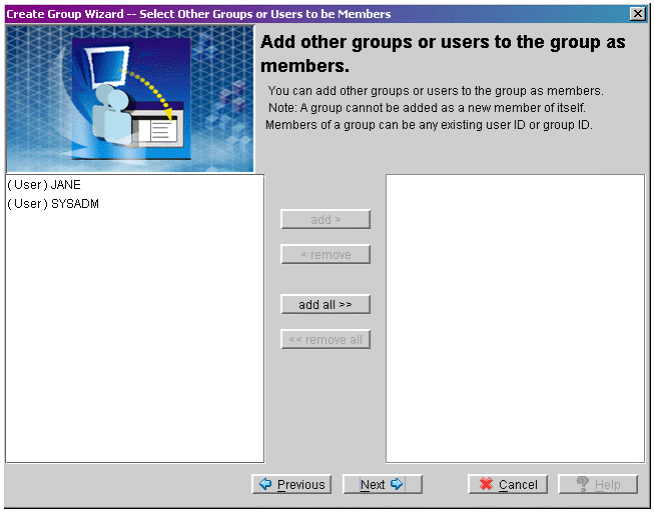


2. Click **Create**. The **Create New Group** wizard will start.

3. Click **Next**. The **Name the Group** window will appear.

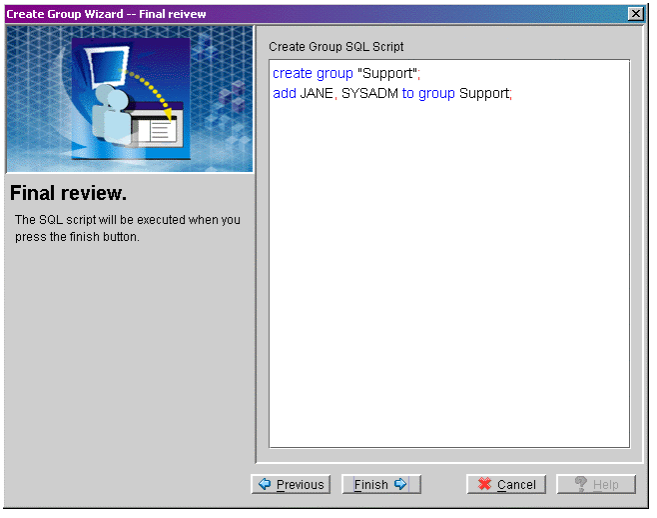


4. Enter the group name into the **Group Name** field. Click **Next**, the **Select Other Groups or Users to be Members** window will open.

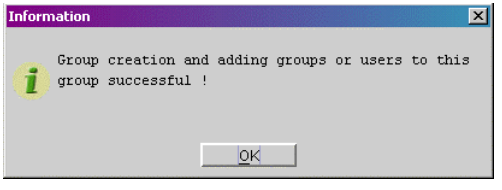


5. A list of available users/groups appears on the left list box. Select a user/ group by double clicking on the name or selecting the user(s)/ group(s) and clicking the **add** button. The **add all** button can be used to select all users/ groups to

the newly created group. All the selected users/ groups will be displayed in the list box to the right. Selected users/groups can be, likewise removed by double clicking on the name, or by selecting the users/groups and clicking the **remove** button. Click **Next**, the **Final review** page will appear.



6. Review the SQL script that has been created. You can add to the SQL script in the **Create Group SQL Script** field if necessary by clicking on the field. Click **Finish** to execute the SQL command. The **Information** dialog box will appear.



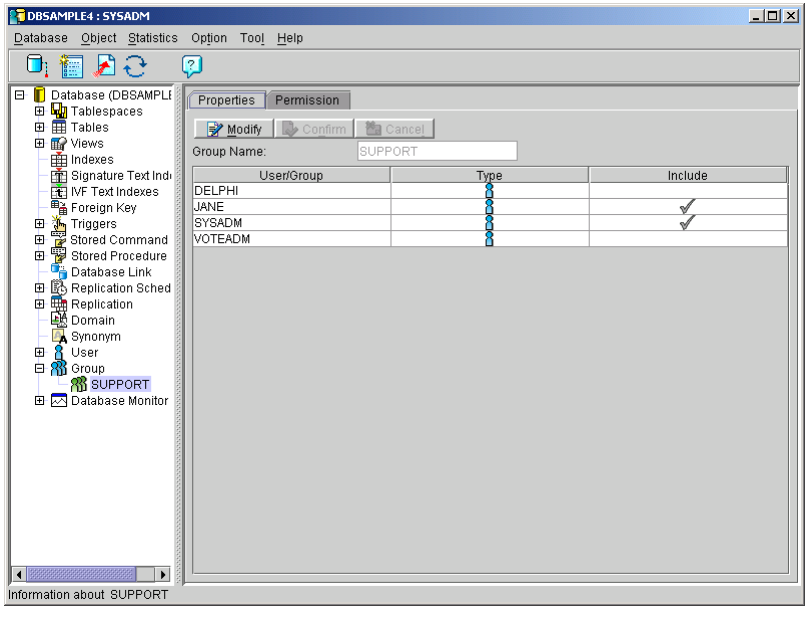
7. Click **OK**.

19.2 Adding and Removing Group Members

Members should be removed or added to groups according to their access needs or position within your organization. Users may be added to groups from the group properties page or the user's property page.

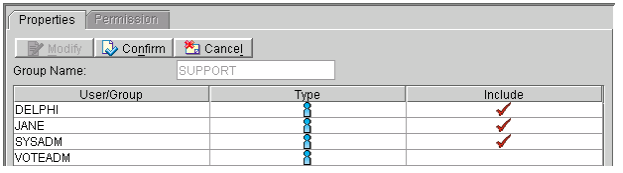
➲ **To add or remove members from a group:**

1. Select the **Group** node from the object tree. The **Groups** window will appear displaying all existing groups, the group owner, and the number of members in the group.



2. Select a group from the tree or double click the group in the **Groups** table. The group **properties** page will appear. The group **properties** table shows all existing users and groups for the database. If a check mark is present in the include column next to a user or group, then they are a member of the group indicated in the **Group Name** field.

3. Click **Modify** from the top of the **Properties** page to include more users or groups or to remove users or groups from the selected group. A check in the **Include** column indicates that the user or group is a member of the selected group.



NOTE *Users may also be added to groups by selecting the user from the user's node, and checking the* ***Include*** *column on the group that they are to be a member of.*

4. Click **Confirm** to save the changes, or **Cancel** to discard them.

19.3 Managing Group Object Privileges

Object privileges control which database objects a user can access and the actions they can perform. There are seven object privileges: SELECT, INSERT, DELETE, UPDATE, INDEX, ALTER, and REFERENCE.

SELECT privilege allows you to select the data in a database object. This privilege applies to the entire object and cannot be granted on specific columns.

INSERT privilege allows you to insert new data into a database object. This privilege can also be restricted to specific columns.

DELETE privilege allows you to delete data from a database object. This privilege applies to the entire object and cannot be granted on specific columns.

UPDATE privilege allows you to update data in a database object. This privilege can also be restricted to specific columns.

INDEX privilege allows you to create an index on a database object. This privilege applies to the entire object and cannot be granted on specific columns.

ALTER privilege allows you to alter the schema of a database object. This privilege applies to the entire object and cannot be granted on specific columns.

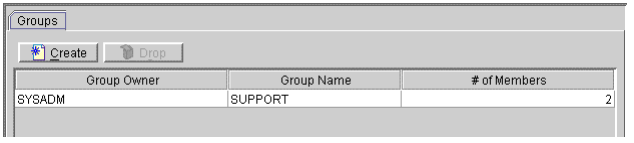
REFERENCE privilege allows you to create referential constraints (such as foreign keys) on a database object. This privilege can also be restricted to specific columns.

EXECUTE privilege allows the user to use a stored procedure or command.

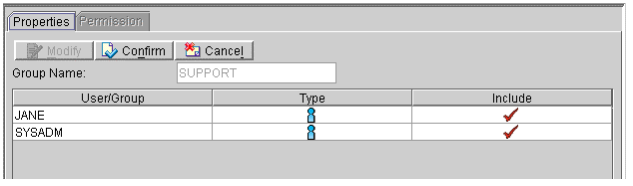
It is possible to grant object privileges to all users simultaneously by granting the privileges to PUBLIC. All current and future users will then have those privileges on the database object.

➲ **To Grant or Remove Group Object Privileges:**

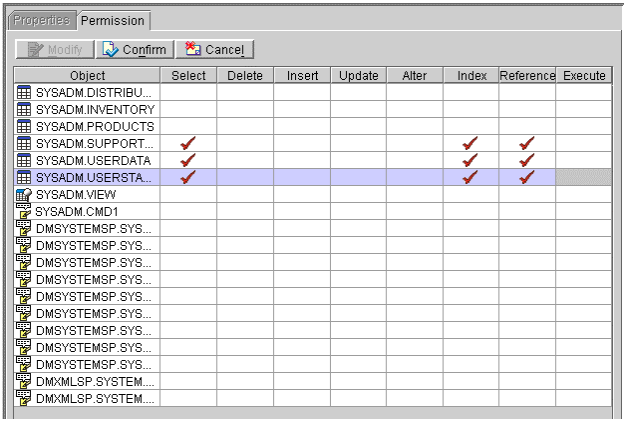
1. Select the **Group** node from the object tree. The **Groups** window will appear displaying all existing groups, the group owner, and the number of members in the group.



2. Double click the group name in the tree or in the group table. The **Properties** page will be displayed.



3. Select the **Permission** tab at the top of the page. The **Permission** page will appear.



4. Click **Modify** to alter the user's object privilege. Choose which type of privilege to grant to the user on each object. Click the column and row corresponding to the object privilege to be granted and the object to grant the privilege to. Privileges can be likewise removed. A check mark will appear for each privilege selected.

5. Click **Confirm** to save the changes, or **Cancel** to discard them.

20 Data Transfer

The Data Transfer Tool provides an user-friendly interface for transferring data in and out of the database. The tool performs the following types of data transformation:

Import from text

Import from XML file

Import from ODBC

Export to text

Export to XML

Batch transfer

Each type of data transformation is performed through a wizard. Each wizard guides the user through every step in the data transformation process, and gives descriptive information on the purpose of each step and the effect of different choices on the result.

The Data Transfer Tool is a separate application it may be opened from windows start>programs>DBMaker 4.1>DataTransfer, or opened from within JDBA Tool. It consists of a main console and a menu bar, as illustrated in *Figure 20-1*. The main console provides five options: import from text, import from XML, import from ODBC, export to text, and export to XML. The Menu Bar consists of three menus: the Transfer menu, the Option menu, and the Help menu. The Transfer menu provides the same transfer functions as the main console, with the addition of the batch transfer function. The option menu can be used to change the language that the

UI is displayed in; currently English, Japanese, and Chinese (traditional) are the supported languages. The help menu provides access to the help system for JDBA Tool.

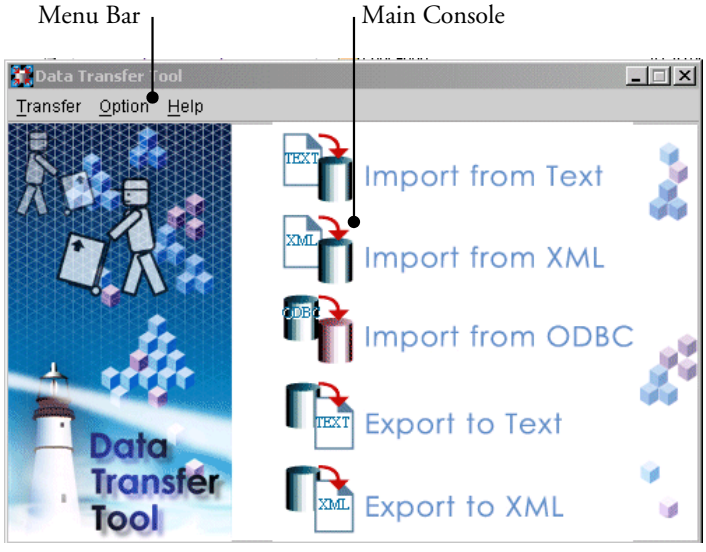


Figure 20-1 The Data Transfer Tool main window

➲ **To open the Data Transfer Tool:**

1. Start JDBA Tool and connect to the database that data is to be transferred to or from.
2. Select **Data Transfer** from the Tool menu. The Data Transfer tool window will open.

20.1 Importing Data from Text

The ability to import table data from a text file is an important feature in a database, and is made easy with the Data Transfer Tool. Text data must be properly formatted to be acceptable for import. Data may be imported to the database only from a properly formatted text file. This section describes the types of formatting that Data Transfer Tool supports, and then describes how to import text to the database using the tool.

Before attempting to import data from a text file, you should be sure that the file is in a format that will result in coherent structured data within the database. Certain programs output data in a fixed text format, if this is the case, check the format of the output file that you want to import. Some important settings to consider in the format of a text file include the following:

Row Delimiter: Determines the type of character that signifies a break between the rows of a table. Possible characters: {CR/LF} (Carriage return / line feed. In Windows applications, a new line in the text is normally stored as a pair of CR LF characters. In Unix applications, a new line is normally stored as a LF character. Some applications use only a CR character to store a new line), {CR}, {semicolon} (;), {comma} (,), {tab}, {vertical bar} (|), (semicolon}{LF}, or {comma}{LF}.

Column Delimiter: Determines the type of character that signifies a break between columns in each row. Possible characters: semicolon, comma, or vertical bar.

Text Qualifier: Determines how each tuple of any data type except BINARY, LONGVARBINARY, or numeric data types (integer, smallint, serial, decimal, double, float) is enclosed. Possible values: none, single quote, or double quote.

Binary Qualifier: Determines how each tuple of BINARY or LONGVARBINARY type data is enclosed. Possible values: none, single quote, or double quote.

Binary Padding: Binary type data may have a character appended to it.

Fixed Field: Instead of using a row delimiter, the text file may be formatted with fixed fields. This means a number of spaces, or fields, defines each column.

Include column name: The first line in a text file may be used to define the column names. The format is *"column1"."table name"."owner name";"column2"."table name"."owner name";* etc. In this case the column delimiter is set to semicolon (;).

Include table schema: The first line in a text file may be used to define the column schema (or the second line if the first line was used to define column names). The format is *data type(scale,precision);data type(scale,precision)* etc. In this case, the column delimiter is set to semicolon (;).

Use NULL to display null data: Columns that contain no data display "NULL".

File link name for FILE type data: The file name for system or user file objects is displayed.

Use escape character "/": This character is used when qualifiers or delimiter characters appear in the data. If the data contains a reserved character, the reserved character will be enclosed by an escape character (/) so that the text import engine recognizes it as data, not a qualifier or delimiter.

Use temp files to store LONGVARBINARY or LONGVARCHAR type data column content: BLOB data is stored as a separate, linked file (as a file object), and the name of the file containing the BLOB is displayed.

Data may be imported to a new table or an existing table. Three options are available when importing data to an existing table. The destination table may be replaced, removing the schema and data of the original table. The rows of the destination table may be replaced, meaning schema is retained, but data is removed. The last option is to append rows to the destination table, in which case the table's original schema and data is preserved, and the data from the text file is appended as new rows. In the last two cases, be aware that the schema of the destination table must be such that it is able

to accept the imported data. For example, a set of integer data can be imported into a column of CHAR type data, but not vice-versa. Refer to the *SQL Command and Function Reference* for more information on acceptable formats for supported data types in DBMaker.

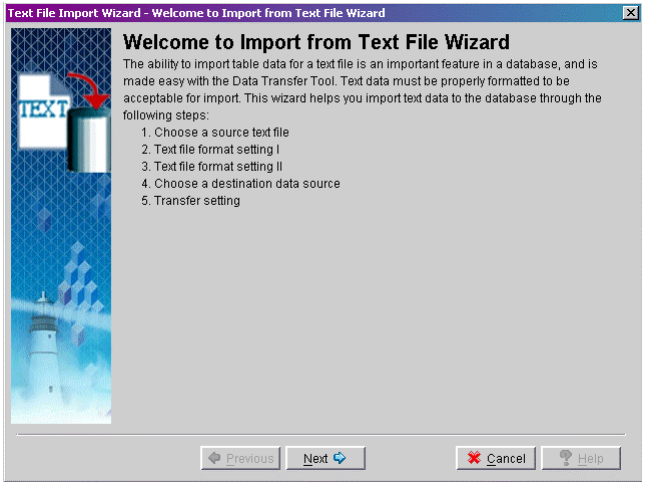
➲ **Example**

The following text file contains owner, table, and column information in the first line, and the table's schema in the second line. Subsequent lines contain raw data. Semicolons delimit columns. Single quotes qualify non-binary data; double quotes qualify binary data. The LONGVARCHAR and LONGVARBINARY columns display temporary file names (the data has been stored as file objects), and binary data is padded with "9".

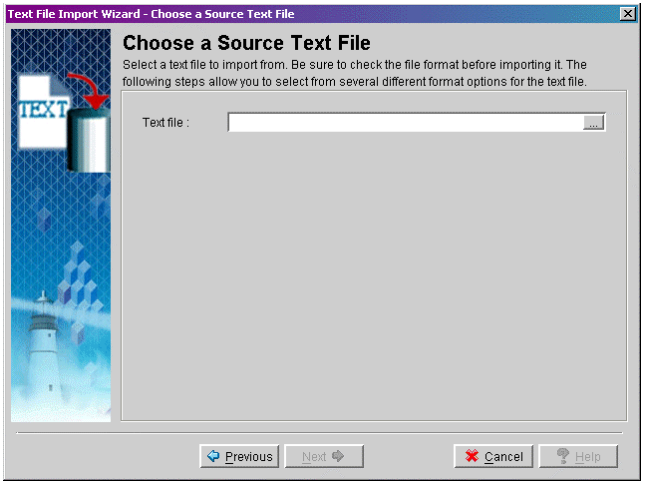
```
"SYSADM"."EXPORT"."LOGINID";"SYSADM"."EXPORT"."REQUEST";"SYSADM"."EXPORT"."REQUES
TTIME";"SYSADM"."EXPORT"."ATTACHMENT";"SYSADM"."EXPORT"."BINARY_C";"SYSADM"."EXPO
RT"."DECIMAL_C"
SQL_CHAR(20);SQL_LONGVARCHAR;SQL_TIMESTAMP;SQL_FILE;SQL_BINARY(10);SQL_DECIMAL(10
, 3)
'A_HOWARD            ';'blobtmpdir2\\blbtmpf0.txt';'2001-09-09
12:47:05.000';'C:\\DBMAKER\\4.1\\BIN\\WEBDB\\FO\\ZZ00000B.GIF';"10000000000000000
000"9;10.250
'A_HOWARD            ';'blobtmpdir2\\blbtmpf1.txt';'2001-09-22
10:14:21.000';'C:\\DBMAKER\\4.1\\BIN\\WEBDB\\FO\\ZZ00000C.GIF';"20000000000000000
000"9;13.550
'A_HOWARD            ';'blobtmpdir2\\blbtmpf2.txt';'2001-10-04
16:22:06.000';'C:\\DBMAKER\\4.1\\BIN\\WEBDB\\FO\\ZZ00000D.GIF';"30000000000000000
000"9;27.333
```

➲ **To import a text file to a database**

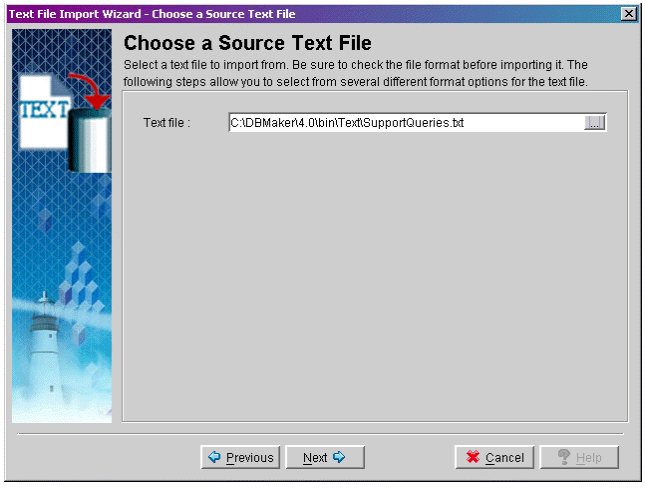
1. Open the Data Transfer Tool.
2. Select **Import Text File** from the main console or the **Transfer** menu. The Welcome to Import from Text File Wizard window will open, displaying a summary of the steps to be taken in the wizard



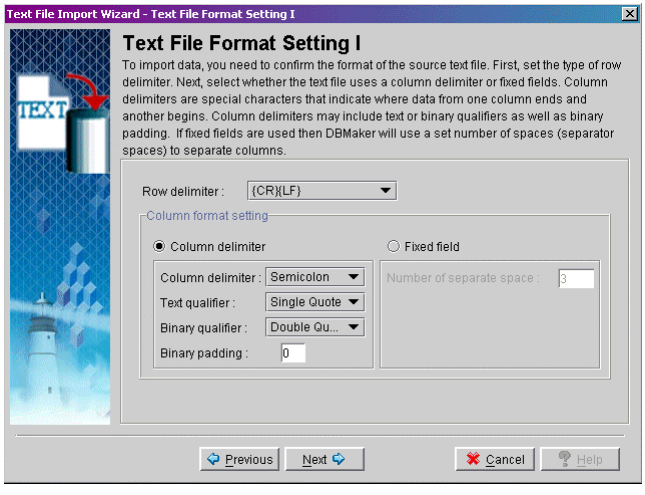
3. Click **Next**. The **Choose a Source Text File** window will open.



4. Enter the full path of a text file to import or click the browse button to search for a text file.

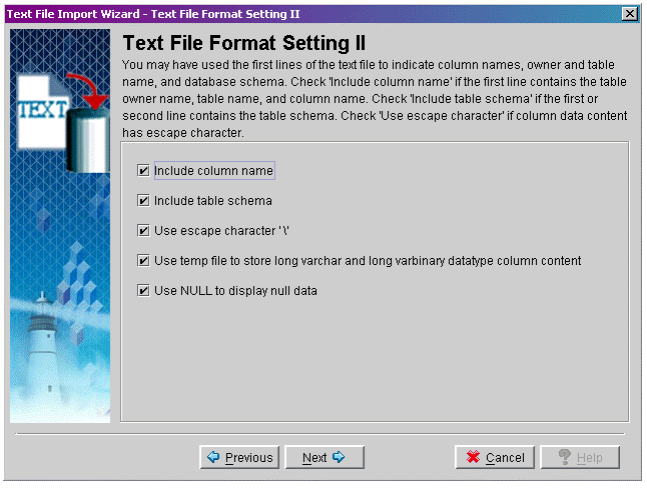


5. After you have selected a text file, click **Next**, the **Text File Format Setting 1** window will open.



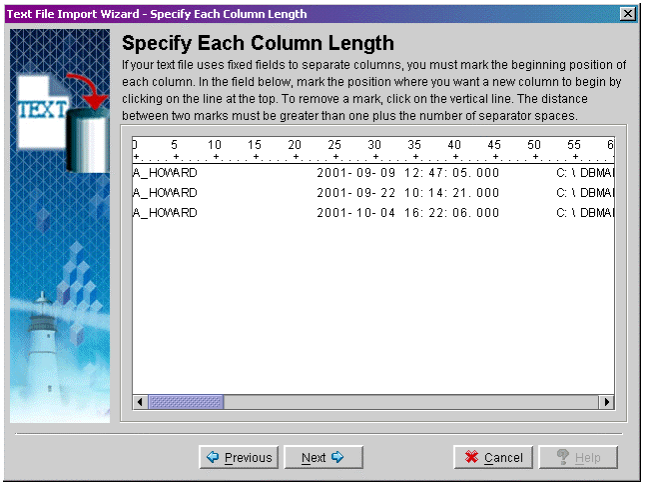
6. Open the text file in a text editor to check the format of the data.
7. Select the appropriate settings for the format of the text file you are importing.

8. Click **Next**, the **Text File Format Setting 2** window will open.

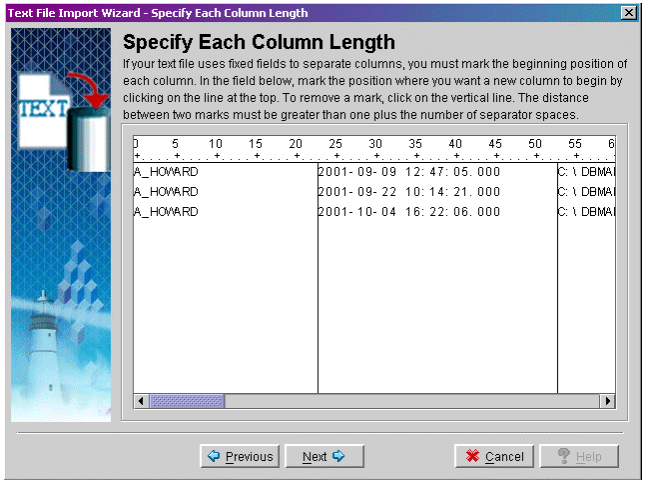


9. Finish selecting the appropriate settings for the format of the text file you are importing. Click **Next**.

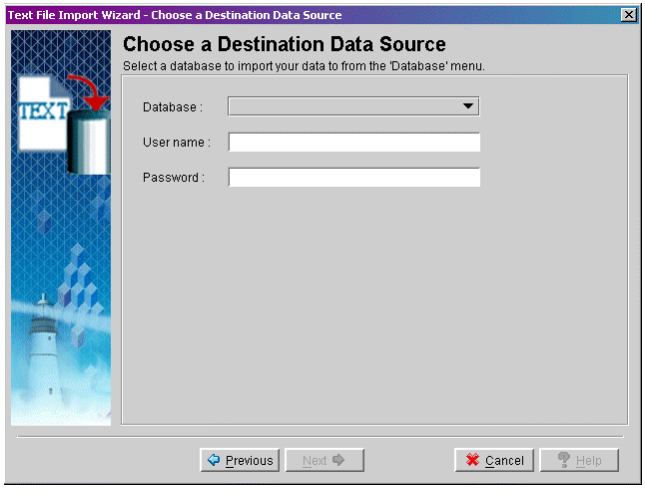
10. If you selected **Fixed field** from the **Text File Format Setting 1** window, the **Specify Column Length** window will open. If you selected **Column Delimiter**, then proceed to step 12.



11. Click the horizontal line where you want to indicate the beginning of a column. A vertical line will appear, marking the column break. Use the scroll bar at the bottom to advance to other columns. Click **Next**.

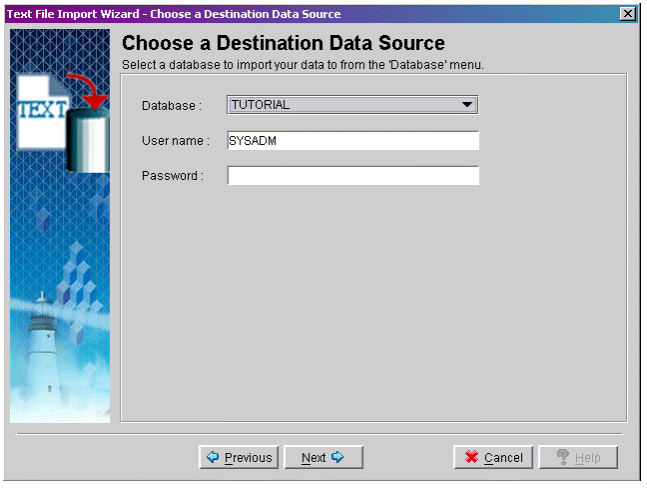


12. The **Choose Destination Data Source** window will open.



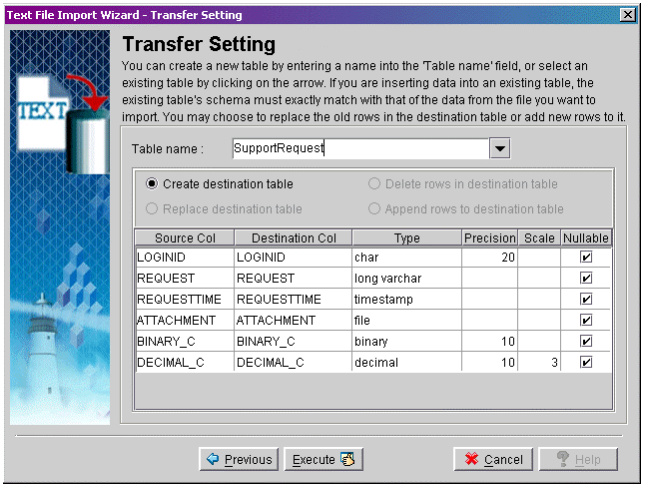
13. Select the database to import data to from the **Database** menu.

14. Enter a user name and password into the appropriate fields.



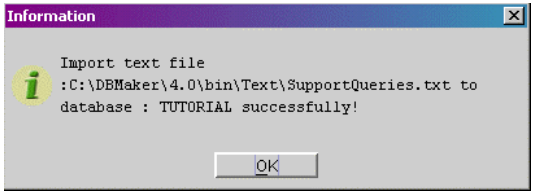
NOTE *A user must have INSERT privilege to import a text file.*

15. Click **Next**. The **Transfer Setting** window will open.



16. Enter a new table name into the **Table Name** field, or select a table from the menu. Selecting a table from the menu will allow you to choose to replace the destination table, delete rows in the destination table, or append new rows to

the destination table. Click **Execute** to import the text file. A confirmation dialog box will appear.



17. Click **OK**

20.2 Importing Data from XML

XML files may also be imported into the database. XML tags may first be defined in a *Document Type Definition* (DTD) file before being imported into the database. Furthermore, the DTD may define the schema in a way that is acceptable to the database.

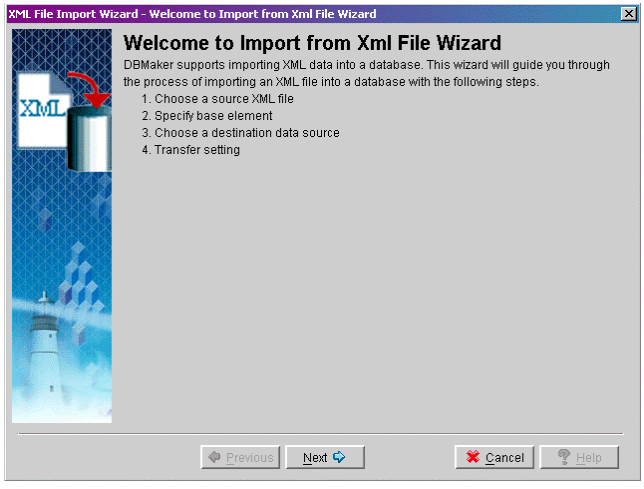
It is important to consider the structure of the XML file you wish to import. To ensure that the structure of the XML file and associated DTD have compatible structure, examine the structure of XML files produced by the Data Transfer Tool: Export to XML File wizard. Examples may be found in section 20.5, *Exporting Data to XML*. Files produced using the Export to XML wizard always can be imported, however, the extent to which a table's schema is reproduced varies. The setting that influences table schema the most is the Column as Element / Attribute setting.

> Column as Element: Stores data items in elements. If table schema information exists as element attributes (data type, column name, length, etc.) in the DTD, then columns will be created with names and of the appropriate data type and length. Columns are child elements, and the table is represented as the parent element. File objects must be referenced as entities in the DTD file if Column as Element is chosen.
>
> Column as Attribute: Stores data in an attribute of an element. Each element is a record. If column names are represented as attributes of the root element (the table) in the DTD, and each tag in the XML file represents one record, then Column as Attribute should be chosen.

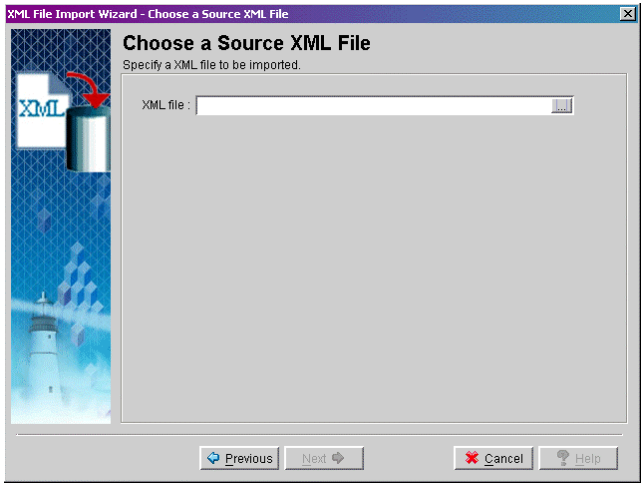
➲ **To import data from an XML file:**

1. Open the Data Transfer Tool.

2. Select **Import XML File** from the main console or the **Transfer** menu. The **Welcome to Import from XML File Wizard** window will open, displaying a summary of the steps to be taken in the wizard.

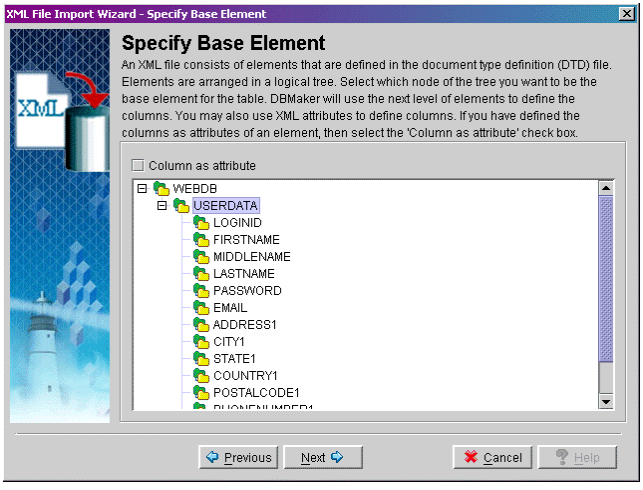


3. Click **Next**. The **Choose a Source XML File** window will open.



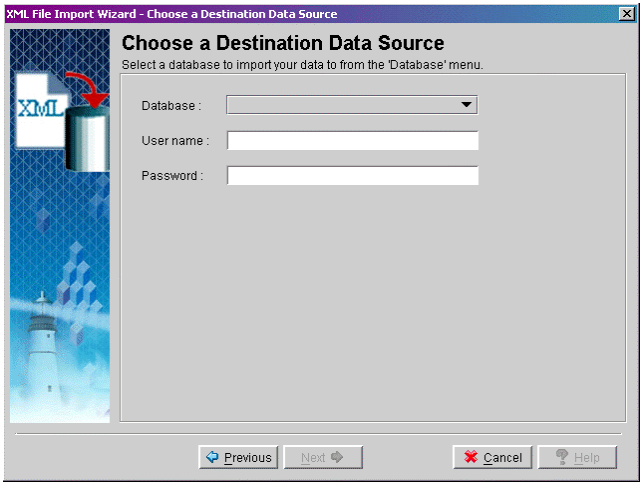
4. Enter the full path of a text file to import or click the browse button to search for a text file.

5. Click **Next**. If the XML file has a structure acceptable to DBMaker's parser, the **Specify Base Element** window will open.



6. The nodes of the tree structure represent the elements in the XML file. Click the nodes on the tree until they are fully expanded. Select a parent element to be the table name. The child elements will become the columns of the table. Check **Column as attribute** if appropriate.

7. Click **Next**. The **Choose a Destination Data Source** window will open.

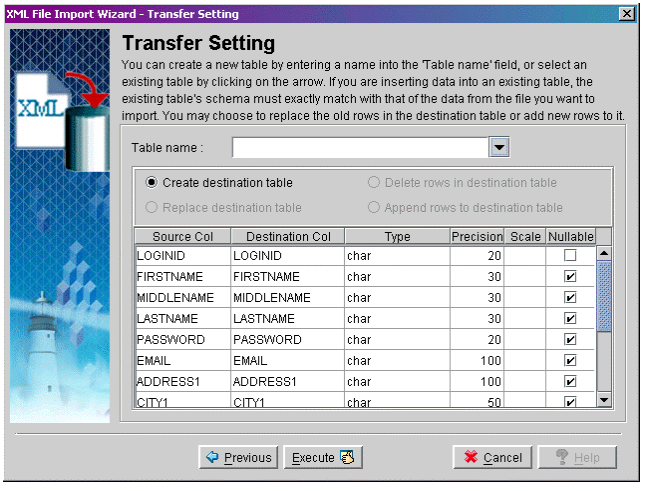


8. Select the database to import data to from the **Database** menu.

9. Enter a user name and password into the appropriate fields.

NOTE *DBA authority or higher is required to import a text file.*

10. Click **Next**. The **Transfer Setting** window will open.



11. Enter a new table name into the **Table Name** field, or select a table from the menu. Selecting a table from the menu will allow you to choose to replace the destination table, delete rows in the destination table, or append new rows to the destination table. Click **Execute** to import the XML file. A confirmation dialog box will appear.

12. Click **OK**.

20.3 Importing Data from ODBC

A large number of software developers have developed applications to be Open Database Connectivity (ODBC) compatible. ODBC is an industry standard for sharing data between diverse data sources. DBMaker can import data from any ODBC compliant data source through the Import from ODBC wizard.

Data may be imported by three methods:

Directly from tables.

Writing one or more SQL SELECT statements.

Importing through an XML batch file.

Furthermore, you may specify the mapping of column data through the transformation function. The transformation function supports direct column-to-column mapping or mapping through SQL SELECT and SQL INSERT statements.

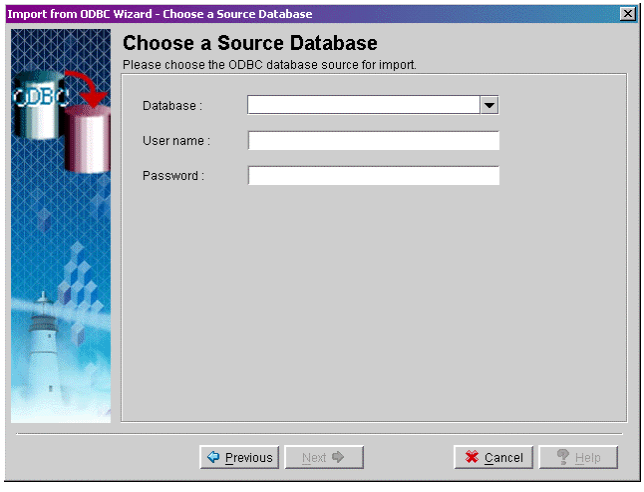
When importing data directly from tables or through SQL SELECT statements it is possible to save a 'map' of the data transformation to an XML batch file. The saved XML batch file is a well-formed XML document with a form that can be parsed by the data transfer tool. Batch files may be used to import table schema from a data source to multiple DBMaker databases.

➲ **To import data from an ODBC database:**

1. Open the Data Transfer Tool.

2. Select **Import XML File** from the main console or the **Transfer** menu. The **Welcome to Import from XML File Wizard** window will open, displaying a summary of the steps to be taken in the wizard.

3. Click **Next**. The **Choose a Source Database** window will open.

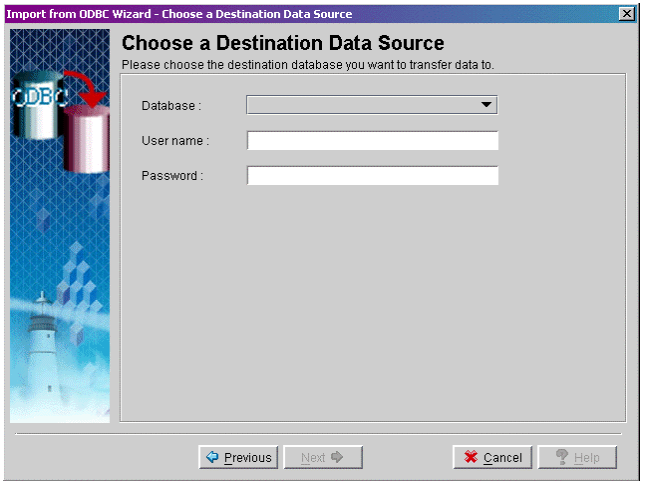


4. Select the database to export data from in the **Database** menu.

5. Enter a user name and password into the appropriate fields.

NOTE *DBA authority or higher is required to export a text file.*

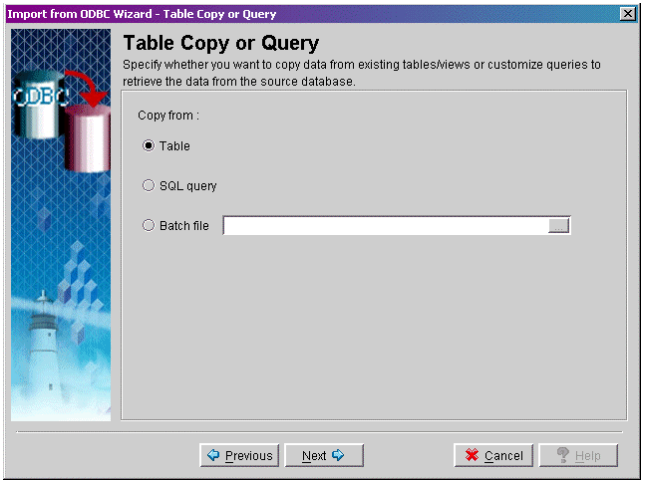
6. Click **Next**. The **Choose a Destination Data Source** window will open.



7. Select the database to import data to from the **Database** menu.
8. Enter a user name and password into the appropriate fields.

NOTE *DBA authority or higher is required to import a text file.*

9. Click **Next**, the **Table Copy or Query** window will open.



10. Select from one of the three methods for data transfer:

To import data from a list of tables, select **Table**.

To import data using a series of SQL SELECT statements, select **SQL query**.

To import data through an XML file, select **Batch file**

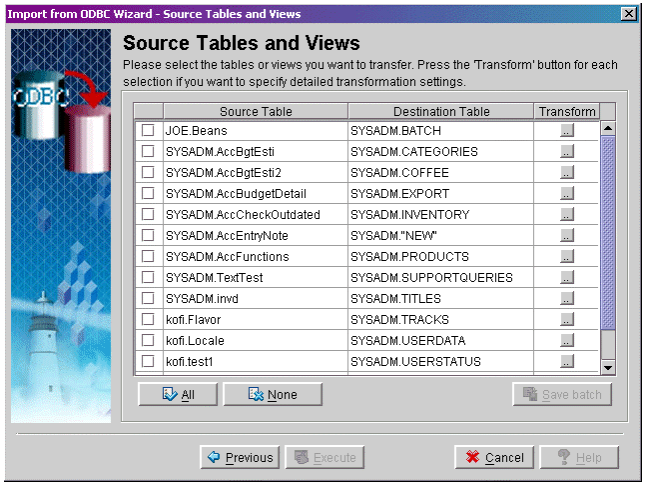
Importing ODBC Data from Tables

ODBC data may be imported from variety of sources by selecting tables directly from the source database. The Import from ODBC Wizard provides an intuitive graphical interface for selecting tables and setting how data should be transformed during the import process.

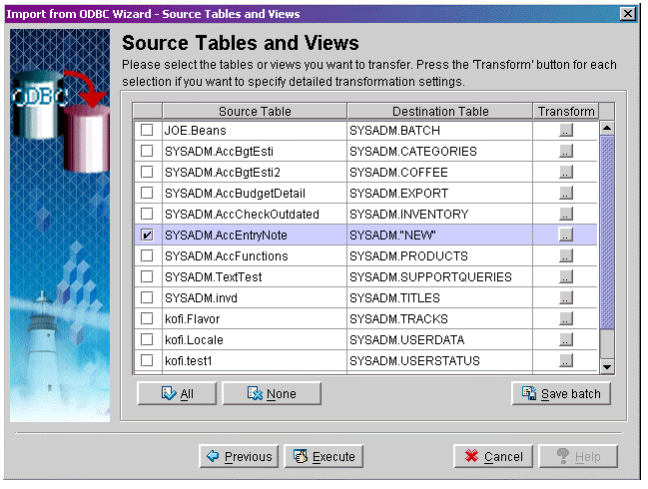
➲ **To import ODBC data from a list of tables**

1. In the Import from ODBC wizard, select a source and destination database, and then select **Table** from the **Table Copy or Query** window.

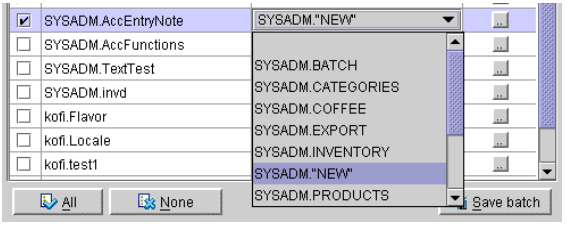
2. Click **Next**. The **Source Tables and Views** window will open.



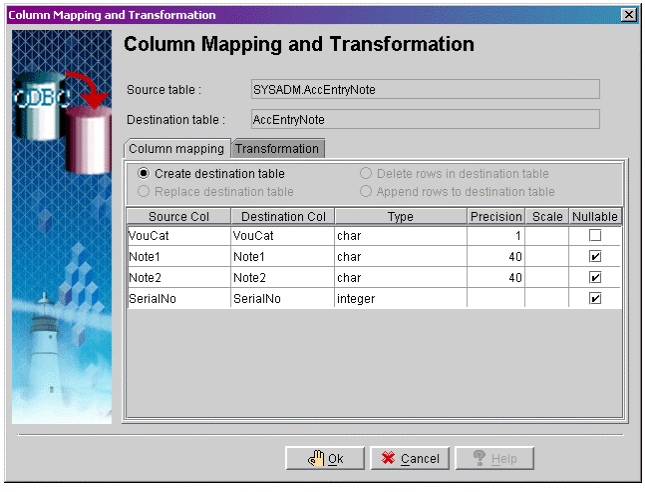
3. All tables from the source database will appear in the **Source Table** column. Check the box to the left of each table to import.



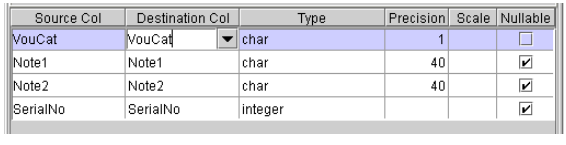
4. For each source table or view selected, click the **Destination Table** field. If desired, change the name of the destination table by selecting a new table from the menu or entering a new name.



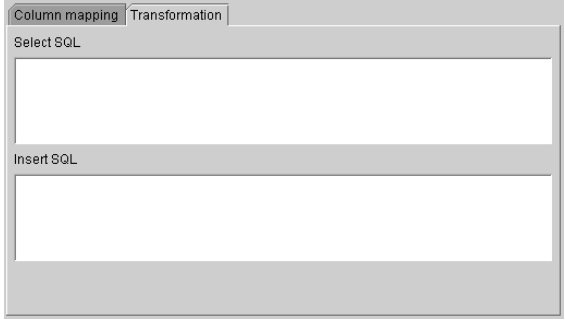
5. You may modify column mapping or the result set to import by clicking on the **Transform** button of the corresponding source and destination table.



6. Change the name of the destination column by selecting a new column from the menu or entering a new name.

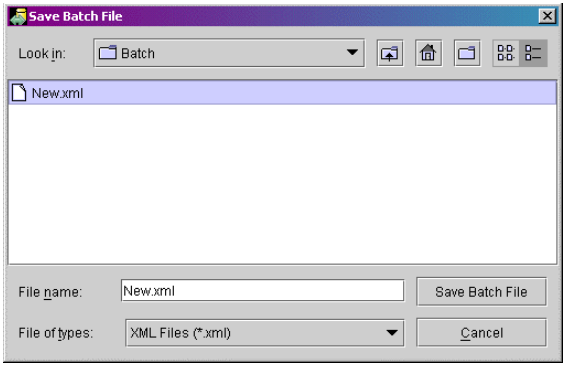


7. Click the **Transformation** tab to specify constraints on the result set. Enter a valid SQL SELECT statement into the **Select SQL** field, and a valid SQL INSERT statement into the **Insert SQL** field.



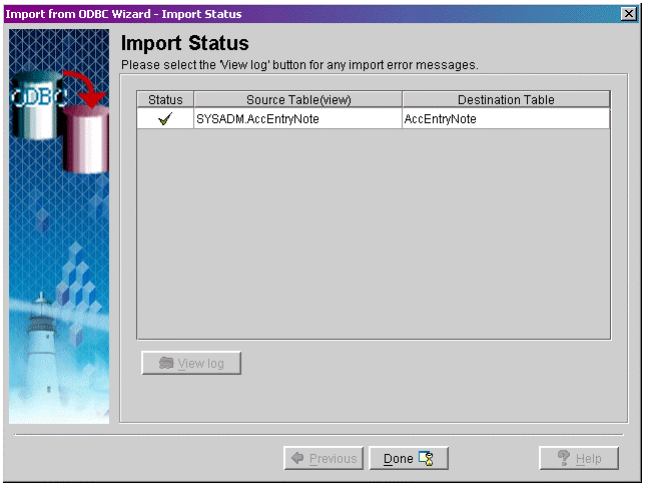
8. Click **OK** to return to the **Source Tables and Views** window.

9. You may also choose to save the map of the import ODBC schema to an XML file by clicking **Save batch**. The **Save Batch File** will open.



10. Select or create an XML file to save the imported ODBC map schema to. Click **Save Batch File** to create the XML file. The **Source Tables and Views** window will reappear.

11. Click **Execute** to import the source data. The Import Status window will appear.



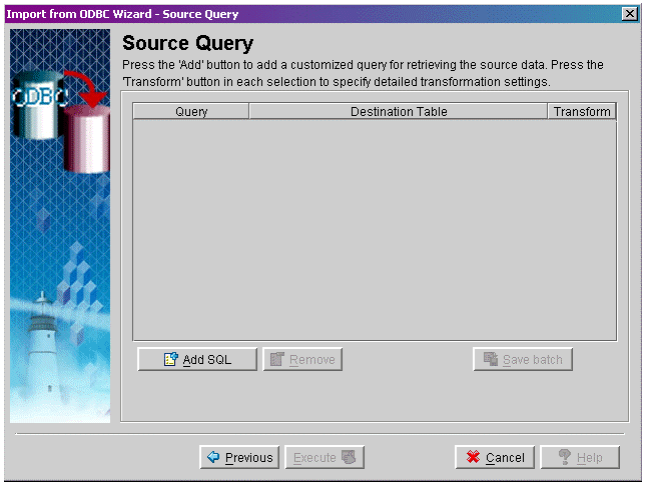
12. If errors appear, click **View log** and scroll to the bottom to see the error message. If no errors occurred, click **Done**.

Importing ODBC Data Using SQL SELECT Statements

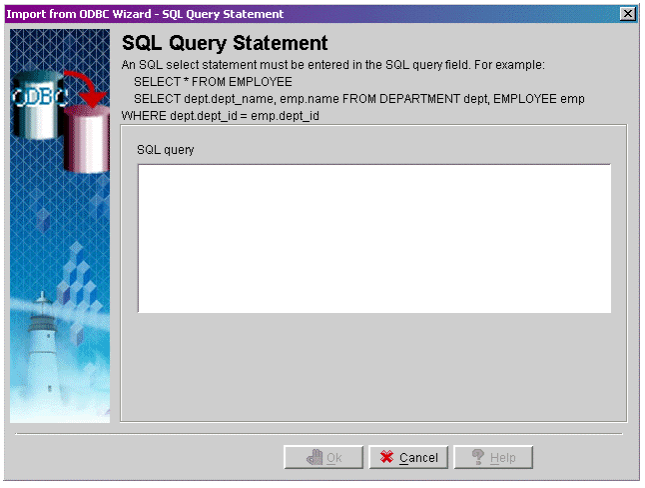
Data may also be imported from ODBC sources by creating a series of SQL SELECT statements. If you have knowledge of the schema of tables you want to import, this may be a faster option.

➲ **To import ODBC data using a series of SQL SELECT statements**

1. In the Import from ODBC wizard, select a source and destination database, and then select **Table** from the **Table Copy or Query** window.
2. Click **Next**. The **Source Query** window will open.

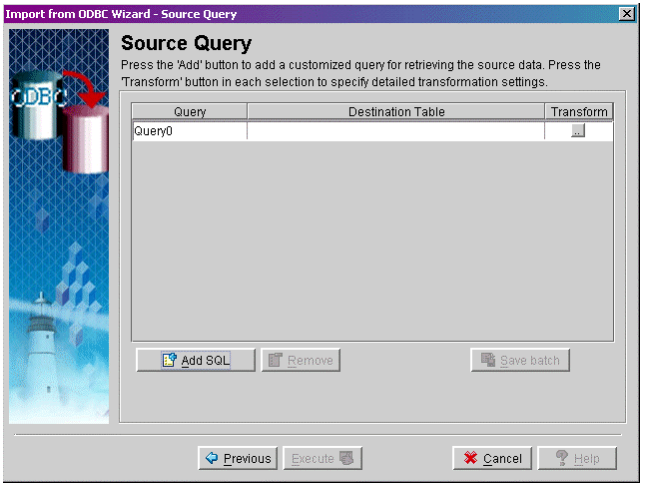


3. Click **Add SQL**. The **SQL Query Statement** window will open.

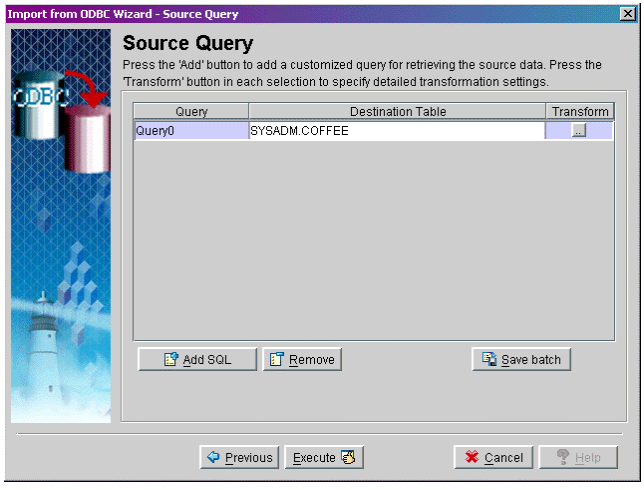


4. Enter a valid SQL SELECT statement into the **SQL Query** field.

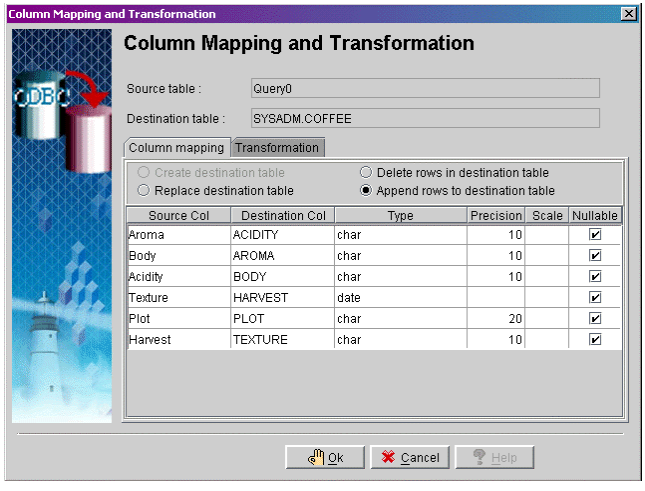
5. Click **OK**. The **Source Query** window will reappear.



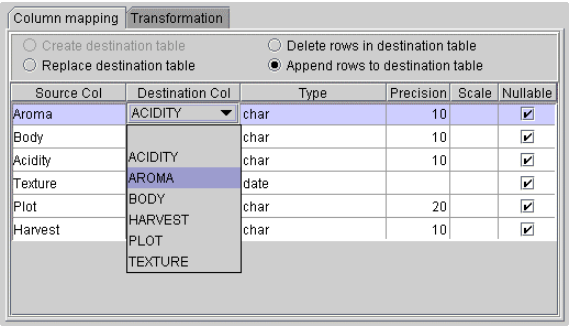
6. Select or create a destination table from the **Destination Table** column.



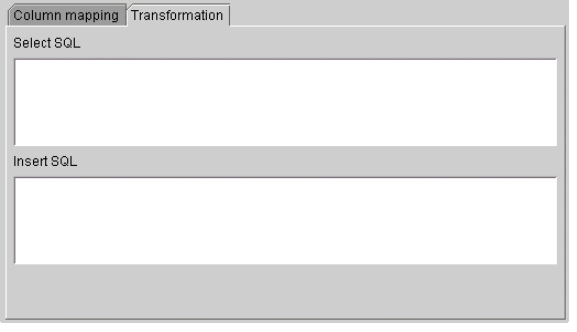
7. You may add more SQL query statements by clicking **Add SQL**, or modify the mapping of source and destination columns by clicking the **Transform** button.



8. Change the name of the destination column by selecting a new column from the menu or entering a new name.

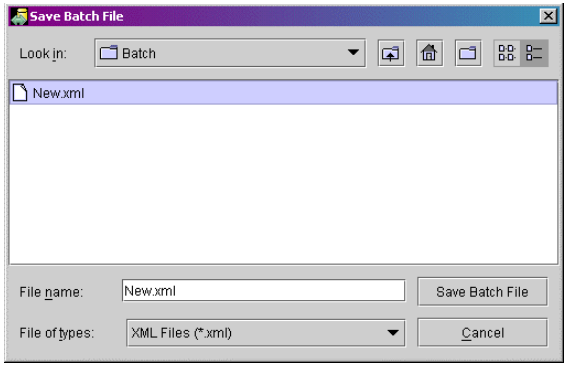


9. Click the **Transformation** tab to specify constraints on the result set. Enter a valid SQL SELECT statement into the **Select SQL** field, and a valid SQL INSERT statement into the **Insert SQL** field.



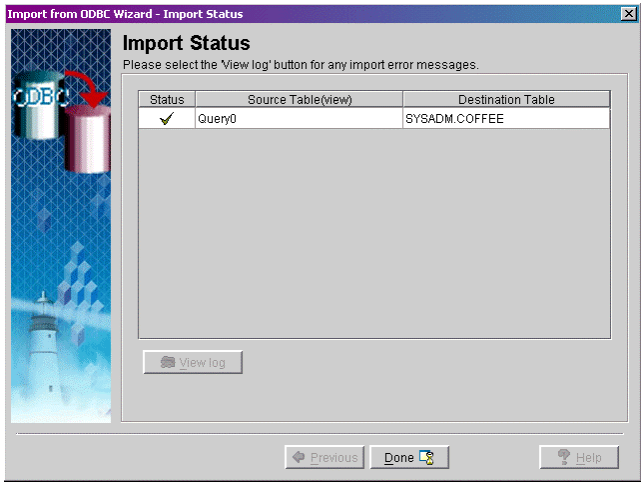
10. Click **OK** to return to the **Source Tables and Views** window.

11. You may also choose to save the map of the imported ODBC schema to an XML file by clicking **Save batch**. The **Save Batch File** will open.



12. Select or create an XML file to save the import ODBC map schema to. Click **Save Batch File** to create the XML file. The **Source Tables and Views** window will reappear.

13. Click **Execute** to import the source data. The Import Status window will appear.



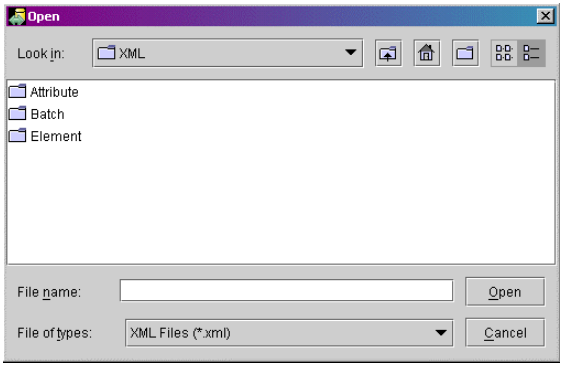
14. If errors appear, click **View log** and scroll to the bottom to see the error message. If no errors occurred, click **Done**.

Importing ODBC Data Through an XML Batch File

It is also possible to use an XML Batch file to specify which tables are to be imported. Users have the option to create an XML batch file when importing ODBC data from tables or with SQL Select statements. Batch files may be used to import table schema from a data source to multiple DBMaker databases

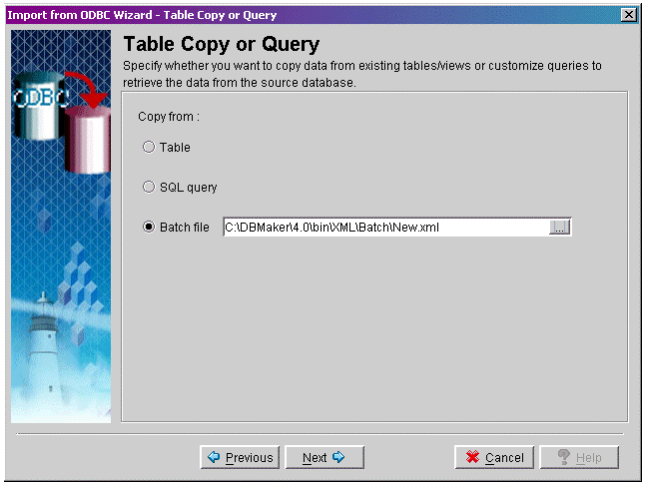
➲ **To import data through an XML batch file**

1. In the Import from ODBC wizard, select a source and destination database, and then select **Batch file** from the **Table Copy or Query** window. The **Open** window will open.

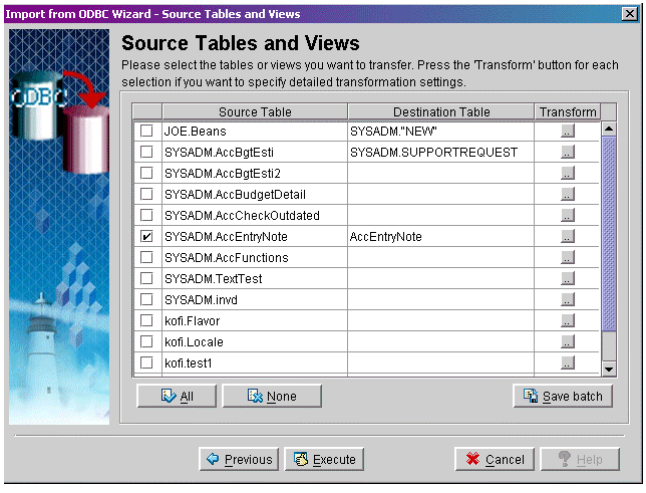


2. Select an XML file from which to import the ODBC map schema.

3. Click **Open**. The **Table Copy or Query** window will reappear.

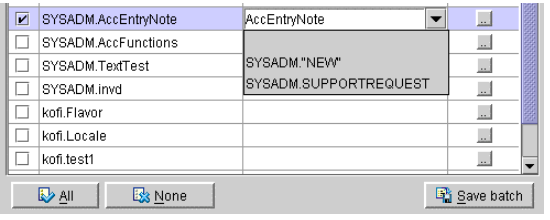


4. Click **Next**. The **Source Tables and Views** window will open, displaying a mapping schema according to the XML file.

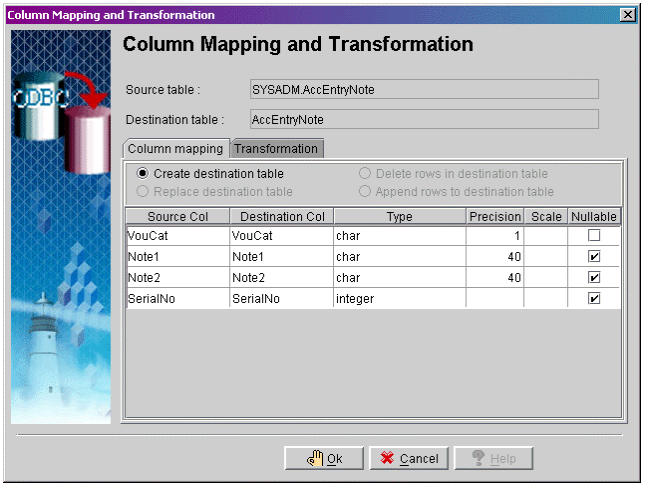


5. All tables from the source database will appear in the **Source Table** column. Check the box to the left of additional tables to import.

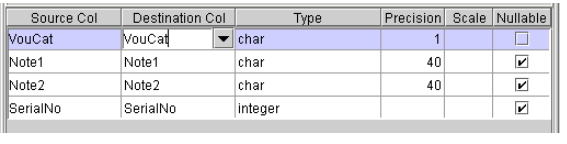
6. For each source table or view selected, click the **Destination Table** field. If desired, change the name of the destination table by selecting a new table from the menu or entering a new name.



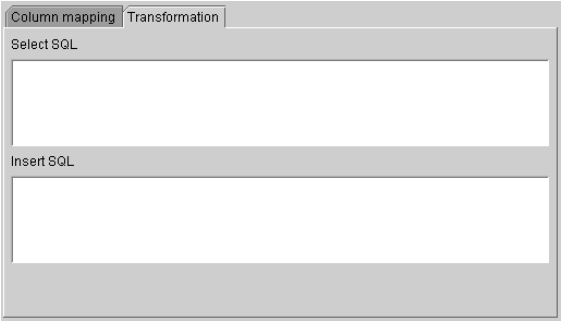
7. You may modify column mapping or the result set to import by clicking on the **Transform** button of the corresponding source and destination table.



8. Change the name of the destination column by selecting a new column from the menu or entering a new name.

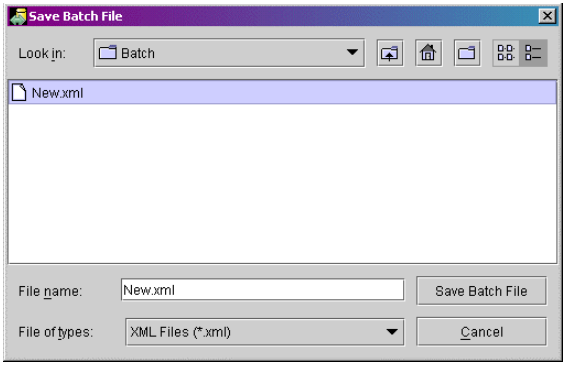


9. Click the **Transformation** tab to specify constraints on the result set. Enter a valid SQL SELECT statement into the **Select SQL** field, and a valid SQL INSERT statement into the **Insert SQL** field.



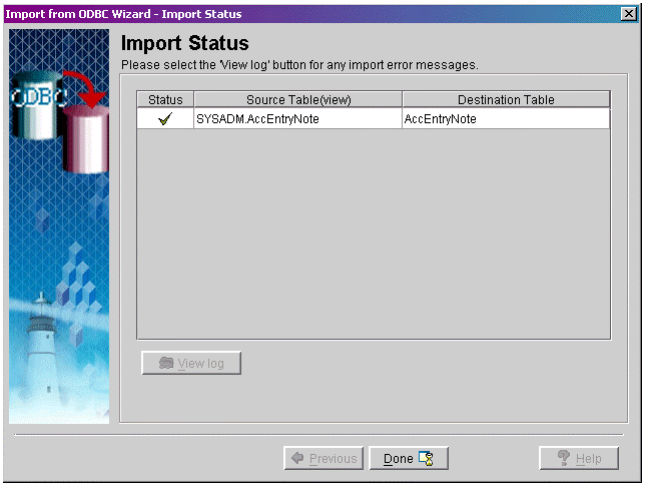
10. Click **OK** to return to the **Source Tables and Views** window.

11. You may also choose to save the map of the import ODBC schema to an XML file by clicking **Save batch**. The **Save Batch File** will open.



12. Select or create an XML file to save the imported ODBC map schema to. Click **Save Batch File** to create the XML file. The **Source Tables and Views** window will reappear.

13. Click **Execute** to import the source data. The Import Status window will appear.



14. If errors appear, click **View log** and scroll to the bottom to see the error message. If no errors occurred, click **Done**.

20.4 Exporting Data to Text

Data may be exported from the database to form a structured text file. This section describes the different text file formats that the Data Transfer tool can produce. Consider the following settings affecting the format of the text file before exporting data.

Row Delimiter: Determines the type of character that signifies a break between the rows of a table. Possible characters: {CR/LF} (Carriage return / line feed. In Windows applications, a new line in the text is normally stored as a pair of CR LF characters. In Unix applications, a new line is normally stored as a LF character. Some applications use only a CR character to store a new line), {CR}, {semicolon} (;), {comma} (,),{tab}, {vertical bar} (|), (semicolon}{LF}, or {comma}{LF}.

Column Delimiter: Determines the type of character that signifies a break between columns in each row. Possible characters: semicolon, comma, or vertical bar.

Text Qualifier: Determines how each tuple of any data type except BINARY, LONGVARBINARY, or numeric data types (integer, smallint, serial, decimal, double, float) is enclosed. Possible values: none, single quote, or double quote.

Binary Qualifier: Determines how each tuple of BINARY or LONGVARBINARY type data is enclosed. Possible values: none, single quote, or double quote.

Binary Padding: A character appended to each tuple containing binary data.

Fixed Field: Instead of using a row delimiter, the text file may be formatted with fixed fields. This means a number of spaces, or fields, defines each column.

Include column name: The first line in a text file may be used to define the column names. The format is *"column1"."table name"."owner*

name";"column2"."table name"."owner name"; etc. In this case the column delimiter is set to semicolon (;).

Include table schema: The first line in a text file may be used to define the column schema (or the second line if the first line was used to define column names). The format is *data type(scale,precision);data type(scale,precision)* etc. In this case the column delimiter is set to semicolon (;).

Use NULL to display null data: Columns that contain no data output "NULL".

File link name for FILE type data: The file name for system or user file objects is displayed.

Use escape character "/": This character is used when qualifiers or delimiter characters appear in the data. If the data contains a reserved character, the reserved character will be enclosed by an escape character (/).

Use temp files to store LONGVARBINARY or LONGVARCHAR type data column content: BLOB data is stored as a separate, linked file (as a file object), and the name of the file containing the BLOB is displayed.

Data can be exported by selecting individual columns from a table, or through a valid SQL SELECT statement. For more information on SQL syntax, refer to the *SQL Command and Function Reference*.

➲ **Example**

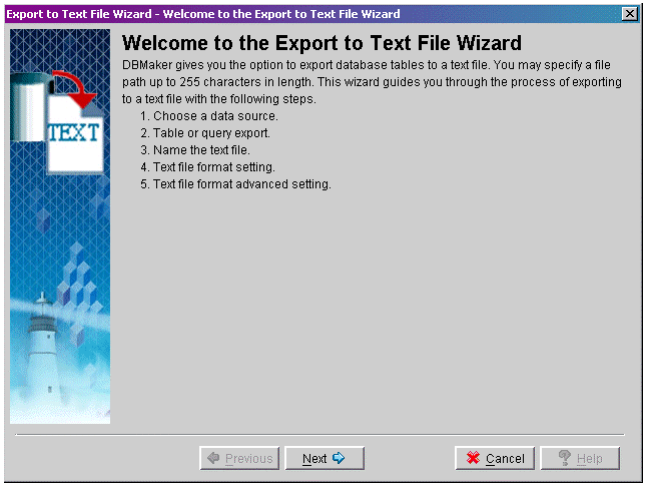
The following is text produced by the Data Transfer Tool Export to Text wizard. The table's name and schema will be displayed on the first two lines. The data has been exported with fixed fields to delimit columns and CR to delimit rows. File link names appear where LONGVARCHAR data once stored as BLOBs in the table have been exported as file objects.

```
"SYSADM"."SUPPORTQUERIES"."LOGINID"   "SYSADM"."SUPPORTQUERIES"."REQUESTTIME"
"SYSADM"."SUPPORTQUERIES"."ATTACHMENT"   "SYSADM"."SUPPORTQUERIES"."DECIMAL_C"
SQL_CHAR(10)  SQL_TIMESTAMP                SQL_LONGVARCHAR    SQL_DECIMAL(10, 3)
A_HOWARD     2001-09-09 12:47:05.000       C:\WEBDB\FO\ZZ000000.GIF       10.250
A_HOWARD     2001-09-22 10:14:21.000       C:\WEBDB\FO\ZZ000001.GIF       13.550
A_HOWARD     2001-10-04 16:22:06.000       C:\WEBDB\FO\ZZ000002.GIF       27.333
```

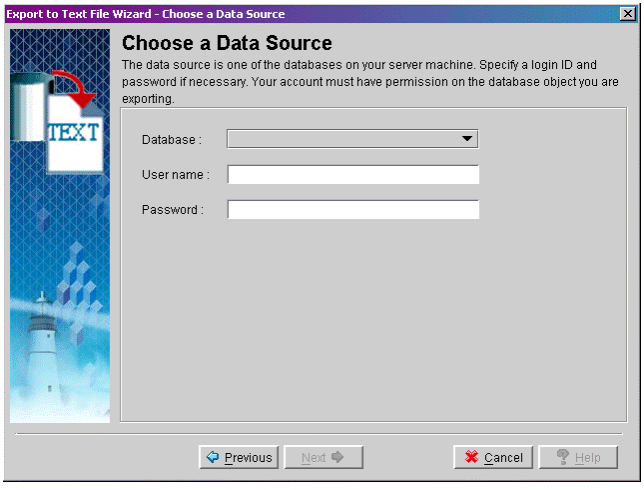
```
A_HOWARD     2001-10-09 17:44:56.000      C:\WEBDB\FO\ZZ000003.GIF      16.140
A_HOWARD     2001-10-12 09:12:38.000      C:\WEBDB\FO\ZZ000004.GIF      88.847
A_HOWARD     2001-10-31 23:16:11.000      C:\WEBDB\FO\ZZ000005.JPG      841.336
```

➲ **To export a table to a text file**

1. Open the Data Transfer Tool.
2. Select **Export Text File** from the main console or the **Transfer** menu. The Welcome to Import from Text File Wizard window will open, displaying a summary of the steps to be taken in the wizard.

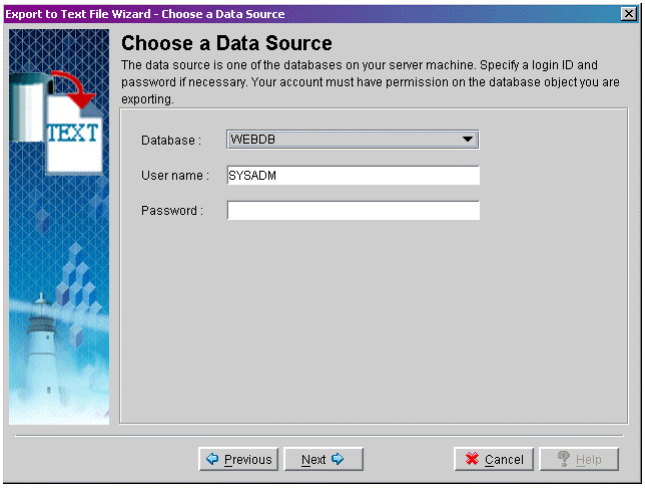


3. Click **Next**. The **Choose a Data Source** window will open.



4. Select the database to export data from in the **Database** menu.
5. Enter a user name and password into the appropriate fields.

NOTE *A user must have SELECT privilege to export a text file.*

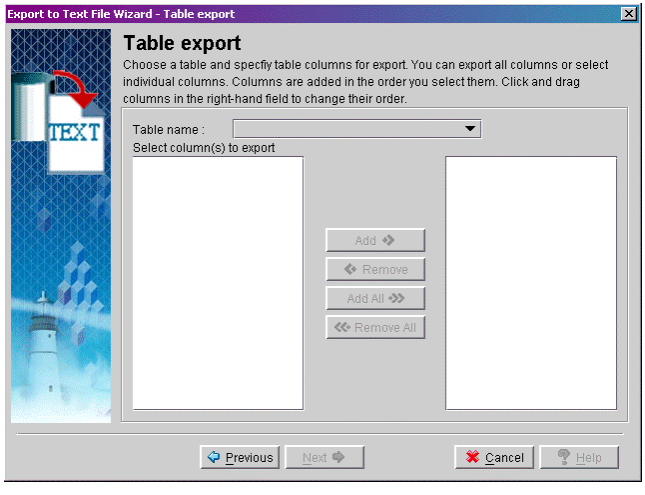


6. After you have selected a database, click **Next**, the **Table or Query Export** window will open.

7. Select **Table** to export data from a table. Select **SQL query** to export data from the result set of an SQL SELECT statement.

8. If you selected **Table** from the **Table or Query Export** window, the **Table Export** window will open. If you selected **SQL query**, then proceed to step 13.

9. Click **Next**, the **Table Export** window will open.



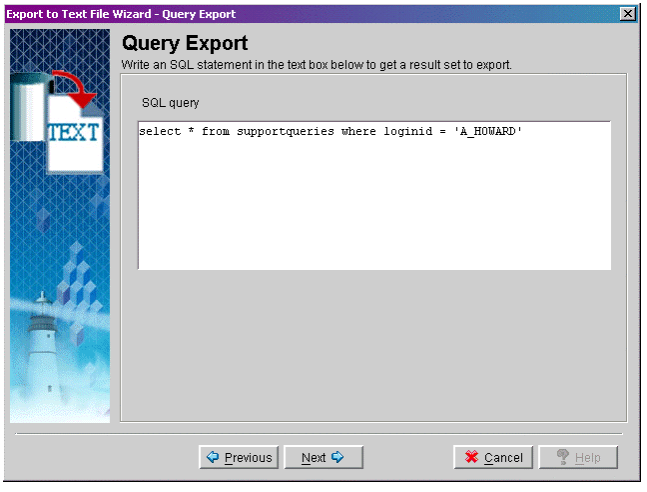
10. Select a table to export from the **Table** name menu. A list of columns in the table will appear in the **Select columns to export** field.

11. Select columns by clicking on the column name and clicking **Add**, or select all columns by clicking **Add All**. Selected column names will appear in the right hand field.

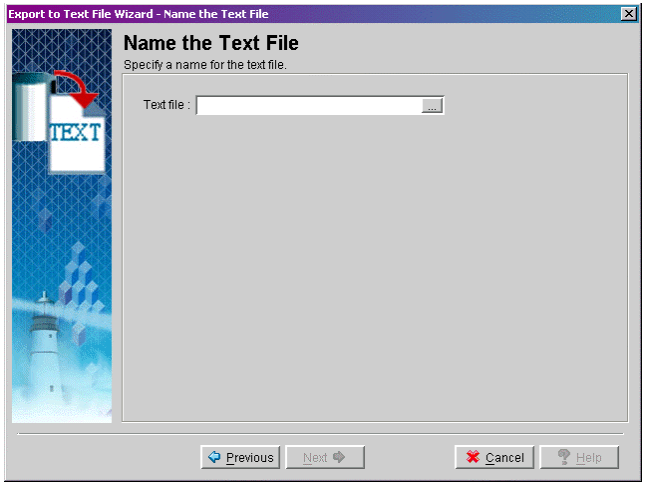
12. Click **Next**. The **Name The Text File** window will appear (proceed to step 16).

13. If you selected **SQL query** from the **Table or Query Export** window, the **Query Export** window will open.

14. Enter a valid SQL select statement into the **SQL query** field.

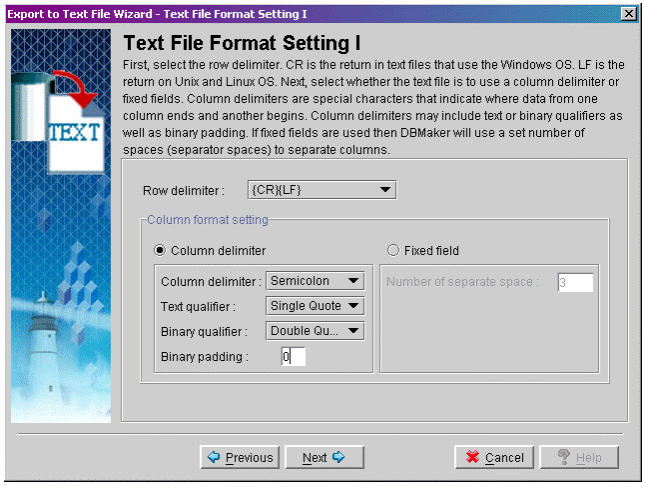


15. Click **Next**. The **Name The Text File** window will appear.



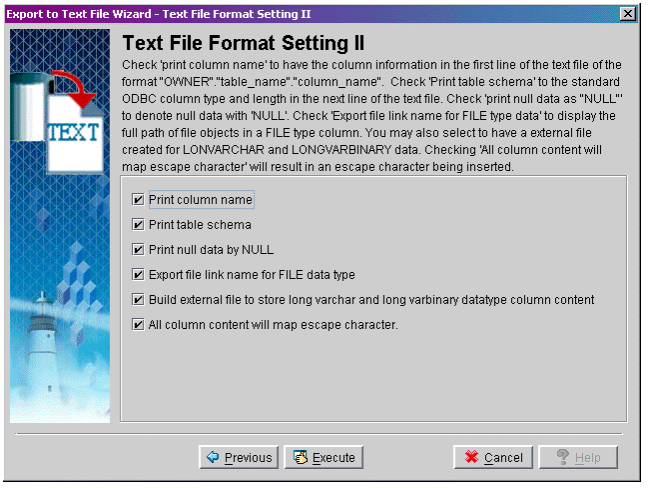
16. Enter the full path of a text file in the **Text file** field, or select one using the browse button.

17. Click **Next**. The **Text File Format Setting 1** window will open.



18. Select the appropriate settings for the format of the text file you will create.

19. Click **Next**. The **Text File Format Setting 2** window will open.



20. Finish selecting the appropriate settings for the format of the text file you are creating. Click **Execute** to export the data to the text file. A confirmation dialog box will appear.

21. Click **OK**

20.5 Exporting Data to XML

DBMaker supports the export of data from a table to an XML file. Columns may be stored as individual elements, or as attributes of the table element. When an XML file is created, an associated DTD file is created. The DTD contains information necessary for defining the elements and attributes of the XML file. The structure of both the DTD and XML file will vary depending on whether the columns are stored as attributes or elements.

Consider how the following settings affect the XML file produced by the Export to XML wizard.

Column as Element: If columns are represented as elements in the resultant files, then schema information will be retained as element attributes (data type, column name, length, etc.) in the DTD. Columns are child elements, and the table is represented as the parent element. If the XML file is later imported back into the database, then the table's structure will be exactly replicated. File objects are referenced as entities in the DTD file if Column as Element is chosen.

Column as Attribute: Columns are represented as attributes of the table element in the DTD. There is no record of the table's schema. An element in the XML file represents each record.

Export file link name for FILE type data: The original full path will reference system and user file objects if this option is selected. If this option is not selected, file type data will be treated as Long Varbinary.

Translate all tag names to uppercase: All tag names are converted to uppercase characters.

Build temp file to store LONGVARCHAR and LONGVARBINARY data type column constant: If this option is chosen, BLOB data will be stored in a temporary directory under the directory the XML file resides in. If this option is not selected, BLOB data is stored directly in the XML file.

XML file cannot include DTD file reference: if this option is selected, no DTD is created. No information about the elements will be preserved in the DTD if this option is selected.

➲ **Example 1**

Assume the table 'supportqueries' with columns 'LOGINID' CHAR(10); 'REQUEST' SQL_LONGVARCHAR; 'REQUESTTIME' SQL_TIMESTAMP; 'ATTACHMENT' 'SQL_FILE; 'BINARY_C' SQL_BINARY(10); DECIMAL_C SQL_DECIMAL(10, 3). The table has two records. The entire table is exported to an XML file with columns as elements. File link names are exported, temp files are built to store BLOB data, and the DTD is included. The resulting XML file follows:

```
<?xml version="1.0" encoding="BIG5"?>
<!DOCTYPE WEBDB SYSTEM "Support.dtd">
<WEBDB>
  <SUPPORTQUERIES>
    <LOGINID>A_HOWARD            </LOGINID>
    <REQUEST>&BLBTMP_TXT0;</REQUEST>
    <REQUESTTIME>2001-09-09 12:47:05.000</REQUESTTIME>
    <ATTACHMENT>&DBMAKER_FO_0;</ATTACHMENT>
    <BINARY_C>10000000000000000000</BINARY_C>
    <DECIMAL_C>10.250</DECIMAL_C>
  </SUPPORTQUERIES>
  <SUPPORTQUERIES>
    <LOGINID>A_HOWARD            </LOGINID>
    <REQUEST>&BLBTMP_TXT1;</REQUEST>
    <REQUESTTIME>2001-09-22 10:14:21.000</REQUESTTIME>
    <ATTACHMENT>&DBMAKER_FO_1;</ATTACHMENT>
    <BINARY_C>20000000000000000000</BINARY_C>
    <DECIMAL_C>13.550</DECIMAL_C>
  </SUPPORTQUERIES>
</WEBDB>
```

The associated DTD follows:

```
<!ELEMENT SUPPORTQUERIES (LOGINID, REQUEST, REQUESTTIME, ATTACHMENT, BINARY_C,
DECIMAL_C)>
<!ELEMENT LOGINID (#PCDATA)>
  <!ATTLIST LOGINID
      TYPE CDATA #FIXED "SQL_CHAR"
```

```
      NAME CDATA #FIXED "LOGINID"
      LENGTH CDATA #FIXED "20"
      ISNULL (true|false) 'true'
      xml:space (default|preserve) 'preserve'
  >
<!ELEMENT REQUEST (#PCDATA)>
  <!ATTLIST REQUEST
      TYPE CDATA #FIXED "SQL_LONGVARCHAR"
      NAME CDATA #FIXED "REQUEST"
      ISNULL (true|false) 'true'
      xml:space (default|preserve) 'preserve'
  >
<!ELEMENT REQUESTTIME (#PCDATA)>
  <!ATTLIST REQUESTTIME
      TYPE CDATA #FIXED "SQL_TIMESTAMP"
      NAME CDATA #FIXED "REQUESTTIME"
      STORAGE CDATA #FIXED "29"
      ISNULL (true|false) 'true'
      xml:space (default|preserve) 'preserve'
  >
<!ELEMENT ATTACHMENT (#PCDATA)>
  <!ATTLIST ATTACHMENT
      TYPE CDATA #FIXED "SQL_FILE"
      NAME CDATA #FIXED "ATTACHMENT"
      ISNULL (true|false) 'true'
      xml:space (default|preserve) 'preserve'
  >
<!ELEMENT BINARY_C (#PCDATA)>
  <!ATTLIST BINARY_C
      TYPE CDATA #FIXED "SQL_BINARY"
      NAME CDATA #FIXED "BINARY_C"
      LENGTH CDATA #FIXED "10"
      ISNULL (true|false) 'true'
      xml:space (default|preserve) 'preserve'
  >
<!ELEMENT DECIMAL_C (#PCDATA)>
  <!ATTLIST DECIMAL_C
      TYPE CDATA #FIXED "SQL_DECIMAL"
      NAME CDATA #FIXED "DECIMAL_C"
```

```
      LENGTH CDATA #FIXED "(10, 3)"
      ISNULL (true|false) 'true'
      xml:space (default|preserve) 'preserve'
  >
<!ENTITY BLBTMP_TXT0 SYSTEM "blobtmpdir0\blbtmpf0.txt">
<!ENTITY DBMAKER_FO_0 SYSTEM "C:\DBMAKER\4.1\BIN\WEBDB\FO\ZZ000000.GIF">
<!ENTITY BLBTMP_TXT1 SYSTEM "blobtmpdir0\blbtmpf1.txt">
<!ENTITY DBMAKER_FO_1 SYSTEM "C:\DBMAKER\4.1\BIN\WEBDB\FO\ZZ000001.GIF">
<!ENTITY BLBTMP_TXT2 SYSTEM "blobtmpdir0\blbtmpf2.txt">
<!ELEMENT WEBDB (SUPPORTQUERIES*)>
```

➲ **Example 2**

Given the same table as example 1, but with the entire table exported with columns as attributes. The resulting XML file follows:

```
<?xml version="1.0" encoding="BIG5"?>
<!DOCTYPE WEBDB SYSTEM "Support.dtd">
<WEBDB>
  <SUPPORTQUERIES
LOGINID="A_HOWARD&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x2
0;&#x20;" REQUESTTIME="2001-09-09 12:47:05.000"
ATTACHMENT="C:\DBMAKER\4.1\BIN\WEBDB\FO\ZZ000000.GIF"
BINARY_C="10000000000000000000" DECIMAL_C="10.250"/>
  <SUPPORTQUERIES
LOGINID="A_HOWARD&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x20;&#x2
0;&#x20;" REQUESTTIME="2001-09-22 10:14:21.000"
ATTACHMENT="C:\DBMAKER\4.1\BIN\WEBDB\FO\ZZ000001.GIF"
BINARY_C="20000000000000000000" DECIMAL_C="13.550"/>
```

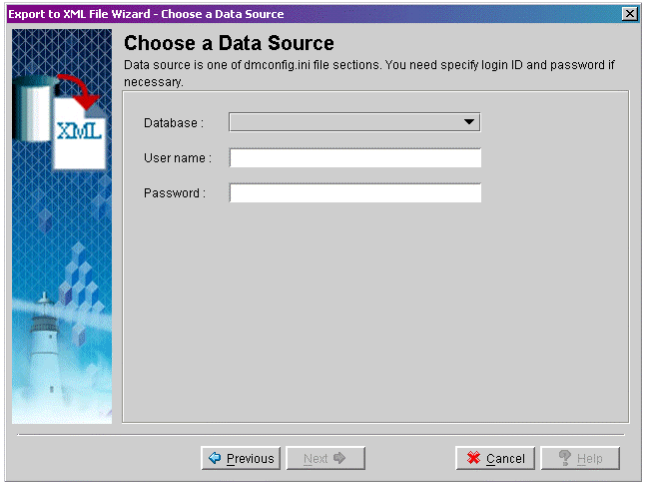
The associated DTD follows:

```
<!ELEMENT SUPPORTQUERIES EMPTY>
  <!ATTLIST SUPPORTQUERIES
              LOGINID     CDATA #IMPLIED
          REQUESTTIME     CDATA #IMPLIED
           ATTACHMENT     ENTITY #IMPLIED
            BINARY_C     CDATA #IMPLIED
           DECIMAL_C     CDATA #IMPLIED
  >
<!ELEMENT WEBDB (SUPPORTQUERIES*)>
```

➲ **To export a table to an XML file:**

1. Open the Data Transfer Tool.
2. Select **Export to XML** from the main console or the **Transfer** menu. The **Welcome to Export to XML File Wizard** window will appear.

3. Click **Next**. The **Choose a Data Source** window will open.

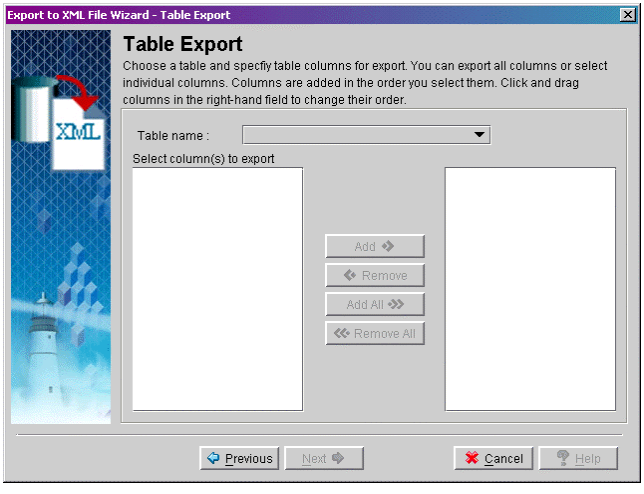


4. Select a database from the **Database** menu. Enter a user name and password into the appropriate fields.

5. Click **Next**. The **Table or Query Export** window will appear.

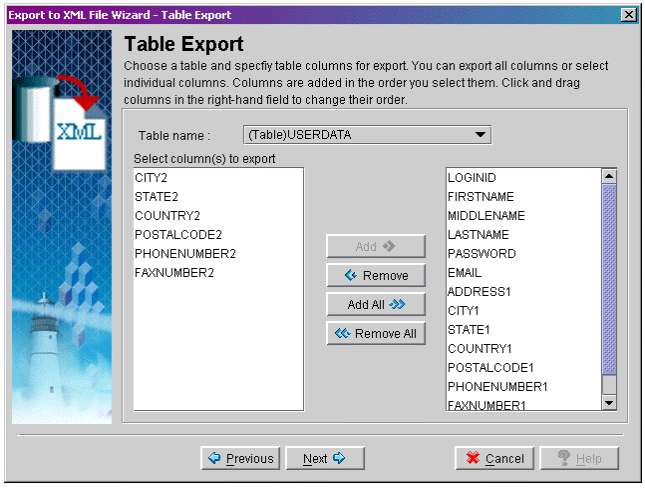
6. If you selected **Table** from the **Table or Query Export** window, the **Table Export** window will open. If you selected **SQL query**, then proceed to step 13.

7. Click **Next**, the **Table Export** window will open.



8. Select a table to export from the **Table name** menu. A list of columns in the table will appear in the **Select columns to export** field.

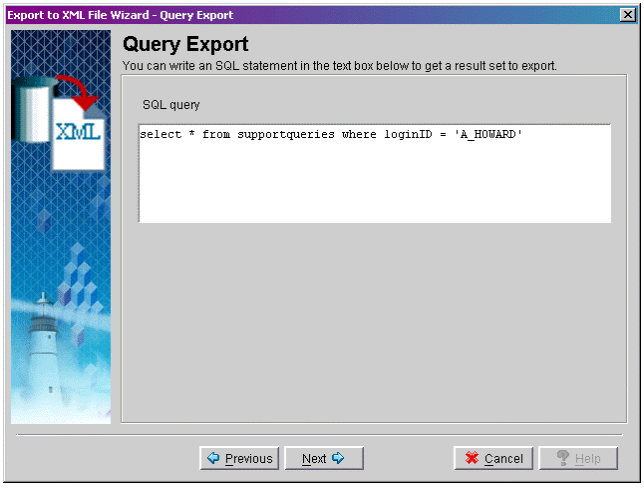
9. Select columns by clicking on the column name and clicking **Add**, or select all columns by clicking **Add All**. Selected column names will appear in the right hand field.



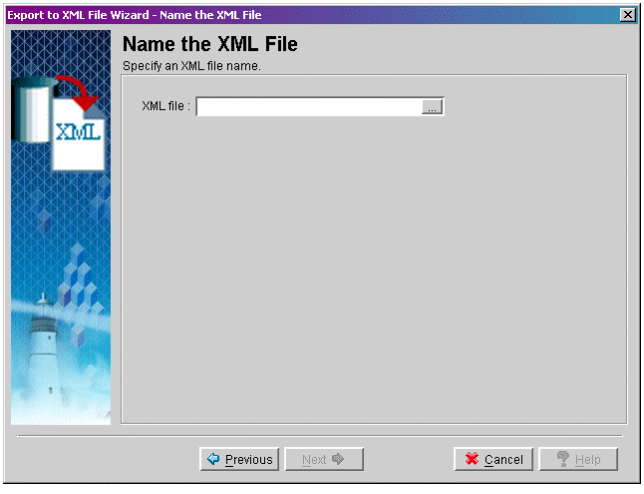
10. Click **Next**. The **Name The XML File** window will appear (proceed to step 16).

11. If you selected **SQL query** from the **Table or Query Export** window, the **Query Export** window will open.

12. Enter a valid SQL select statement into the **SQL query** field.

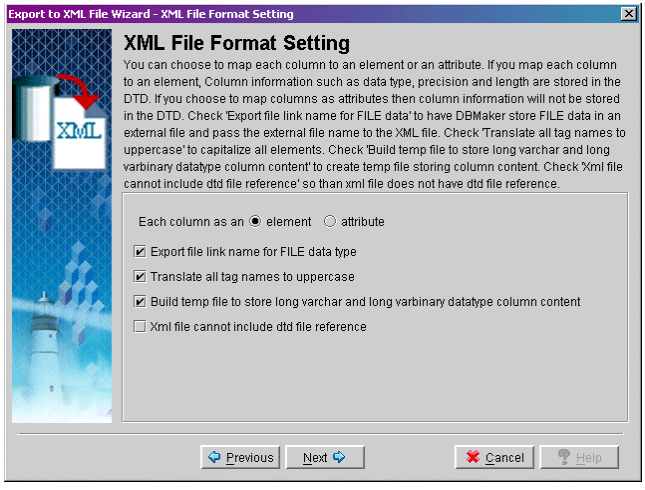


13. Click **Next**. The **Name The XML File** window will appear.



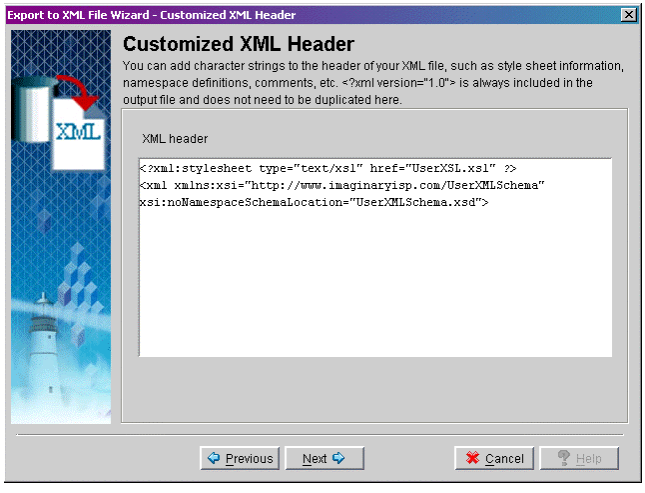
14. Enter the full path of an XML file to export to, or select one by using the browse button.

15. Click **Next**. The **XML File Format Setting** window will open.



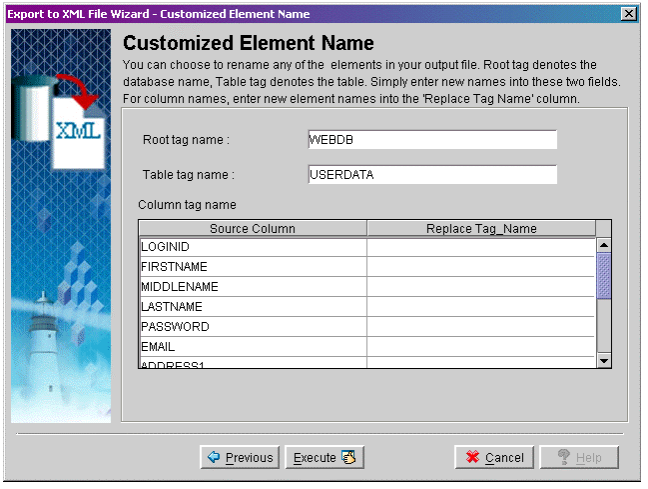
16. Select the appropriate settings for the format of the XML file you will create.

17. Click **Next**. The **Customized XML Header** window will open.



18. Enter appropriate information, such as namespace and style sheet definitions, if relevant.

19. Click **Next**. The **Customized Element Name** window will appear.



20. It is possible to modify the tag definitions. Enter new tag definitions into the **Replace Tag Name** column. The name of the corresponding column will be changed in the resulting XML file.

21. Click **Execute** to export the table to the XML file. A confirmation window will appear.

22. Click **OK**.

21 Monitoring a Database

DBMaker provides two interfaces for monitoring the status of a database: the Database Monitor Tool, and the Database Monitor tree object.

The Database Monitor Tool provides a graphical representation of various aspects of the database over time. It is ideally suited for monitoring overall database performance.

The Database Monitor tree object should be familiar to users who have experience with previous versions of JDBA Tool. It provides a series of tables that allow the viewer to monitor the status of individual connections.

21.1 Database Monitor Tool

The Database Monitor Tool is a separate user interface that may be accessed through the Tool menu of the JDBA Tool or through the Windows Start menu. Opening the Database Monitor Tool allows performance related statistics to be viewed on a graph that plots a statistical value on the Y-axis and time on the X-axis. Statistical parameters (or 'objects') to display on the graph are selected from a menu at the bottom of the screen. Each statistical parameter is sampled for a period of time, and the values are plotted on the x-axis. The scale of the graph determines the sample period.

The Database Monitor Tool provides an interface to customize aspects of the view, including the following:

Items: track up to 34 statistical parameters

Customize the line color, line width, and scale of each item

Select from sample periods of 5 seconds, 2 minutes, 1 hour, and 1 day.

Specify the grid color and background color of the chart.

Specify the scale of the chart's Y-axis.

The Database Monitor Tool can track 34 different performance-related statistical parameters. The statistical parameters are summarized in Table 20-1.

PARAMETER	DESCRIPTION
Number of Transactions Started	The number of transactions started within the sample period.
Number of Transactions Committed	The number of transactions committed within the sample period.
Number of Transactions Aborted	The number of transactions aborted during the sample period.
Number of Checkpoints	The number of checkpoints that occurred within the sample period.
Number of Lock Requests	The total number of locks requested during the sample period.

Number of Deadlocks Detected	The number of deadlocks occurring within the sample period.
Number of Journal Block Reads	The number of journal blocks read from file within the sample period.
Number of Journal Block Writes	The number of journal blocks written to file within the sample period.
Number of Journal Record Writes	The number of journal records generated during the sample period.
Number of Journal Forced Writes	The number events within the sample period that flush dirty journal buffer to disk.
Number of Commit Waits	The number of transactions within the sample period waiting for group commit transactions.
Number of Logical Reads	The number of page/frame read events from the data buffer within the sample period.
Number of Logical Writes	The number of page/frame write events from the data buffer within the sample period.
Number of Physical Reads	The number of page/frame read events from disk within the sample period.
Number of Physical Writes	The number of page/frame write events from disk within the sample period.
Number of Index Page Splits Occurring	The number of index pages split within the sample period.
Number of Data Pages Compressed	The number of data pages reorganized within the sample period.
Number of Row to Page Lock Escalations	The number of events within the sample period causing escalation from row locks to page locks.
Number of Page to Table Lock Escalations	The number of events within the sample period causing escalation from page locks to table locks.
Number of Failed Events due to Lock Timeout	The number of events within the sample period that failed due to a lock time-out
Number of Events Waiting for Locks	The number of events within the sample period that were required to wait for the release of a lock on an object.
Number of SELECT Operations	The number of SELECT statements within the sample period.
Number of INSERT Operations	The number of INSERT statements within the sample period.

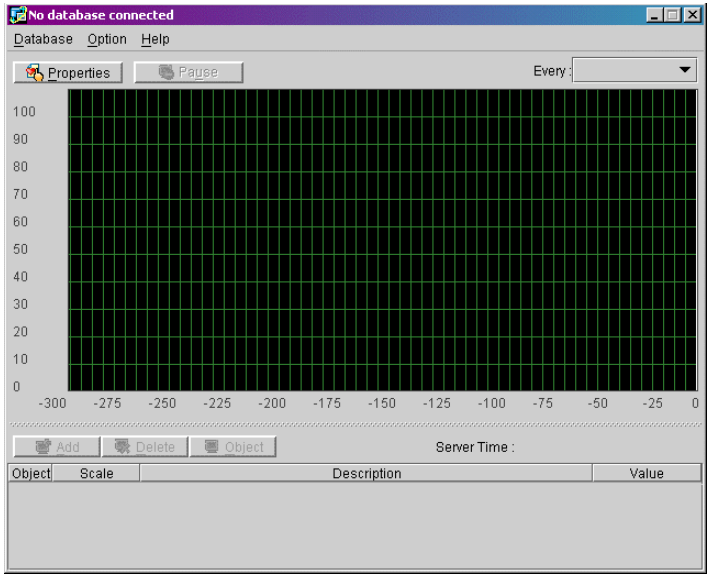
Number of UPDATE Operations	The number of UPDATE statements within the sample period.
Number of DELETE Operations	The number of DELETE statements within the sample period.
Number of SQLPrepare() Calls	The number of SQLPrepare() calls within the sample period.
Number of SQLExecute() Calls	The number of SQLExecute() calls within the sample period.
Number of SQLExecDirect() Calls	The number of SQLExecDirect() calls within the sample period.
Number of Fetches	The number of fetch operations within the sample period.
CPU Usage*	The average percentage of total CPU processing capacity experienced within the time period.
Peak Number of Connections	The maximum number of simultaneous connections occurring within the sample period.
Free Physical Memory (KB)	The average amount of free physical memory occurring within the sample period.
Total Swap Space (Bytes)	The total operating system memory swap space allocated on disk.
Free Swap Space (Bytes)	The total operating system memory swap space available on disk.

Table 21-1 Database Monitor Tool Statistical Parameters

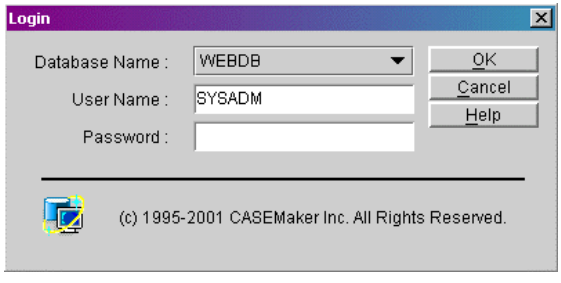
*Windows 2000 will only display the percentage of the first CPU in multi-processor systems.

➲ **To view statistical parameters with the Database Monitor Tool:**

1. Open the Database Monitor Tool from the Windows Start menu or from the Tool menu in the JDBA Tool. The Database Monitor Tool will start.

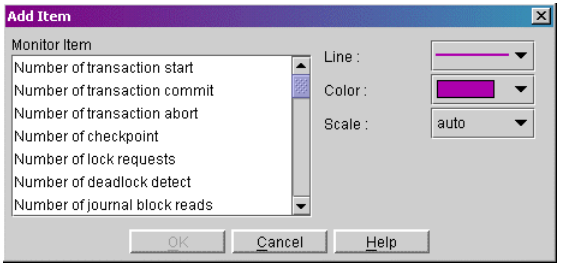


2. If the Database Monitor Tool was started from the Tool menu in JDBA Tool it will be automatically connected to the database. If the Database Monitor Tool was started from the Windows Start menu, you will need to connect to the database. Connect to a database by selecting **Connect Database** from the **Database** menu. The **Login** window will appear.



3. Select a database and enter a username and password into the appropriate fields. You will return to the Database Monitor Tool.

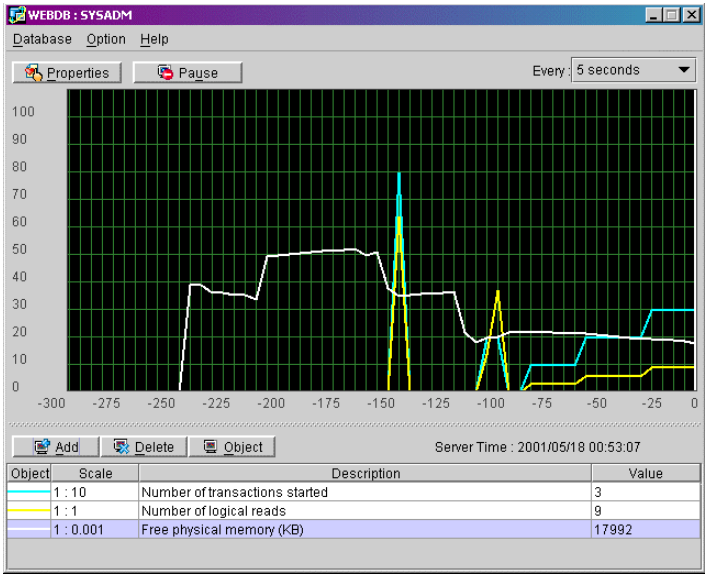
4. Click the **Add** button to add a statistical parameter to the chart. The **Add Item** dialog box will open.



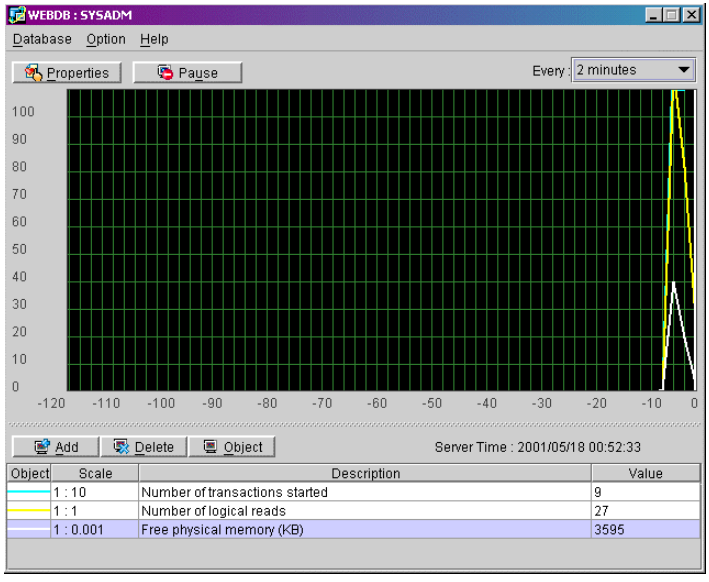
5. Select the item to monitor.

6. Select the line type, color, and scale.

7. Click **OK**. The **Add Item** dialog box will disappear and the monitor item will appear on the graph. Monitoring begins the moment you click **OK**. The newly plotted data will not appear until after the first sample period has elapsed.

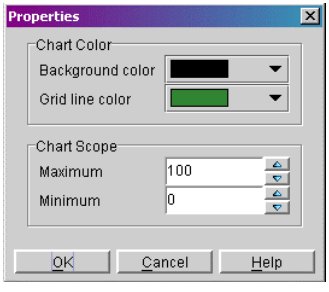


8. To change the sample period of the chart, select a value from the **Every** menu. The values of items will change in proportion to the new sample period.



9. To change the properties of the chart:

a) Click **Properties**. The **Properties** dialog will open.



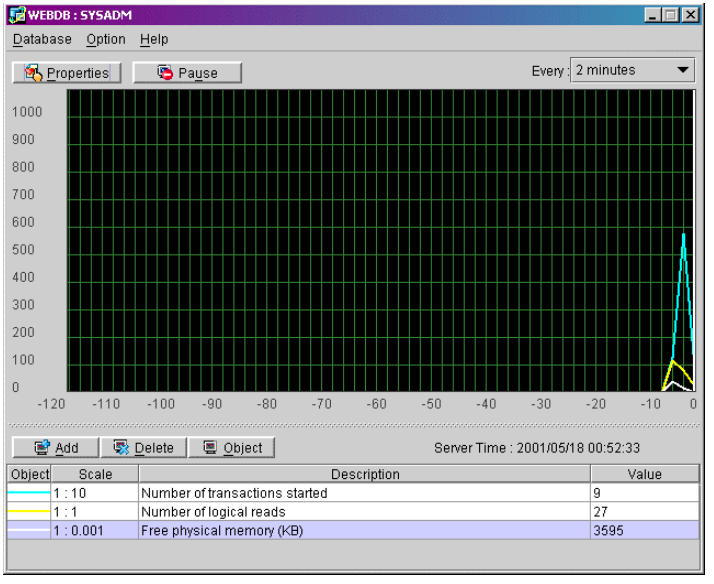
b) Select a background color and grid line color.

c) Enter a chart scope; minimum value will reflect the bottom of the chart, maximum value will reflect the top of the chart.

NOTE *The maximum value must be greater than the minimum value by at least 10, and the difference between minimum and maximum values*

must be a factor of 10. Maximum range (10 – 9,000,000). Minimum range (0 – 8,999,990)

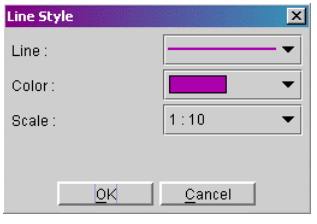
d) Click **OK**. The changes will be reflected in the chart.



10. To change the line type of an item:

a) Select one of the items.

b) Click **Object**. The **Line Style** window will open.



c) Select a new line type, color, and scale.

d) Click **OK**.

11. To delete an item:

a) Select one of the items.

b) Click **Delete**. The item will be removed from the chart.

12. Click **Pause** to stop the progress of the chart.

21.2 Database Monitor (Tree)

JDBA Tool provides monitoring functions that allow you to view the status of individual user connections, transactions, and upper memory handling. The statistics provided by the monitoring functions can be useful for database performance evaluation and tuning. There are four main database-monitoring functions provided by JDBA Tool: Session information, Lock status, Database status, and Pending Transaction status.

Session information shows all connections with the database (users, backup demon, IO daemon…), and information related to the connection. Sessions can be terminated by the DBA with the Kill connection function.

Lock status is for viewing database objects that have locks on them, and the type of lock on the object.

Database status allows the DBA to view statistics relating to journal files and buffers, as well as view the current database setup.

Pending Transaction status provides information on distributed transactions that have been prepared to commit but have not been committed yet.

Monitoring Database Sessions

Session information allows you to view information that relates to users connected to the database and resources used by each user. Session Information is divided into five categories:

Login Information

Execution Information

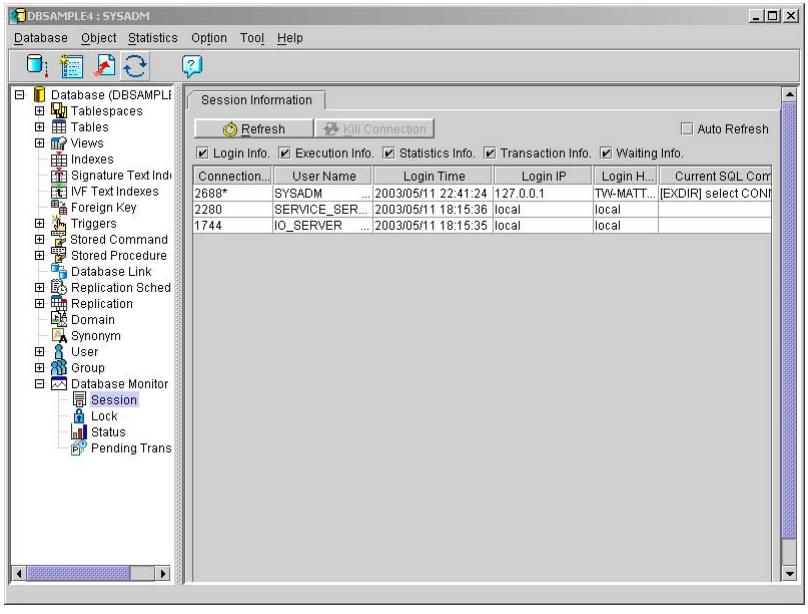
Statistical Information

Transaction Information

Waiting Information

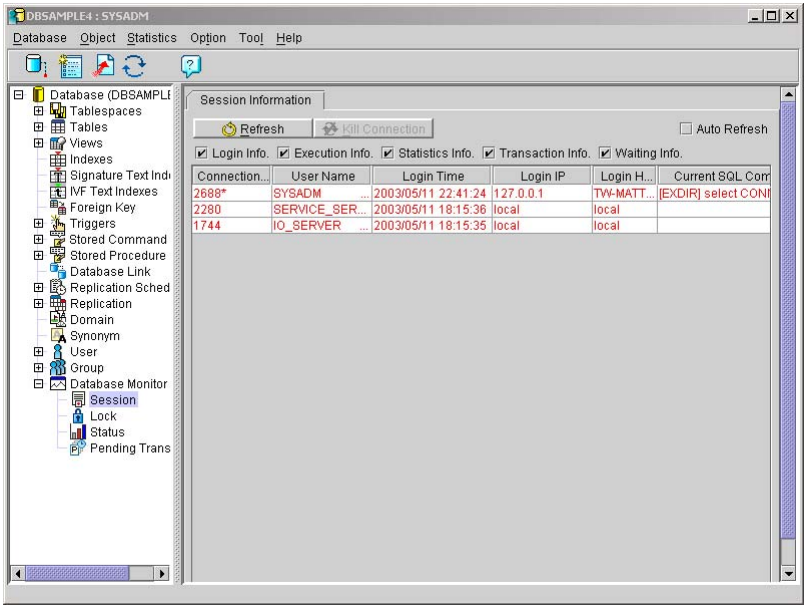
➲ **To view Session Information:**

1. Select the **Database Monitor** node from the tree. Four nodes will appear, select the **Session** node. The following window will appear.



2. You can view session information by sliding the scroll bar at the bottom of the page, or by selecting the appropriate check boxes next to each category.

3. After a refresh, newly updated information is shown in red. Any information that remained the same is shown in black.



LOGIN INFORMATION

This category contains the user's name and information relevant to the connection and computer the user is logged into.

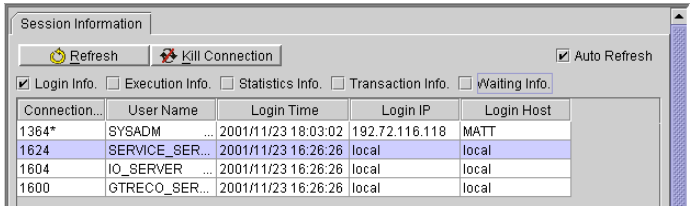


Figure 21-1 Session Information: Login Information

COLUMN NAME	DESCRIPTION
Connection ID:	The ID of the session.

Column Name	Description
User_Name:	The name of the user using the session.
Login_Time:	The time that the user connected to the database.
Login_IP_Addr:	The IP address of the user using the session.
Login_Host:	The host computer using the session.

Table 21-2 Login Information

Execution Information

This category describes the type and status of SQL commands being executed by each user.

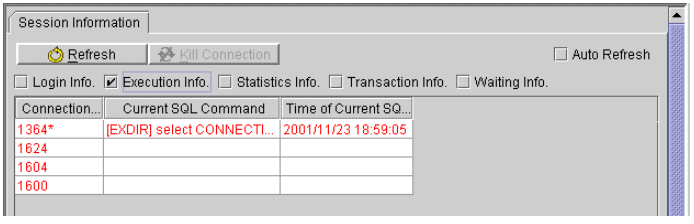


Figure 21-2 Session Information: Execution Information

Column Name	Description
Connection ID:	The ID of the session.

Column Name	Description
Current SQL Command	The most recently executed SQL command and the command status. The command status can be the following: [PRE] - The SQL command is being prepared. [EXEC] - The SQL command is being executed from SQLExecute call. [EXDIR] - The SQL command is being executed from SQLExecDirect call. [FETCH] - The operation is in the fetch data phase. [EXIT] - The SQL command is finished with the prepare, execute or fetch operation.
Time of Current SQL Command	The time when the most recently used SQL command was executed.

Table 21-3 Execution Information

Statistical Information

This category displays statistics relevant to the data.

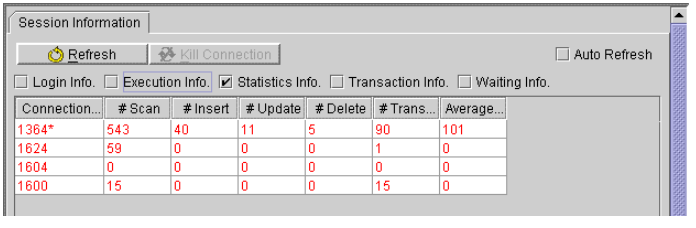


Figure 21-3 Session Information: Statistical Information

Column Name	Description
Connection ID:	The ID of the session.
#Scan:	The number of SELECT operations in the session.

Column Name	Description
#Insert:	The number of INSERT operations in the session.
#Update:	The number of UPDATE operations in the session.
#Delete:	The number of DELETE operations in the session.
#Tranx:	The number of transactions in the session.
Average Journal Bytes Per Transaction	The average number of journal bytes used per transaction in the session.

Table 21-4 Statistical Information

Transaction Information

Displays journal file activity, specifically the start and end points of transactions within the journal.

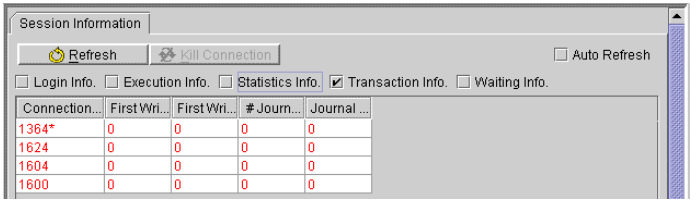


Figure 21-4 Session Information: Transaction Information

Column Name	Description
Connection ID:	The ID of the session.
First Written Journal block of Current Transaction	The first journal block that is fully written to in the current transaction..
First Written Journal file of Current Transaction	The first journal file that is fully written to in the current transaction.

Column Name	Description
# Journal Bytes of Current Transaction	The total amount of journal data used in the current transaction in the session.
Journal Blocks Distance of Current Transaction	The distance of the current journal block from the first written journal block in the current transaction.

Table 21-5 Transaction Information

Waiting Information

Displays information on which users are waiting for locks on objects and which users have the locks on the objects being waited for.

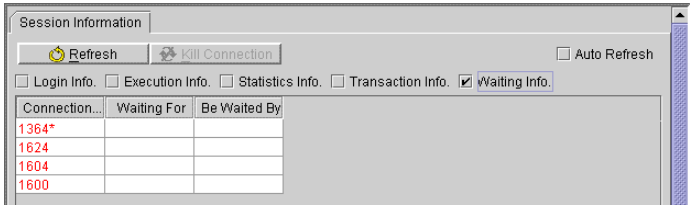


Figure 21-5 Session Information: Waiting Information

Column Name	Description
Connection ID:	The ID of the session.
Waiting For	The connection ID number of a user that has a lock on an object that the current user's transaction is waiting for
Be Waited By	The connection ID of a user that is waiting for release of a lock on an object held by the current user.

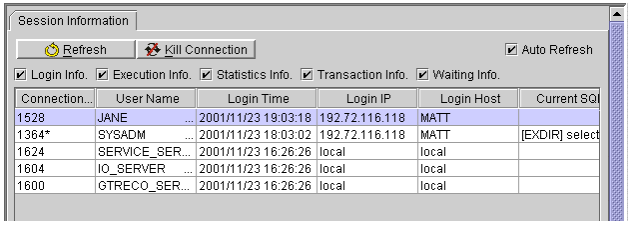
Table 21-6 Waiting Information

Killing Connections

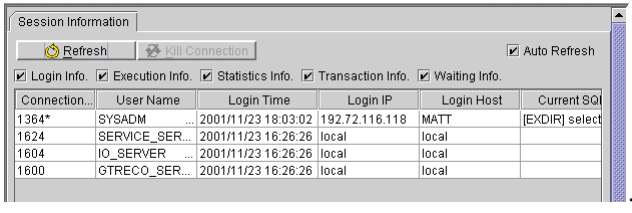
You can choose to terminate an active connection.

➲ **To kill a connection:**

1. Select the **Database Monitor** node from the tree. Four nodes will appear, select the **Session** node. The **Session Information** page will appear.
2. Select a connection from the list in the **Session Information** table. It will highlight blue and the **Kill Connection** field will be active.



3. Clicking **Kill Connection** will terminate the connection.



Monitoring Database Locks

You can monitor details of the locks used in the database. This information includes the lock table, lock granularity, lock status, and lock modes for all data objects.

Table 21-7 summarizes the items on the lock page:

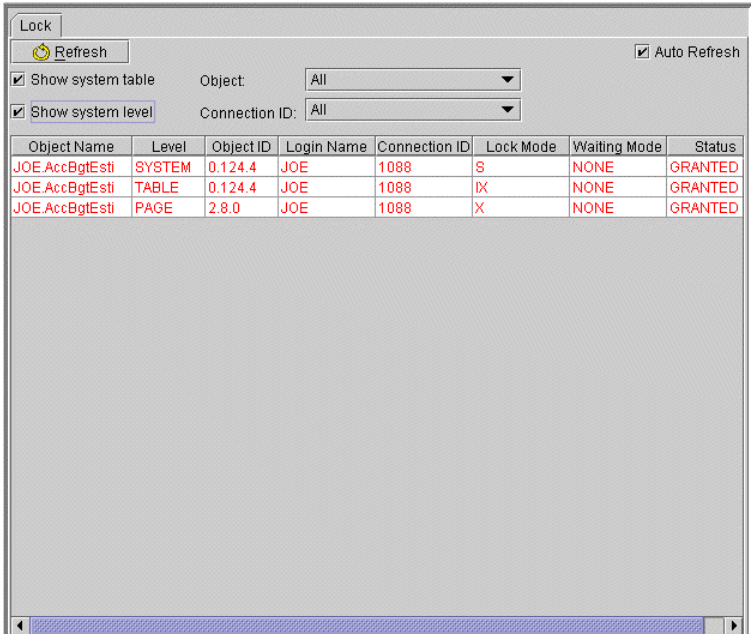
ITEM	DESCRIPTION
Object Name:	The object name for a locked object, or an object that a lock is requested for.

ITEM	DESCRIPTION
Level:	The current lock level. Four lock levels exist for an object: System, Table, Page, and Row.
Object ID	A identifying number unique to each table, page, or row in the database.
Login Name	Name of the user that has the lock on the object.
Connection ID:	The ID of the session
Lock Mode:	The current lock mode of an object. The shown lock modes are either S, IS, IX, X and SIX. IS, SIX, and IX denote that the object has an intention lock; part of the lock on the object is specified at a lower granularity.
Waiting Mode:	The lock mode that the user is requesting. It can be "none", indicating that the user does not desire a new lock on the object, or "S", "IS", "IX", "X" or "SIX", indicating that the user has requested the corresponding lock type on the object and is waiting for the lock.
Status:	Status indicates the condition of the Waiting Mode. "Granted" indicates that the user has been granted the requested lock and should correspond to a waiting mode of "none". "Waiting" indicates that the user has not yet been granted the requested object lock and should correspond to a Waiting mode of "S", "IS", "IX", "X" or "SIX". "Convert" indicates that the user has requested a change in lock level (for example, from "S" to "X") and the user is waiting for the new lock level on the object.

Table 21-7 Database Lock items

➲ **To view database lock status:**

1. Select the **Lock** node from the **Database Monitor** node in the tree. The following window will appear.



2. Click **Refresh** at any time to update the statistics shown on this page. You may also choose what information to display on the **Lock** page.

Selecting **Show System Table** allows you to see any locks placed on the system table.

Selecting **Show System Level** allows the DBA to see locks placed on objects in the system level.

You can choose to view locks only on certain objects. Select an object name from the **Object** menu.

You can choose to view locks on all objects associated with a user. Select a connection ID from the **Connection ID** menu.

Monitoring Database Status

The database status node of the database monitor tree item gives access to database statistics stored in the SYSINFO system table. Database status items are grouped into eight categories of related items. Each category is grouped together on a tabbed page. The categories include the following:

Locks

Connections

Data Operations

Database Info

System Info

Page and I/O

Journal

Transaction

VIEWING LOCK STATUS

Lock status items are statistics related to the cumulative number of locks, lock escalations, deadlocks, and failed events due to lock timeouts. The lock status page displays lock information statistics from the SYSINFO table. Lock status items and their corresponding SYSINFO.INFO values are summarized in Table 21-8.

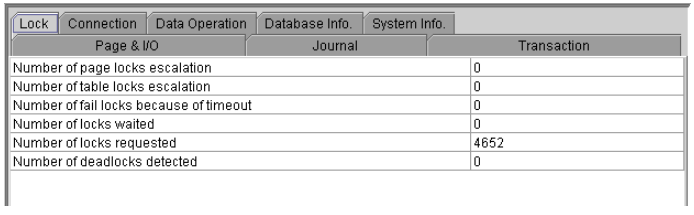
LOCK STATUS ITEM	DESCRIPTION
Number of Page Locks Escalation	The accumulated number of row locks that have been escalated to page locks. (NUM_ROW_LOCK_UPG)
Number of Table Locks Escalation	The accumulated number of page locks that have been escalated to table locks. (NUM_PAGE_LOCK_UPG)

Lock Status Item	Description
Number of fail locks because of timeout	The accumulated number of actions failing due to a timeout while waiting to acquire a lock on an object. (NUM_LOCK_TIMEOUT)
Number of Locks Waited	The accumulated number of events waiting to place locks on objects. (NUM_LOCK_WAIT)
Number of Locks Requested	The accumulated number of locks requested on data objects. (NUM_LOCK_REQUEST)
Number of deadlocks detected	The accumulated number of deadlocks detected. (NUM_DEADLOCK)

Table 21-8 Lock Status Items

➲ **To view Lock Status:**

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Lock** tab. The following page will be displayed.



Viewing Connection Status

The connection status page displays connection information statistics from the SYSINFO table. Connection status items and their corresponding SYSINFO.INFO values are summarized in Table 21-9.

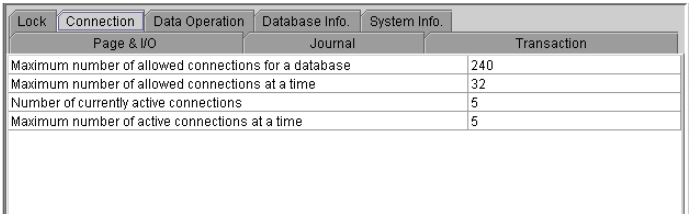
Connection Status Item	Description
Maximum number of allowed connections for a database	The hard limitation of connections. This value is determined by the DB_MaxCo keyword when the database is created or started with a new journal. (NUM_MAX_HARD_CONNECT)

Connection Status Item	Description
Maximum number of allowed connections at a time	The soft limitation of connections, which is less than or equal to the number of hard connections (NUM_MAX_SOFT_CONNECT)
Number of currently active connections	The current number of connections to the database (NUM_CONNECT)
Maximum number of active connections at a time	Records the largest number of active connections experienced by the database since the database was started (NUM_PEAK_CONNECT)

Table 21-9 Connection Status items

➲ **To view Connection Status:**

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Connection** tab. The following page will be displayed.



Viewing Data Operation Status

The data operation status page displays data operation statistics from the SYSINFO table. Data operation status items and their corresponding SYSINFO.INFO values are summarized in Table 21-9.

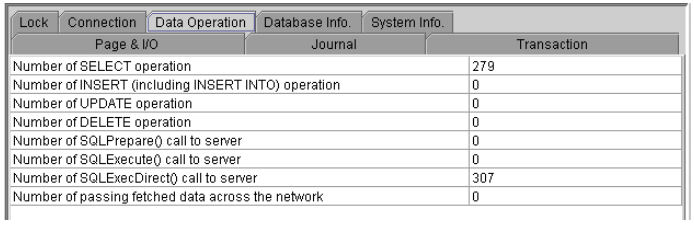
Data Operation Status Item	Description
Number of SELECT Operations	The total number of SELECT operations performed since the database was started. (NUM_SQL_SELECT)

Data Operation Status Item	Description
Number of INSERT (including INSERT INTO) Operations	The total number of SELECT operations performed since the database was started. (NUM_SQL_ INSERT)
Number of UPDATE Operations	The total number of UPDATE operations performed since the database was started. (NUM_SQL_UPDATE)
Number of DELETE Operations	The total number of DELETE operations performed since the database was started. (NUM_SQL_DELETE)
Number of SQLPrepare() calls to server	The total number of SQLPrepare() calls to the database since the database was started. (NUM_SQL_PREPARE)
Number of SQLExecute() calls to server	The total number of SQLExecute()calls to the database since the database was started. (NUM_SQL_EXECUTE)
Number of SQLExecDirect() calls to server	The total number of SQLExecDirect()calls to the database since the database was started. (NUM_SQL_EXEDIRECT)
Number of passing fetched data across the network	The total number of fetch operations performed by clients since the database was started. (NUM_SQL_FETCH)

Table 21-10 Data Operations Status items

➲ **To view Data Operation Status:**

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Data Operation** tab. The following page will be displayed.



VIEWING DATABASE INFORMATION

The database information page displays database information statistics from the SYSINFO table. Database information items and their corresponding SYSINFO.INFO values are summarized in Table 21-11.

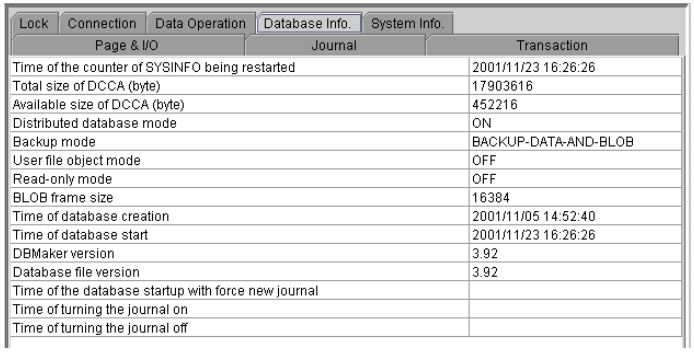
DATABASE INFORMATION ITEMS	DESCRIPTION
Time of the counter of SYSINFO being started	The last time and date that data in the SYSINFO table was reset. The SYSINFO table gets reset when one of the counters overflows, or the SET SYSINFO CLEAR command is used. (SYSINFO_RESET_TIME)
Total size of DCCA	Total memory size of the Database Communication and Control Area (DCCA). Refer to the *Database Administrators Guide* for detailed descriptions of the DCCA. (DCCA_SIZE)
Available size of DCCA	Amount of memory available in the DCCA. (FREE_DCCA_SIZE)
DDB Mode	Distributed Database Mode. (DDB_MODE)
Backup Mode	Indicates whether incremental backups can be performed. It also indicates whether the incremental backup will include BLOB files, 1 means to backup the data file only, and 2 means to backup both data and BLOB files. (BACKUP_MODE)
User File Object Mode	Shows if user file objects have been enabled. A value of 1 means user file objects are enabled. (USER_FO_MODE)
Read-Only Mode	Shows if the database has been started in read-only mode. A value of 1 means he database has been started in read-only mode. (READ_ONLY_MODE)
Blob Frame Size (Bytes)	The size of each BLOB frame in bytes. (FRAME_SIZE)
Time of database creation	The date and time the database was created. (CREATE_DB_TIME)
Time of database start	The date and time the database was last started. (START_DB_TIME)

DATABASE INFORMATION ITEMS	DESCRIPTION
DBMaker version	The version number of the DBMaker engine. (VERSION)
Database file version	The version number of the database (FILE_VERSION)
Time of database startup with force new journal	The date and time the database was last started in new journal mode. (FORCE_NEW_JNL_TIME)
Time of turning the journal on	The most recent date and time the journal was started. (END_NO_JNL_TIME)
Time of turning the journal off	The most recent date and time the journal was stopped. (START_NO_JNL_TIME)

Table 21-11 Database Information Items

➲ **To view Database Information:**

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Database Info** tab. The following page will be displayed.



VIEWING SYSTEM INFORMATION

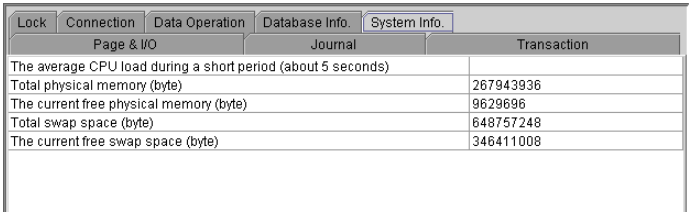
The system information page displays operating system statistics from the SYSINFO table. System information items and their corresponding SYSINFO.INFO values are summarized in Table 21-11.

System Information Item	Description
The average CPU load during a short period (about 5 seconds)	The maximum percentage of total CPU processing capacity experienced within a time period. (CPU_USAGE)
Total physical memory	The total physical memory available to the system (TOTAL_MEMORY)
The current free physical memory (KB)	The amount of free physical memory. (TOTAL_FREE_MEMORY)
Total swap space (Bytes)	The total operating system memory swap space allocated on disk. (TOTAL_SWAP_SPACE)
The current free swap space (Bytes)	The total operating system memory swap space available on disk. (TOTAL_FREE_SWAP_SPACE)

Table 21-12 System Information items

➲ **To view System Information:**

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **System Info** tab. The following page will be displayed.



Viewing Page and I/O Status

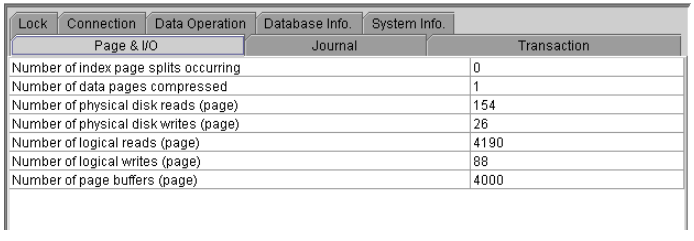
The page and I/O status page displays operating system statistics from the SYSINFO table. Page and I/O status items and their corresponding SYSINFO.INFO values are summarized in Table 21-13.

PAGE AND I/O STATUS ITEMS	DESCRIPTION
Number of Index Page Splits Occurring	The accumulated number of index pages split in the current run time. (NUM_IDX_SPLIT)
Number of Data Pages Compressed	The accumulated number of data pages reorganized in the current run time. (NUM_PAGE_COMPRESS)
Number of Physical Reads	The accumulated page number of page/frame disk reads in the current run time. (NUM_PHYSICAL_READ)
Number of Physical Writes	The accumulated number of page/frame disk writes in the current run time. (NUM_PHYSICAL_WRITE)
Number of Logical Reads	The accumulated page number of logical reads (reading data from page buffer cache) in the current run time. (NUM_LOGICAL_READ)
Number of Logical Writes	The accumulated number of logical writes (writing data into page buffer cache) in the current run time. (NUM_LOGICAL_WRITE)
Number of Page Buffers	The number of page buffers currently used by the database. (NUM_PAGE_BUF)

Table 21-13 Page and I/O Status items

➲ **To view Page and I/O Status:**

1. Select the **Status** node from the **Database Monitor** node in the tree.
2. Select the **Page & I/O** tab. The following page will be displayed.



VIEWING JOURNAL STATUS

The journal status page displays operating system statistics from the SYSINFO table. Journal status items and their corresponding SYSINFO.INFO values are summarized in Table 21-14.

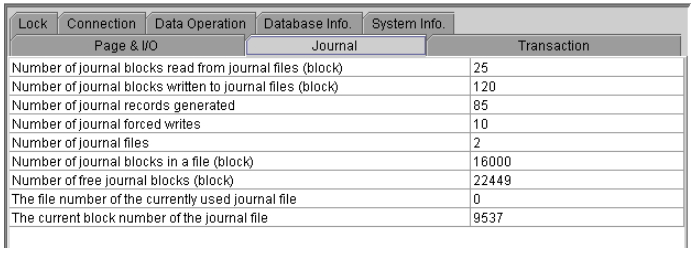
JOURNAL STATUS ITEMS	DESCRIPTION
Number of journal blocks read from journal files	The accumulated number of journal block reads in the current run time. (NUM_JNL_BLK_READ)
Number of journal blocks written to journal files	The accumulated number of journal block writes in the current run time. (NUM_JNL_BLK_WRITE)
Number of journal records generated	The accumulated number of journal record writes in the current run time. (NUM_JNL_REC_WRITE)
Number of journal forced writes	The accumulated number of journal force writes in the current run time. (NUM_JNL_FRC_WRITE)
Number of journal files	The number of blocks set in a journal file. (NUM_JOURNAL_FILE)
Number of journal blocks in a file	The number of journal blocks in each journal file. (NUM_JOURNAL_BLOCKS)
Number of free journal blocks	The current number of free journal blocks in all journal files. (NUM_JNR_BLOCK_FREE)
File number of the currently used journal file	For databases with multiple journal files, the number of the current journal file (CURRENT_JOURNAL_FN)
The current block number of the journal file	The number of the currently used journal block. (CURRENT_JOURNAL_BN)

Table 21-14 Journal Status items

➲ **To view Journal Status:**

1. Select the **Status** node from the **Database Monitor** node in the tree.

2. Select the **Journal** tab. The following page will be displayed.



VIEWING TRANSACTION STATUS

The transaction status page displays operating system statistics from the SYSINFO table. Transaction status items and their corresponding SYSINFO.INFO values are summarized in Table 21-14.

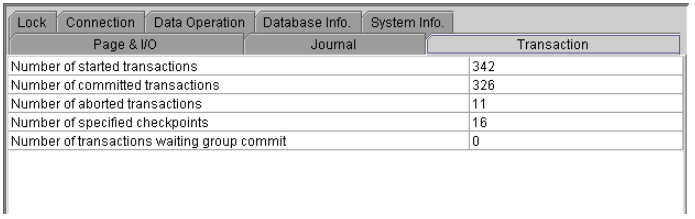
TRANSACTION STATUS ITEMS	DESCRIPTION
Number of started transactions	The cumulative number of transactions started during the current runtime. (NUM_STARTED_TRANX)
Number of committed transactions	The cumulative number of committed transactions during the current runtime. (NUM_COMMITED_TRANX)
Number of aborted transactions	The cumulative number of transactions aborted in the current runtime. (NUM_ABORTED_TRANX)
Number of specified checkpoints	The cumulative number of checkpoints made during the current runtime. (NUM_CHECKPOINTS)
Number of transactions waiting group commit	The cumulative number of transactions waiting for a group commit action during the current runtime. (NUM_COMMIT_WAITER)

Table 21-15 Transaction Status items

➲ **To view Transaction Status:**

1. Select the **Status** node from the **Database Monitor** node in the tree.

2. Select the **Transaction** tab. The following page will be displayed.



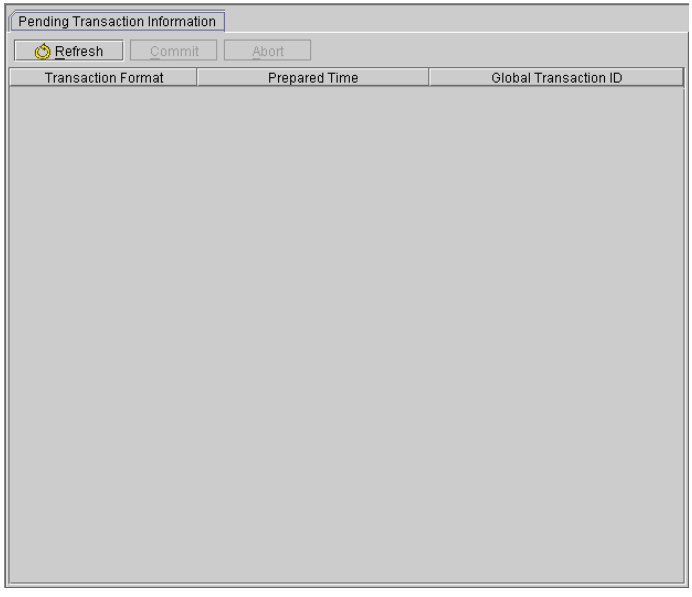
Monitoring Pending Transactions

In a distributed database environment, transactions can be made across multiple databases. To effectively manage this, DBMaker uses a two-phase commit method. The user inputs a command affecting the remote databases at the coordinator database. First, the coordinator database sends the command to the appropriate remote databases where they prepare to commit the transaction. After confirmation that all remote databases are prepared, the coordinator database sends a COMMIT transaction command to the remote databases, where the transaction is completed. It is possible that a network failure can occur between these two phases. The global transaction recovery daemon is responsible for logging all such errors and holding them as pending transactions until they can be committed at a later time. Pending transactions can be viewed using the JDBA Tool and can also be committed or aborted.

➲ **To view pending transactions:**

1. Select **Pending Transaction** from the **Database Monitor** node of the tree

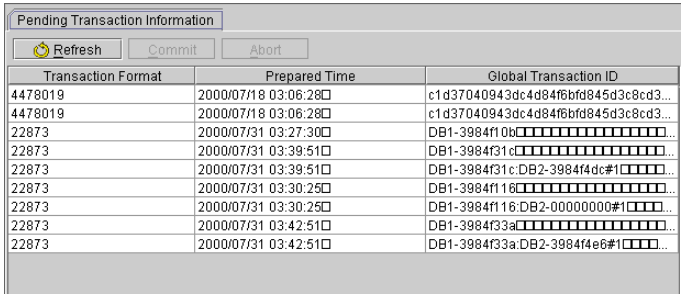
2. A list of current pending transactions will appear.



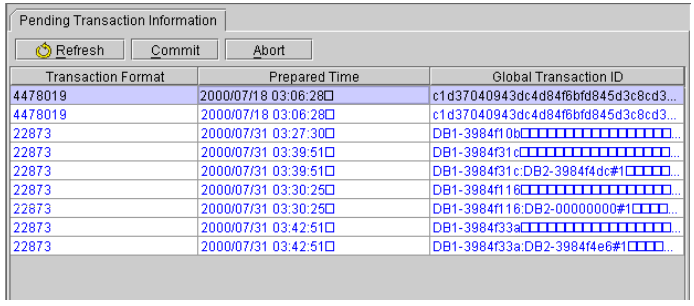
3. Select the **Refresh** button to view the current pending transaction status.

➲ **To commit pending transactions:**

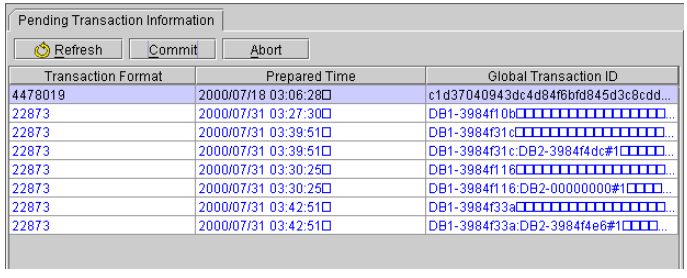
1. Select **Pending Transaction** from the **Database Monitor** node of the tree
2. A list of current pending transactions will appear.



3. Select the transaction to commit from the list in the main console. It will highlight blue.



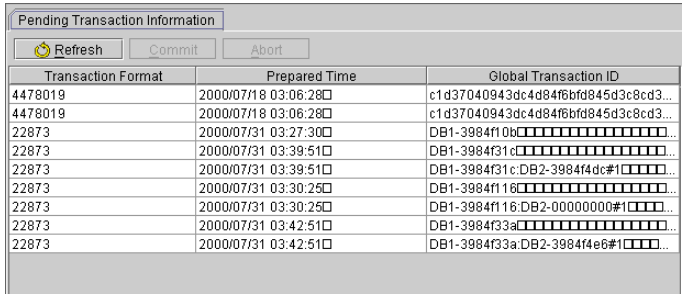
4. Click the **Commit** button at the top of the page. The transaction will commit and will no longer appear in the list. The next transaction will highlight blue.



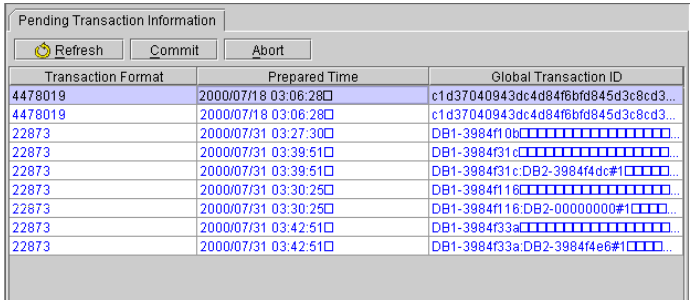
➲ **To abort pending transactions:**

1. Select **Pending Transaction** from the **Database Monitor** node of the tree

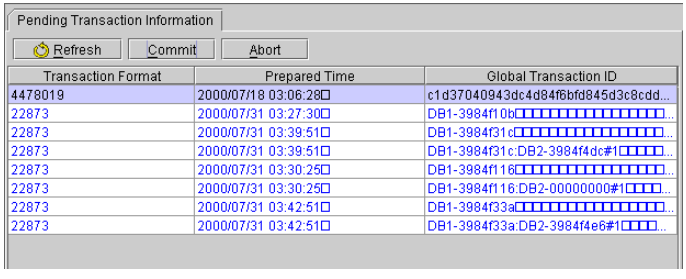
2. A list of current pending transactions will appear.



3. Select the transaction to abort from the list in the main console. It will highlight blue.



4. Click the **Abort** button at the top of the page. The transaction will abort and will no longer appear in the list. The next transaction will highlight blue.



22 Setting JDBA Options

The JDBA Options menu gives access to the following menu items:

Preferences: Provides a method for adjusting the way certain information is presented in the user interface. Preferences include language setting, auto-refresh, table, and SQL command display options.

Run Time settings: The Run Time Settings window provides an interface for changing database configuration parameters while the database is running. Settings include Backup settings, file object settings, system control settings (including distributed database mode and autoextend tablespace options), and connection settings.

Refresh: The Refresh option gets current database information to display on the JDBA window, and renews the screen.

22.1 Altering Run Time Settings

Certain database configuration parameters may be changed while the database is on-line. JDBA Tool provides an interface for changing these settings quickly. These settings include:

Backup settings: Allow you to modify automatic full and incremental backup frequency and location, set file object backup mode, or disable the backup server daemon.

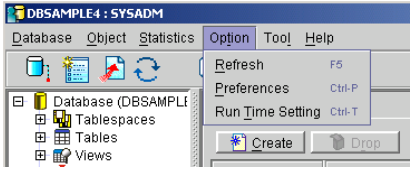
File object settings: Allow you to enable or disable user file objects and modify the location of system file objects.

System control settings: Allow you to enable distributed database mode for synchronous table replication, and set the number of pages or frames to extend autoextend tablespaces.

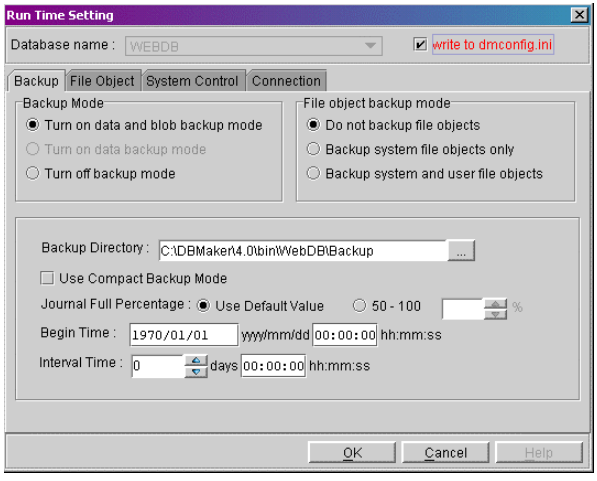
Connection settings: Give the option to block new connections.

➲ **To access the Run Time settings window:**

1. Select **Run Time Settings** from the **Option** menu on the menu bar.



2. The **Run Time Setting** dialog appears.



3. To use the updated settings in the next session, make sure that the **write to dmconfig.ini** check box is enabled.

4. To allow the updated settings to apply to the current session only, clear the checkmark in the **dmconfig.ini** check box.

NOTE *The **write to dmconfig.ini** option is only available on the server side.*

5. Set the following options:

Backup Settings

File Object Settings

System Control

Connection

Changing Backup Settings

Incremental Backup settings that may be altered in the runtime include the following: file object backup mode, backup location, and deactivation of backup BLOB or deactivation of backup data. During runtime, backup modes may only be changed to a state where less data is backed up (except in the case of file objects). This may be

useful, for example, if the database administrator wants to free up resources by changing the backup mode from "backup data and BLOB" to "backup data"

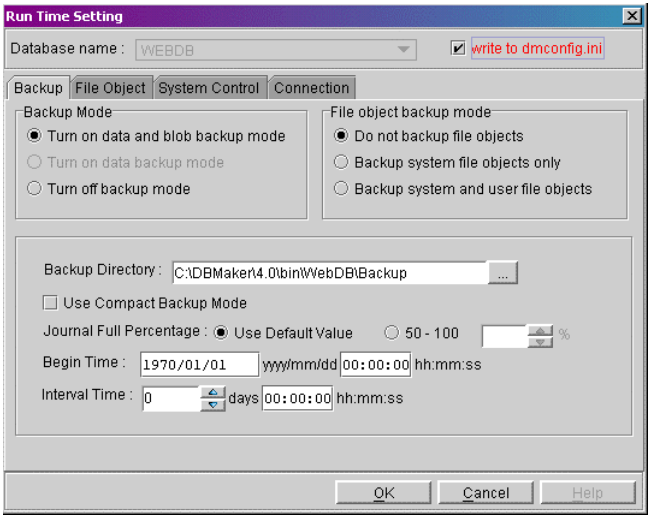


Figure 22-1 The Backup page of the Run Time Settings dialog

BACKUP MODE

You can choose to turn off the incremental backup mode, or turn off just backup BLOB mode. Run time settings only allow for deactivation of the incremental backup mode, either deactivating backup BLOB, or deactivating backup data.

To change the backup mode to Backup Data or Backup Data and BLOB you must shut down the database, start the database, select the **Setup** button, and make the changes to the **Backup** page of the **Start Database Advanced Settings** window.

The following backup modes are available.

BACKUP MODE	DESCRIPTION
Turn off backup mode	Disables the incremental backup daemon. In this mode the journal files are not backed up.
Turn on data backup mode	Sets the incremental backup daemon to backup all data journal pages.

Backup Mode	Description
Turn on data and BLOB backup mode is	BLOB data is written to the journal and the incremental backup daemon backs up all journal files.

Table 22-1 Backup Modes

File Object Backup Mode

You can choose what types of file objects to backup during incremental backups by changing the file object backup mode. The following file object backup modes are available.

FO Backup Mode	Description
Do not backup file objects	No file objects are backed up during full or incremental backups
Backup system file objects	System file objects are backed up during any full or incremental backup.
Backup system and user file objects	System and user file objects are backed up during any full or incremental backup.

Table 22-2 File Object Backup Modes

Backup Directory

This is the directory where the backup server puts all full backup files and incremental backup (journal) files. You should create a backup directory on a different disk from the database files to prevent the loss of both the database and the backup files in the event of a media error. The default path for backup files is the database directory */backup* and is automatically created by DBMaker. The total length of the backup directory path must not exceed 255 characters.

Use Compact Backup Mode

User Compact Backup Mode ensures that only needed journal files are copied by the Backup Server when performing a backup. As a result, storage space is saved in the backup location. However, it also means restoring a database may take more time.

JOURNAL FULL PERCENTAGE

The user may want to allow DBMaker to create an incremental backup before the journal file is completely full. The journal trigger value specifies the percentage a journal file must fill before the Backup Server performs an on-line incremental backup. You can specify a percentage between 50 and 100 for the journal trigger value. You can combine the journal trigger value with the backup schedule to backup your database on a regular schedule.

INCREMENTAL BACKUP BEGIN TIME

To set the time in which the first incremental backup will begin for the database, enter the date in the **yyyy/mm/dd** field, as well as the time in the **hh:mm:ss** field. Reenter the numbers if they appear incorrectly the first time; JConfiguration Tool automatically enters values into the first two spaces of the **yyyy** field and the first space of the **hh** field if values are entered into other fields.

INCREMENTAL BACKUP INTERVAL TIME

The number in the combo box specifies the time interval at which the Incremental Backup occurs in days. Next to the combo box is a field for time input, which specifies the time interval in hours, minutes, and seconds. The total time interval is determined by adding the two values together, so inputting 1 into the **days** field and inputting 12:00:00 into the **hh:mm:ss** field would cause an incremental backup to be made every day and a half.

➲ **To change backup settings:**

1. Select a backup mode from the **backup mode** box.

   To change the backup mode to backup data only, select the **Turn on data backup mode** option button.

   To turn off the backup mode, click the **Turn off backup mode** option button.

2. Select a file object backup mode from the **File object backup mode** box.

To not backup file objects, select the **Do not backup file objects** option button.

To backup system file objects, click the **Backup system file objects** option button.

To backup system and user file objects, click the **Backup system and user file objects** option button.

3. Enter a path into or select the browse button ... next to the **Backup Directory** field to indicate a destination for the full and incremental backup files.
4. To enable compact backup, click on the **Use Compact Backup Mode** check box.
5. Incremental backups can be set to automatically execute when journal files have filled to a set percentage. Next to **Journal Full Percentage**:

   Select the **Use Default Value** option button to set incremental backups to execute when any journal file is completely filled.

   Enter a value from 50 to 100 in the **50 – 100 %** field to set incremental backups to execute when any journal file is filled to the value entered.
6. Indicate a date and a time at which incremental backups are to begin in the **Begin Time** fields.
7. Enter the number of days, hours, minutes, and seconds between each successive incremental backup in the **Interval_Time** time fields.
8. Select one of the other tabs at the top (to change other runtime settings) or select **OK** from the bottom of the **Run Time Settings** window

Changing File Object Settings

Selecting the **File Object** tab in the **Run Time Settings** window allows the user to enable external user file objects and change the storage location of system file objects. DBMaker can automatically create subdirectories within the FO directory. Each subdirectory is filled with new file objects up to a threshold value. When the threshold is reached, DBMaker creates a new FO subdirectory.

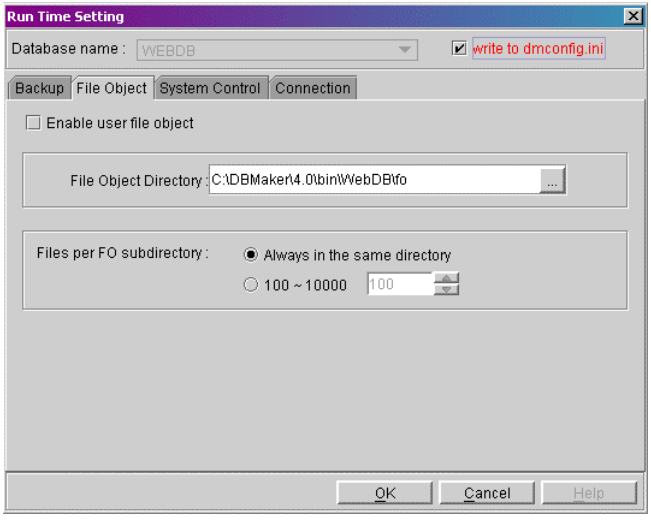


Figure 22-2 The File Object page of the Run Time Settings dialog

➲ **To change file object settings:**

1. Select the **File Object** tab from the **Run Time Setting** window.
2. To enable the database to use external file objects, click on the **Enable User File Object** check box.
3. Enter a path into or select the browse button ... next to the **System File Object Directory** field to indicate the location of the System File Object Directory.
4. Select from **Files per FO Subdirectory**:

   To have DBMaker always store system file objects in the system file object directory, select always in the same directory.

   To have DBMaker create subdirectories with a set number of file objects in each subdirectory, select the option button next to 100 ~10000 and enter the threshold number of file objects (between 100 and 10000) at which a subdirectory is declared full and a new one created.
5. Select one of the other tabs at the top (to change other runtime settings) or select **OK** from the bottom of the **Run Time Settings** window.

Changing System Control Settings

The system control page allows settings to be changed that directly affect database performance. It also allows distributed mode to be enabled during the runtime and allows for the number of pages to extend a file when an autoextend tablespace is full.

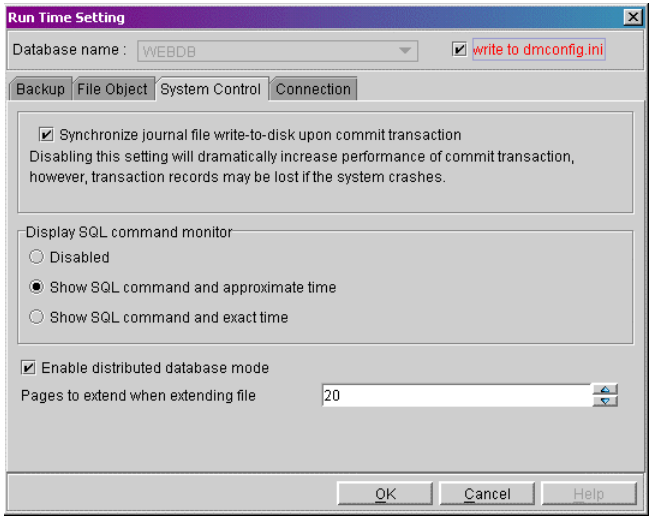


Figure 22-3The System Control page of the Run Time Setting dialog

JOURNAL FILE SYNCHRONIZATION

Under normal operating conditions, when a transaction is committed, DBMaker initiates a two-step process with regards to the journal blocks involved in the transaction. The first step is to flush the relevant journal blocks into the file system of the operating system, or the operating system's cache. The second step is to force the operating system to write these journal blocks to disk, so that the transaction is preserved in the event of a system crash.

It is possible to greatly increase the performance of the commit transaction process if the second step of this process is omitted. The journal blocks in this case remain in the operating system's cache until the operating system writes them to disk. However, if

the system crashes at this point, the records of any transactions that were still in the operating system's cache will be lost.

DISPLAY MODE OF SQL COMMAND MONITOR

Display Mode affects the display content of the **SQL_CMD** and **TIME_OF_SQL_CMD** columns in the **SYSUSER** system table. JDBA Tool Users can view information about users accessing the database with the Database Monitor function. **Session Information** displays users currently connected to the database. The columns *Current SQL command* and *Time of Current SQL Command* display the most recent SQL transaction committed by the user and the time of execution, respectively. **Display Mode of SQL Command Monitor** sets the configuration for how these attributes will be displayed. No SQL commands are shown if **Disabled** is clicked. The most recent SQL command executed by the user and its approximate time of execution are shown if **Show SQL command and approximate time** is clicked. The most recent SQL command executed by the user and its exact time of execution are shown if **Show SQL command and exact time** is clicked. Displaying the exact time of execution uses more CPU resources and slows down the database. The default mode is **Show SQL command and approximate time**

ENABLE DISTRIBUTED DATABASE MODE

This setting enables the database to be used in distributed mode. It must be enabled for synchronous table replication to work. Synchronous table replication is set using the JDBA Tool. For more information on distributed data, synchronous table replication, or coordinator and participant databases, refer to the *JDBA Tool User's Guide*, or the *Database Administrator's Guide*.

PAGES TO EXTEND WHEN EXTENDING A FILE

When all pages in a data file or BLOB file are full, DBMaker can automatically extend the number of pages or frames in the file to allow the database to grow. The **Number of Pages to Extend While Extending File** setting tells DBMaker how many pages or frames to add to the full file in the event that it is full. If the database administrator expects that the database will grow very quickly, then a higher number should be

picked to lessen the frequency at which the file is appended. One page is equal to approximately 4KB.

➲ **To change System Settings:**

1. To disable synchronized journal block writing to disk, remove the check mark from the **Synchronize journal file write-to-disk upon commit transaction** checkbox.
2. Select a setting for the SQL command monitor:

   Select **Disabled** to not display any SQL commands

   Select **Show SQL command and approximate time** to show the most recent SQL command executed by the user and its approximate time of execution.

   Select **Show SQL command and exact time** to show the most recent SQL command executed by the user and its exact time of execution.
3. To enable distributed database mode, check the **Enable Distributed Database Mode** box.
4. Enter the number of pages to extend a file in the **Number of Pages to Extend While Extending File** `field`.
5. Select one of the other tabs at the top (to change other runtime settings) or select **OK** from the bottom of the **Run Time Setting** window.

Changing Connection Settings

The Connection page of the Run Time Setting dialog allows you to prevent new connections handles from being made. This is useful if the database is experiencing performance problems and needs to be evaluated, or to allow more processor resources to be used for critical database operations. Database administrators may enable or disable this function at any time from the DBMaker server.

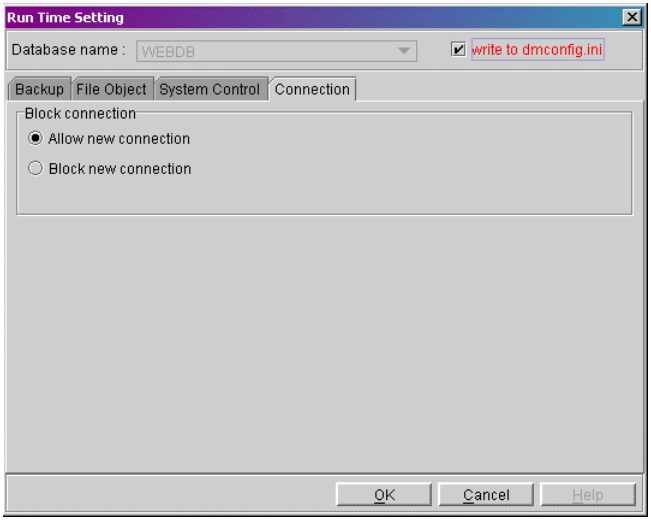


Figure 22-4 The Connection page of the Run Time Setting dialog

➲ **To change connection settings:**

1. To allow new connections to be made to the database, select **Allow new connection**.
2. To prevent new connections from being made to the database, select **Block new connection**.

22.2 Setting Preferences

JDBA Tool allows you to set preferences for the GUI. The following settings are available in the user preference window:

Language Preference: JDBA Tool supports English, Chinese, and Japanese interfaces.

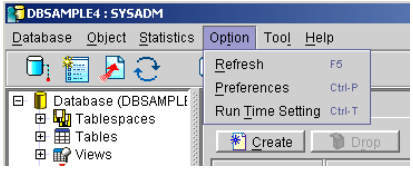
Refresh rate: Determines the rate at which database monitor information is updated to the screen.

Show System Table: Shows system objects in the tree view.

Show SQL Script: Allows the DBA to view and edit the SQL script that corresponds to functions carried out in JDBA Tool.

➲ **To Access the Preferences window:**

1. Select **Preferences** from the **Option** menu on the menu bar.



2. The **User Preference** window will appear with the **General** page displayed.

Setting General Options

Items included on the General Options page of the **User Preference** window are language selection and auto refresh options. Language selection toggles the GUI language; currently English and Chinese are supported by the user interface.

The **Auto Refresh for Database Monitor** check box activates or deactivates the automatic refresh function. All statistics shown in the Database Monitor are periodically automatically updated by the automatic refresh function. The time interval is indicated in the **Auto Refresh Time** field.

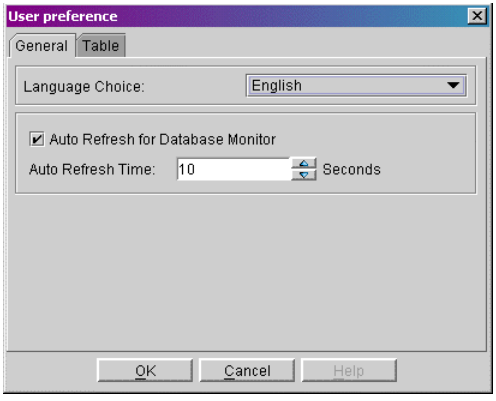
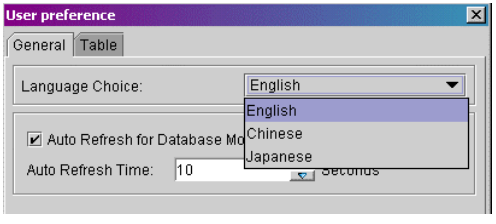


Figure 22-5 The General page of the User Preference Window

➲ **To change the GUI language setting:**

1. Select the language for the GUI from the **Language Choice** drop down menu.

NOTE *To view Chinese your operating system must support Big 5 Chinese character coding.*



➲ **To Change the Auto Refresh for Database Monitor settings:**

1. Selecting the check box toggles the automatic refresh function
2. To change the **Auto Refresh Time**, click the field and enter a new value.

NOTE *Setting a low **Auto Refresh Time** will use more processor resources and can substantially slow down the database.*

Setting Table Options

The **Table** page of the **User Preference** window contains the following settings:

Show System Table: Shows system objects in the tree view.

Show SQL Script: Allows the DBA to view and edit the SQL script that corresponds to functions carried out in JDBA Tool.

Records Retrieved: sets the number of records to retrieve from a table at one time when viewing a large number of records.

You can choose to disable this feature by changing the setting in the user preference window, or by enabling **do not show SQL command** at the bottom of the **Alter SQL Command** window.

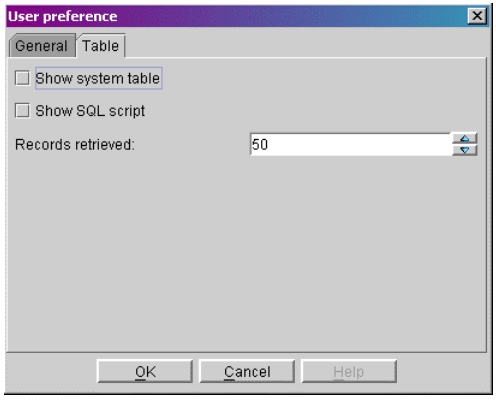


Figure 22-6 The Table page of the User Preference Window.

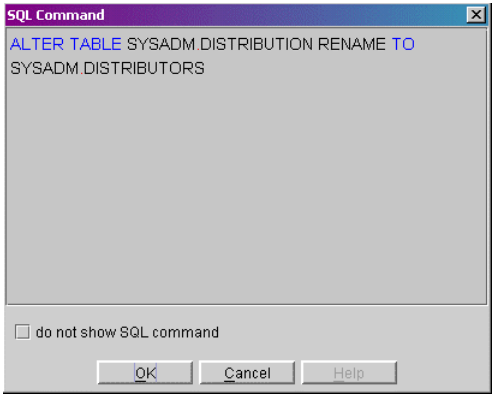
➲ **To view system tables in the tree view:**

1. Select the check box next to **Show system table**. A check mark in the box indicates that system tables are shown.

➲ **To view SQL scripts:**

1. Select the check box next to **Show SQL script**. A check mark in the box indicates that the **Alter SQL Command** window will open before execution.

   Subsequent actions in JDBA that have a corresponding SQL Command will open a window as in the following figure:



➲ **To Alter SQL Scripts before execution:**

1. Select the check box next to **Show SQL script** on the **Table Page** of the **User Preferences** window. A check mark in the box indicates that the **Alter SQL Command** window will open before execution.
2. Follow the procedure of the function that you want to apply to the database.
3. Selecting **OK** at the end of the procedure will open the **Alter SQL Command** window. You can edit the command in the text field.
4. Selecting **OK** will execute the command; selecting **Cancel** will abort the whole command.

Glossary

Asynchronous Table Replication
A set of columns selected to form a projection that is replicated to the destination tables of another database. Asynchronous table replication occurs according to a schedule. Changes made to the source table are saved to a log file that is used to periodically update the destination tables.

BLOB
Acronym for Binary Large Objects. A table that consists of one column of LONGVARCHAR or LONGVARBINARY type data.

BLOB Frames
Measurement unit used to allocate disk space for BLOB type data.

Concurrency Control
A system of locks placed on objects to prevent multiple users from simultaneously manipulating the same data sets.

Coordinator Database
In a distributed database environment, the database that the client connects to. If the client accesses data from another database then that database is a participant database.

Daemon
A routine that automatically executes at a set time interval.

Data Pages
Data measurement unit for allocation of disk space.

Destination Database
The database that receives data for table replication. A database containing destination tables for a synchronous or asynchronous replication.

Destination Tables
Tables that receive replicated data from a source table. The table on the destination database that the data is replicated to.
Distributed Database Environment
A system of networked remote databases that allow for any table in the system to be accessed by a client on any of the participant (remote) databases.
Exclusive Lock (X Lock)
An access block placed on a database object that prevents other users from accessing the object
Foreign Key
A column or set of columns whose rows contain the same values as the set of columns in the primary key or unique index of another table.
Fragment
Also called a horizontal partition, a fragment is the replication of a given range of data tuples.
Journal Blocks
Internal data measurement unit (514 bytes) that DBMaker uses to manage journal data.
Journal Buffer
Upper memory where current journal blocks are stored before writing them to disk.
Journal Pages
Measurement unit for allocation of disk space.
Lock
Locks allow only one user update and delete permission on the locked object at the time the object is locked.
Page Buffer
Upper memory allocated for data pages accessed by a user.
Participant Database
In a distributed database environment, a database that is accessed by the client through a coordinator database.
Primary Database
The source database for database replication.
Primary key
A column or set of columns in a table that contain values that uniquely identify the rows in the table.
Projection
The selected columns from a base table chosen for replication.

Publication
A data set on the source table available for the replication.
Remote Database
A database located on another server from that being accessed by the client.
Remote Tables
Tables on a database located on a server that is different from the one the client is connected to.
Replication Domain
The replication fragment (horizontal partition) and projection (vertical partition) together are called a replication domain. It is the range of a table's data that is replicated.
Select Lock (S Lock)
An access block placed on a database object that allows other users to browse the object but not to update or delete any part of it.
Slave Database
A read-only database that receives data from a database replication
Source Database
The database that contains source tables used to replicate.
Source Table
The table on the source database that the replicated data is from.
Subscription
The data set on the destination table to receive a publication.
Synchronous Table Replication
A set of columns selected to form a projection that is replicated to the destination tables of another database. Synchronous table replication occurs simultaneously – changes made to the source table are simultaneously made to the destination table(s).

Index